# Dynamic Power Management for QoS-Aware Wireless Multimedia Sensor Networks

by

Afshin Fallahi

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES
*****
COPYRIGHT PERMISSION

# Dymanic Power Management for QoS-Aware Wireless Multimedia Sensor Networks

BY

Afshin Fallahi

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

DOCTOR OF PHILOSOPHY

**Supervisor:** Prof. E. Hossain

## ABSTRACT

The integration of multi-hop capability into conventional wireless networks is perhaps the most promising architectural upgrade in next generation wireless networks with high throughput and coverage requirements. Multi-hop communication covers wireless sensor networks as an important component in the future heterogeneous wireless networks. In a distributed sensor network, each node acts as a relay node to forward data packets from other nodes. More recently, the production of cheap CMOS cameras and microphones, which can acquire rich media content from the environment, has created a new wave into the evolution of wireless sensor networks towards wireless multimedia sensor networks. Wireless multimedia sensor networks are gaining increasing popularity due to numerous potential applications such as video surveillance, environmental and habitat monitoring, etc. In a wireless multimedia sensor network, limited battery power at the wireless sensor nodes along with the transmission quality requirements for multimedia traffic make quality of service (QoS) provisioning a very challenging task.

In this work, the tradeoff between the QoS and the energy saving performance is first analyzed. In particular, a novel queueing analytical framework is presented for performance evaluation of a node in a multi-hop wireless network with distributed and energy-aware medium access control (MAC) protocol. The MAC protocol in the node acts as a server for different types of multimedia arrival traffic in the node with different priorities. A vacation queueing model is also used to model the sleep and wakeup mechanism of the server. Two different power saving mechanisms due to the standard exhaustive and the number-limited exhaustive vacation models (both in multiple vacation cases) are analyzed to study the tradeoff between the QoS performance of the relayed packets and the energy saving at a relay node. We use matrix-geometric method to obtain the stationary probability distribution for the system states from which the performance metrics are derived. Using phase-type distribution for both the service and the vacation processes and combining the priority queueing model with the vacation queueing model make the analysis very general and comprehensive.

A dynamic power management framework is developed based on a Markov decision process (MDP) for a wireless multimedia sensor node to improve the energy saving performance while satisfying the multimedia transmission quality requirements. Dynamic programming and reinforcement learning are used as two different approaches to solve the MDP problem. The dynamic programming framework considers the video traffic arrival process in the sensor node, the sleep and wakeup processes in both the camera and wireless transceiver electronics, the queue status, and the multi-rate wireless channel condition. Dynamic programming approach is used to find the optimum policy to achieve the desired performance measures in an energy-limited sensor node.

To overcome the curse of dimensionality in dynamic programming approach, a reinforcement learning-based distributed dynamic power management framework is presented. The queueing model in this case captures bursty nature of the multimedia (e.g, video/voice, data) traffic arrival process and prioritization between video/voice and data traffic. A model-based reinforcement learning approach is used to solve the MDP problem in which the structure of the transition probability matrix in the MDP formulation is exploited to reduce the convergence time in the learning process.

**Examiners:**

Prof. E. Hossain, Supervisor, Dept. of Electrical & Computer Engineering

Prof. A. S. Alfa, Member,Dept. of Electrical & Computer Engineering

Prof. J. Misic, Member, Dept. of Computer Science

Prof. S. Noghanian, Member, Dept. of Electrical & Computer Engineering

Prof. A. Boukerche, External Examiner, University of Ottawa

# Table of Contents

# List of Figures

# List of Tables

# *Acknowledgement*

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

In the first place I would like to record my gratitude to Professor Ekram Hossain for his supervision, advice, and guidance from the very early stage of this research.

I would also like to thank the committee members, Dr. Attahiru Sule Alfa, Dr. Sima Noghanian, and Dr. Jelena Misic for their helpful advice and comments.

I would like to thank all my friends here in Winnipeg and specifically Haitham Abu Ghazaleh for taking the time to carefully read my thesis and for his helpful feedback and suggestions.

I wish to thank my parents for their prayers and their encouragements throughout my graduate work in Canada.

Finally, I would like to give my special thanks to my wife Leila whose patient love enabled me to complete this work. One of the best experiences that we lived through in this period was the birth of our daughter Bahar, who provided an additional and joyful dimension to our life mission.

# Chapter 1

# Introduction

## 1.1 Wireless Multimedia Sensor Networks: An Introduction

Wireless sensor networks [1] are composed of large sets of small, densely deployed inexpensive devices with hardware for sensing and radio for communication with each other through wireless transmission. They have emerged during the last years due to the advances in low-power hardware design and the development of appropriate software, which enables the creation of tiny devices which are able to compute, control and communicate with each other.

Sensor networking is an emerging technology for the future with many potential applications [2]. Introduction of low-power and cheap microprocessors as well as low-power RF design are fueling the rapid growth of this technology. Real-time traffic monitoring, smart environment control, habitat monitoring, health care, weather data gathering as well as military applications such as distributed surveillance and targeting are some examples of applications of wireless sensor networks.

More recently, the production of cheap CMOS cameras and microphones, which can acquire rich media content from the environment, created a new wave into the evolution of wireless sensor networks. Thus, a new class of wireless sensor networks came to the scene, the wireless multimedia sensor networks [3, 4].

A wireless multimedia sensor network interconnects low-cost and low-power sensor nodes with limited data processing capabilities which are equipped with miniature camera and wireless transceiver. Typically, wireless multimedia sensors are small in size and communicate over short distances. Each multimedia sensor node captures video scenes and audio as well as other scalar data and transmits them to a central base station or a data sink. One such scenario is depicted in Fig. 1.1 for a surveillance application or habitat monitoring. A heterogeneous sensor network may consist of video, audio and conventional sensors (or any combination of them) to gather audio-visual as well as other sensory data. However, regardless of the gathered information by the node itself, each node is capable of relaying multimedia to next node or to the sink in a multi-hop manner.

A multimedia sensor is generally battery-operated or operated by power generated dynamically via solar panels or wind-powered generators [5]. Therefore, energy consumption is a fundamental issue associated with the quality of service (QoS) provisioning in wireless multimedia sensor networks.

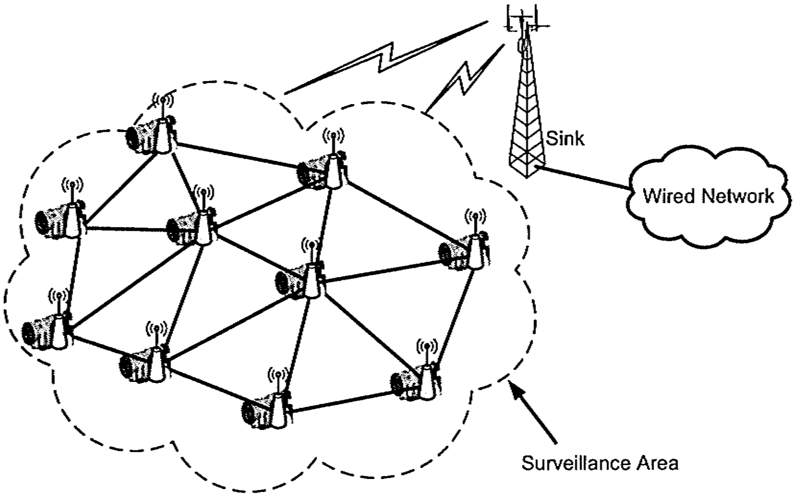


**Figure 1.1.** *Network Architecture for wireless multimedia sensors in surveillance application.*

Wireless multimedia sensor networks, apart from boosting the existing application of wireless sensor networks, will create new applications. They have been envisioned for a wide range of applications including environmental and habitat monitoring, emergency response, health monitoring, and video security [3],[6]. In an environmental monitoring application scenario, the video sensors will require to collect, store, and then transmit only the most important video sensory data. In an emergency application scenario, the video sensors capture and transmit high-quality video for a specified period of time to provide emergency response personnel with the critical information they need throughout the incident. Multimedia sensors improve the ability to prevent, detect, respond to and recover from both man-made and natural catastrophes.

Wireless multimedia sensor networks have some unique properties when compared to the conventional data sensor networks. Data generation rate in a multimedia sensor is generally high and the required network bandwidth and power consumption for wireless transmission would be large, especially without any efficient compression scheme before transmission. Efficient video compression often involves sophisticated and computationally intensive encoding operations [6]. Therefore, a significant portion of the total energy supply in a wireless video sensor is consumed by the unit responsible for receiving the video frames and processing them (i.e., for camera operation).

## 1.2 Dynamic Power Management in Wireless Multimedia Sensor Networks

Dynamic power management (DPM) is an effective technique to dynamically control power consumption in battery-operated wireless sensor nodes [7]. A widely employed policy is to place the sensor nodes in sleep mode in which the energy consumption in a node as well as its operational capabilities are reduced. The simplest approach to put a sensor node in sleep mode to extend its life time is to use a fixed duty cycle [8]. This is not efficient for sensor networks specifically with bursty traffic such as video traffic. In another approach, sensors in sleep mode are woken up by external means, if needed, by using a low-power wakeup radio dedicated for this purpose [9]. Yet another approach is to modify the power-saving function in the medium access control (MAC) protocol to reduce the energy consumption as much as possible [10], [11].

In general, a power manager can monitor the power consuming components in the sensor node and decide when a sensor node should go to the sleep or idle state and the amount of time to stay in these states. A *distributed* power management and communication framework in a sensor node should be designed by exploiting the knowledge about the application QoS requirements, traffic load and network conditions, and the energy constraints in the sensor nodes. The choice of the policy that minimizes power consumption under performance constraints is a constrained optimization problem. Again, since there is a direct relationship between the rate of communication (and hence the delay/throughput performance) and the quality of the communication channel, for a given amount of energy for communication, better performance can be achieved by exploiting the physical properties of the channel.

Due to the limited and generally irreplaceable power sources in sensor nodes, a power management and communication framework in a sensor node should be designed by exploiting the knowledge about the traffic, the channel condition, and the energy constraints in the sensor nodes (e.g., battery condition). Such a framework should be capable of extending the battery lifetime as well as maintaining the application QoS requirements. This generally involves all the layers of the system, and in particular, the physical and the MAC layers in the communication protocol stack. For such a power management and communication framework in a multimedia sensor network, the energy consumption behavior of individual sensor nodes needs to be analyzed under a dynamic communication environment (i.e., fading environment). Since a sensor node is the basic operational unit of a wireless multimedia sensor system, performance analysis of such a node is the first step towards performance analysis and optimization of the entire sensor network.

## 1.3 QoS Provisioning in Multimedia Sensor Networks

While there exist a significant amount of research results on issues related to data gathering, routing, and medium access control in wireless sensor networks, QoS provisioning for multimedia communication in these networks has remained unexplored. Many of the existing solutions proposed for multimedia communications in wireless and Internet environments cannot be directly applied to the sensor networks due to the unique characteristics and resource constraints in sensor networks. High and variable error rates and delays in wireless channels and the intrinsic limitations in sensors due to limited energy and simple hardware are significant obstacles for providing QoS support for multimedia applications in sensor networks. In particular, multimedia applications impose significant resource requirements on bandwidth and energy-constrained wireless sensor networks.

In this section, we present some major issues and the related approaches involved in provisioning QoS in multimedia sensor networks where we emphasize more on the communications and energy saving aspects (rather than the computations and data-centric aspects such as data fusion). Different physical layer techniques and energy-conserving MAC protocols for wireless multimedia sensor networks are then reviewed.

### 1.3.1 Distributed Energy-Aware MAC Protocols with Service Differentiation

Although contention-free protocols would be more desirable for multimedia applications, they are complex and require centralized control which makes them unfavorable for the sensor networks [4]. Due to the lack of any infrastructure, distributed QoS-aware MAC protocols would be required for radio resource sharing in wireless multimedia sensor networks. Such a MAC protocol should be also adaptive to the variations in traffic pattern as well as network topology. MAC protocols in wireless multimedia sensor networks have to be energy-efficient, because encoding, transmission, and retransmission operations cause fast depletion of energy at these battery-operated sensor nodes. For this reason, a dynamic sleep/wake up discipline (rather than static duty cycle as in Sensor-MAC (S-MAC) [11]) is generally preferred to achieve efficient energy conservation in a power-limited wireless sensor node.

Distributed and contention-based (or random access) schemes, in which nodes contend for access to the shared medium, are more suitable for autonomous and infrastructureless sensor networks. However, they are less likely to be able to guarantee QoS for real-time traffic and basically support asynchronous data transfer on a best-effort basis. Scalable and easy to use contention-based schemes can handle varying and bursty traffic unlike the contention-free schemes for which a fair accurate estimate of the traffic is a must. However, for the case of distributed and random access scheme, nodes still need to contend fairly with each other to access the shared media.

In presence of both video/voice traffic and scalar data traffic, a priority mechanism would be required in the MAC layer to prioritize video/voice traffic, which has more stringent delay and bandwidth requirement over scalar data. Wireless sensor nodes need to be equipped with the appropriate scheduling algorithms to handle such a prioritized scheduling.

In a wireless video sensor node, scalable video encoder can be used to encode the raw data into multiple layers with varying degrees of importance. The base layer containing low-frequency and low-resolution data is the most important, and the enhancement layers containing high-frequency and high-resolution refinements may be ignored depending on the availability of the resources. It has been shown in the literature that multi-layer encoding provides better performance. The MAC layer in a resource-limited node has to be aware of the priority of the video frames and provide priority mechanism not only for the video traffic over scalar data traffic, but also for the base layer video frames over enhancement layers frames.

**Table 1.1.** *Qualitative comparison among different sensor MAC protocols for wireless video sensor applications.*

| Protocol | Priority mechanism | Traffic adaptivity | Channel access | Energy efficiency | Delay performance |
|---|---|---|---|---|---|
| NAMA [10] | Inter-node | No | TDMA | Low due to high minimum duty cycle | Higher than contention-based schemes |
| B-MAC [11] | No | No | CSMA | Ultra low power by adaptive preamble sampling | Long preambles introduce more delay |
| S-MAC [11] | No | No | CSMA | Static duty cycle decreases energy efficiency | Static duty cycle increases delay |
| T-MAC [10] | No | Yes | CSMA | Improved by adaptively varying the active period | Similar to S-MAC |
| Z-MAC [11] | No | Yes | TDMA/CSMA | Low in low data rates because of large backoff window size | Improves in high contention networks |
| DSMAC [10] | No | Yes | CSMA | Improved in low traffic scenario by using dynamic duty cycle | Improved in high traffic scenario by using dynamic duty cycle |
| TRAMA [10] | No | Yes | TDMA/CSMA | Low because of complexity | higher than contention-based schemes |
| WiseMAC [10] | No | Yes | Non-persistent CSMA | Large power consumption due to overhead in reception | Sleeping delay introduced at each hop |

The major characteristics of some of the MAC protocols for sensor networks proposed in the literature are summarized in Table 1.1. These characteristics are related to the adaptability and performance of the MAC protocols for video transmission over wireless sensor networks. The MAC protocols considered here are Node Activation Multiple Access (NAMA) [10], Traffic-Adaptive Medium Access (TRAMA) [10], Berkeley MAC (B-MAC) [11], Sensor MAC (S-MAC) [11], Zebra-MAC (Z-MAC) [11], Dynamic Sensor-MAC (DSMAC) [10], and Timeout-MAC (T-MAC) [10]. Note that, although NAMA proposes a priority mechanism among the nodes in the network, no priority is provided in the case of heterogeneous traffic served by a node.

### 1.3.2 Channel-Aware Adaptive Radio

Channel-aware radio protocols such as those for adaptive error control and joint source and channel coding can significantly affect the energy saving performance in wireless multimedia sensor networks. Error control is an issue that spans several layers in the protocol stack from physical to transport, or even up to the application layer [12]. Automatic repeat request (ARQ) and forward error correction (FEC) are the two commonly used error control mechanisms. Using a static FEC mechanism (i.e., ignoring the dynamic channel conditions) reduces the achievable data rate and hence causes additional delay in transmission. ARQ may result in an even more significant increase in delay and delay variations than FEC [12]. Buffering and retransmissions require additional memory in both transmitter and receiver and the delay may be unacceptable for video transmission for which QoS is a major concern.

The performance of ARQ scheme depends on the channel state. If the channel is good, retransmissions occur rarely and ARQ performs well. However, in poor channels, number of retransmissions increases and the QoS performance of ARQ degrades. The energy consumption increases not only at the transmitter, but also at the receiver, since it has to wait for the acknowledgements (without going to the energy-saving mode). Avoiding transmission during bad channel conditions reduces the number of unsuccessful transmissions and saves energy in the nodes when it is known that the receiver has little chance of receiving the data correctly.

Joint source and channel coding techniques mainly focus on designing codes that minimize the overall distortion of the transmitted bits over the channel. For the case of energy-limited video sensor networks, power consumption should be also taken into account. Distributed coding of the multimedia source is a promising approach which contributes to significant energy savings in wireless sensor networks.

Distributed source coding techniques may find a good tradeoff between processing and communication cost by reverting the traditional balance of complex encoder and simple decoder in order to fit to the power constraints of sensor nodes [13]. Distributed source coding and joint source-channel coding can reduce the size of multimedia data significantly while making it resilient to errors [4]. Both of them result in improved energy saving performance.

The existing schemes for joint source and channel coding mostly consider point-to-point wireless links, and neglect interference from neighboring nodes and multi-hop routes. Therefore, performance analysis models and algorithms are required which consider these issues in a wireless multimedia sensor network presumably through a cross-layer optimization framework.

## 1.4 Research Issues in Wireless Multimedia Sensor Networks

### 1.4.1 Distributed and Energy-Efficient MAC Protocols

Distributed MAC protocols [14] are necessary in a distributed multi-hop network where the communication between any two nodes typically relies on the other intermediate nodes with limited battery power [15]. Distributed nature of the network is a challenging issue. All nodes play the same role and no one is supposed to organize the resource sharing as in cellular networks.

A major design consideration for ad hoc network MAC protocols is the power consumption of individual nodes, and the overall power consumption of the network because of limited battery power in the nodes. In centralized networks, a resourceful base station is responsible for managing channel access and allocation, while nodes consume most of their power for data transmissions. In ad hoc networks, however, the absence of a base station places the burden of control on the nodes and furthermore increases the chances of collisions and channel assignment conflicts that lead to higher power consumption in the form of control signaling and retransmissions.

The energy expenditure in a node is typically dominated by the communication subsystem and especially the transmission unit. A major research issue in distributed mobile computing is to design efficient mechanisms for minimizing energy consumption in the wireless terminals ( [16, 17, 9, 18, 8, 7, 19, 20, 21]). A simple way to minimize energy consumption in a node is to allow the node to go to the *sleep mode.* While in the active state the node is fully working and is able to transmit/receive data, in sleep state, transmission circuit, which is the most power consuming part of the node, is disabled and node does not really transmit. Receiver may also be disabled for more energy saving. Therefore, it does not use a significant amount of battery power.

Although this power saving method is the most obvious means of power conservation and provides energy gains, it's not enough in sensor nodes that communicate using short data packets. The shorter the packets are, the more the dominance of startup energy is. If we blindly turn the radio off, over a period of time we might consume more energy than if the radio had been left on.

A wide variety of solutions were introduced for power management in battery-operated wireless multi-hop networks by putting a node in the sleep mode [8],[7]. The problem of energy efficient protocol design for ad hoc and sensor networks was addressed quite extensively in the literature ( [22], [23], [24]). In [18], a wake up scheme was designed to reduce the degradation in end-to-end delay due to the sleeping/wakeup mechanism in a sensor network. Using Erlang theory, the tradeoff between energy conservation and quality of service in a wireless sensor network was quantitatively analyzed in [24]. For wireless sensor networks, a topology and energy management scheme was proposed in [16]. However, the queueing dynamics at a node due to the sleep/wakeup mechanism was not investigated in the above works.

HIPERLAN [25] also allows nodes to go into sleep mode to conserve power. Such nodes are called p-savers, and must set up a specific wakeup pattern by notifying specialized neighboring nodes called p-supporters. Bluetooth [26] supports three low-power states: park, hold, and sniff. Park state provides the lowest duty cycle and thus the lowest energy consumption. In this state a node releases its MAC address but remains synchronized with the piconet. The node wakes up occasionally to synchronize and listen for broadcast messages. The next higher low-power state is hold state in which a node keeps its MAC address and transmits immediately after waking up. Finally, in the sniff state a node listens to the piconet more often than in the hold state, but still at a lower rate than normal.

### 1.4.2 Markov Model and Queueing Analysis of MAC Protocols

A Markov model to analyze the sleep/active dynamics in a sensor node was developed in [27]. In [20], a discrete-time Markov chain model was used to represent the sensor dynamics in sleep/active mode while taking into account the channel contention and the routing issues. A vacation model along with an M/G/1 queue for MAC layer service process was used in [28] to analyze the sleeping mechanism in a sensor network. However, no prioritization was assumed between different arrival streams. Allowing differentiated services would require consideration of a priority queueing model. In a traditional priority queueing system, the server is ready all the time to serve one of the queues. When the server is serving one of the queues, it is said to be in vacation for the other queues. In the presence of a sleeping mechanism, however, when it is assumed that the server process in a node is in vacation (i.e., in sleep mode), it serves none of the queues. Therefore, a system combining both the priority and the vacation models needs to be considered for modeling the sleep and the wake up mechanism in the MAC protocol at a node with differentiated services mechanism.

Priority queueing and vacation queueing models were analyzed separately in [29] and [30], respectively, using matrix-geometric method. Also, a few other works used matrix-geometric method of queueing analysis for wireless networks [31]. In [32], the distribution of transmission delay for packets over a wireless link with correlated errors was analyzed as a phase-type distribution assuming hybrid automatic repeat request (HARQ) type of error control mechanism. A performance analysis model for GPRS (General Packet Radio Service) networks was presented in [33] in which traffic inter-arrival time was modeled by a phase-type distribution (to capture the correlation in the arrival process) and the transmission time of IP (Internet Protocol) packets was modeled as an MMAP (marked Markovian arrival process).

One of the earliest works on *priority-based* access schemes can be found in [34] which presented a prioritized carrier sense multiple access (CSMA) protocol for broadcast channel access. Also, in [35], a prioritized carrier sense multiple access with collision detection (CSMA-CD) scheme was analyzed assuming Poisson packet arrival, where higher priority packets are transmitted in 1-persistent mode and lower priority packets are transmitted in non-persistent mode.

### 1.4.3 Analysis Under Non-Saturated Traffic

A saturation traffic condition is often considered in the analysis of random access protocols [36]. Under this condition, all stations are assumed to be saturated such that whenever they successfully complete a data frame transmission, they are immediately ready for another data frame transmission. But this assumption does not provide any insight into the performance of a protocol under real statistical traffic conditions, especially in sensor networks. Performance of IEEE 802.11 distributed coordination function (DCF) MAC protocol in the saturation state was investigated in [36],[37],[38]. In [39], [40], more practical queueing models for the IEEE 802.11 DCF MAC were proposed which incorporated practical packet arrival processes. However, the service rate for each node was based on the results obtained in [36], where saturation state was assumed. This limitation was overcome in [41], where performance analysis in non-saturation state was considered by introducing probability generating functions, which allow the computation of the probability distribution function (pdf) of the delay. However, computing the pdfs using the proposed method has a high computational cost, and therefore, the approach is of limited practical use. Furthermore, energy efficiency and service differentiation at the MAC layer were not considered.

### 1.4.4 QoS in Differentiated Services Networks

Rapid development of multi-hop wireless networks, such as wireless ad hoc/sensor networks and wireless mesh networks, has promoted growing curiosity in studying the QoS performance and the related control mechanisms in these systems [42]. Several approaches for QoS support in a multi-hop relay network such as a sensor network were discussed in [43],[44].

A critical concern is the efficient use of the available energy at a node and thus, extending the lifetime of the network, while satisfying the QoS requirements for the packets relayed through that node [42]. Undoubtedly, a tradeoff exists between the node energy saving and the network performance in terms of waiting time for the relayed packets and packet loss rate (due to buffer overflow) at a relay node. Again, to support traffic with different QoS requirements, a prioritization mechanism is required at the MAC protocol to achieve service differentiation.

Most of the existing approaches for distributed MAC, such as the slotted Aloha protocol and IEEE 802.11 DCF, fail to provide differentiated service to users. There is no separation between high and low priority flows, which results in equal competition for channel access. This leads to severe performance degradation in such environments of several multimedia applications, having tight bandwidth, delay and/or jitter requirements. IEEE 802.11e Enhanced DCF (EDCF) aims to address these issues by incorporating additional mechanisms to provide support for service differentiation.

A great amount of work on wireless MAC protocols in the literature focused on supporting heterogeneous traffic over the shared wireless channel (e.g., in [45],[46]). A number of QoS differentiation mechanisms for the heterogeneous wireless networks were discussed in [47]. In [48], dual queue scheme, a software upgrade-based method, was proposed to provide QoS for voice over IP (VoIP) service over legacy 802.11 WLAN. With two separate queues on top of the 802.11 MAC controller for real-time (RT) and non real-time (NRT) packets, a strict priority queueing discipline was implemented to serve these two queues in order to give higher priority to the RT packets. In this model, the NRT queue is never served as long as the RT queue is non-empty. Moreover, the energy efficiency issue was not considered.

### 1.4.5 Multi-Rate Transmission and FSMC Channel Model

A current trend in wireless communication is to enable devices to operate using many different transmission rates and many current and proposed wireless networking standards have this multi-rate capability. It stems from the fundamental properties of wireless communication. Due to the physical properties of communication channels, there is a direct relationship between the rate of communication and the quality of the channel required to support that communication reliably. Multi-rate transmission is assumed to be achieved through adaptive modulation and coding (AMC) [49] to adjust the transmission rate according to the channel condition.

A Markov channel for mobile propagation channels was presented in [50]. The work in [51] provided a throughput analysis for adaptive modulation systems with ARQ over fading channels using Markov error structure. A finite state Markov chain (FSMC) model for Nakagami-$m$ fading channel was proposed in [49]. An analytical framework was presented in [52] that uses this model to analyze the performance of multi-rate wireless networks where the transmission batch size varies according to the chosen transmission modes in the physical layer. However, neither a distributed MAC protocol nor an energy management scheme was considered in this work.

## 1.5 Contributions

In this work, we develop a novel queueing analytical model to evaluate the performance of an energy-aware distributed MAC protocol for differentiated services multi-hop communication in wireless multimedia sensor networks considering non-saturated bursty traffic arrival patterns. It enables us to investigate the QoS performance as the node dynamics (due to the sleeping and wake up mechanism) varies. Performance analysis in this work exploits the radio link layer and physical layer characteristics to enable cross-layer design for energy-aware multi-hop networks with distributed MAC protocol.

- A novel analytical model, which combines priority queueing with vacation queueing, is developed in Chapters 2 and 3 to analyze the tradeoff between the QoS and the energy efficiency at a node in a relay network. While we study unlimited buffers in Chapter 2, analysis in Chapter 3 considers the case of finite buffer capacities. Two different disciplines, namely, standard exhaustive vacation and number-limited exhaustive vacation are analyzed, which correspond to two different sleeping mechanisms for the service process at a node. Bursty traffic arrival process and non-saturation MAC protocol state are considered in the queueing analysis. As an application of our analysis, we demonstrate how to obtain the optimum wake up probability for the two different sleep mechanisms under different QoS constraints.
- An MDP-based formulation is developed in Chapter 4 for the distributed dynamic power management problem in a wireless video sensor node considering

the queueing dynamics at a sensor node under a realistic video traffic arrival process and a general wireless fading channel model. A dynamic programming approach is used to find the optimum policy which achieves the desired performance in an energy-limited sensor node. The complexity of the solution approach is quantified and also the structure of the transition probability matrix in the MDP formulation is used to reduce the computational complexity of the solution.

- In Chapter 5 an MDP-based model is developed for the distributed dynamic power management problem in a wireless multimedia sensor node considering the queueing dynamics at a sensor node. A Q-learning approach is used to find the optimum policy which achieves the desired performance in an energy-limited sensor node. The structure of the transition probability matrix in the MDP formulation is used to reduce the convergence time in the learning process. The computational complexity of the solution is much less than that for model-based approaches.

## 1.6 Outline of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, the tradeoff between the QoS and the energy consumption is analyzed in a node in an energy-limited multi-hop wireless network. A queueing analysis is presented for the case of infinite buffers for both low-priority and high-priority queues in the node. Two different strategies for vacation are applied and the queueing analytical models for both standard and number-limited exhaustive vacation models are developed. Specifically, we setup the Markov chain and present a matrix-geometric analysis of this queueing model for non-preemptive priority queue with multiple vacations. Steady state distribution of the packets in the system and the different QoS measures are obtained. We also demonstrate how to apply the general analytical model to specific MAC protocols and demonstrate how to obtain the optimal system parameter settings. Finally, typical numerical results obtained from the analytical models are presented and validated through simulations.

Chapter 2 establishes a framework that addresses the two main requirements of multimedia transmission over wireless sensor networks, namely, energy saving and service differentiation. Using a queueing analysis approach, this framework is obtained by combining priority queueing model with vacation queueing model and providing analytical solution for this unified model.

The analytical methodology in Chapter 2 is also used in Chapter 3 for the case of finite queues where we distinguish between head-of-line (HOL) packet in the queue and other packets. Therefore, this is an enhanced version of the work in Chapter 2 for more realistic cases in wireless networks. A non-ideal channel is considered and retransmissions of the failed packets are taken into account.

Chapter 4 presents a dynamic power management framework for transporting MPEG (moving picture experts group)-coded video traffic over a distributed MAC protocol in a wireless video sensor network. The distributed dynamic power management problem is formulated as a Markov decision process (MDP). The MDP formulation is developed based on a vacation queue model [53] with Markovian arrival process (MAP), phase-type service distribution in the transmitter and phase-type sleep mechanism. The traffic source model here in the sensor node captures both short-range correlation and long-range dependence in MPEG video stream. The performance of the proposed framework is analyzed in terms of different QoS metrics considering an FSMC model [54] and multi-rate transmission feature along with ARQ-based radio link level error recovery for wireless transmission. Multi-rate transmission is assumed to be achieved through AMC to adjust the transmission rate according to the channel condition.

The approach in Chapter 5 is to overcome the limitations of the model presented in Chapter 4 and reduce the complexity of the system. In this chapter, a reinforcement learning-based dynamic power management framework is presented for QoS-aware wireless multimedia sensor networks. The power management framework uses a controllable sleep/wakeup mechanism considering the queueing dynamics at a sensor node. The queueing model at a sensor node considers bursty nature of the multimedia (e.g, video/voice, data) traffic arrival process and prioritization between video/voice and data traffic. The optimal power management problem is formulated as a MDP problem. A model-based reinforcement learning approach is used to solve the MDP problem in which the structure of the transition probability matrix in the formulation is used to reduce the convergence time in the learning process. Performance analysis results show that the proposed mechanism can achieve considerable energy saving in a wireless sensor node while providing the target level of QoS performance.

Some possible future research directions are outlined in Chapter 6.

## 1.7 Preliminaries

### 1.7.1 Markov Process

A Markov process is a stochastic process $X(t)$ such that

$$\begin{aligned} & P(X(t_{n+1}) = x_{n+1} | X(t_1) = x_1, \dots, X(t_n) = x_n) \\ & = P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n) \end{aligned} \tag{1.1}$$

In other words, the next state in the stochastic process depends only on the current state and not upon the history of the process.

Although this definition applies to Markov processes with continuous state-space, we will be mostly concerned with discrete-space Markov processes, commonly referred to as Markov chains. A discrete-time Markov chain is a discrete-time, discrete-space (with state-space $I \subseteq \mathbb{N}$) stochastic process such that

$$\begin{aligned} & P(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) \\ & = P(X_{n+1} = j | X_n = i) = p_{ij} \end{aligned} \tag{1.2}$$

$p_{ij}$ is called one-step transition probability from state $i$ to state $j$.

The transition matrix of the Markov chain is defined as follows:

$$P = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0j} & \cdots \\ p_{10} & p_{11} & \cdots & p_{1j} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{i0} & p_{i1} & \cdots & p_{ij} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{1.3}$$

Matrix $P$ is called *stochastic matrix* if satisfies the following conditions:

$$p_{ij} \geq 0 \quad \forall i, j \in I \tag{1.4}$$

$$\sum_{j \in I} p_{ij} = 1 \quad \forall i \in I \tag{1.5}$$

The probability of going from state $i$ to state $j$ in $n$ steps is also defined as $n$-step transition probability $p_{ij}^{(n)}$ by

$$p_{ij}^{(n)} = P(X_n = j | X_0 = i) \tag{1.6}$$

If we define $P^{(n)} := [p_{ij}^{(n)}]$, following Chapman-Kolmogorov equation we have $P^{(n)} = P^n$ where $P^n$ is the $n^{th}$ power of the matrix $P$. Then we define $\pi_n(i) := P(X_n = i)$ that represents unconditional probability that the system is in state $j$ at time $n$. We need to know $\pi_0(i) := P(X_0 = i)$ for all $i \in I$ to compute $\pi_n(i)$ as follows:

$$P(X_n = j) = \sum_{i \in I} P(X_n = j | X_0 = i)\pi_0(i) = \sum_{i \in I} p_{ij}^{(n)} \pi_0(i) \tag{1.7}$$

and in matrix form we have

$$\pi_n = \pi_0 P^n \tag{1.8}$$

and as a result

$$\pi_{n+1} = \pi_n P \tag{1.9}$$

If the Markov chain with matrix $P$ is *irreducible* and *aperiodic*, the limiting state distribution function $\pi(i) = \lim_{n \to \infty} \pi_n(i)$ exists and we can find $\pi$ by solving $\pi = \pi P$. We need also add normalization condition $\sum_{i \in I} \pi(i) = 1$ or in matrix form, $\pi e = 1$. $e$ is a column vector of ones.

A Markov chain is irreducible if for all $i, j \in I$, $j$ is *reachable* from $i$ and $i$ is reachable from $j$. A state $j$ is reachable from a state $i$ if $p_{ij}^{(n)} > 0$ for some $n \geq 1$. A Markov chain is aperiodic if all states are aperiodic.

#### 1.7.1.1 Absorbing Markov Chains

A state $X_i$ of a Markov chain is called *absorbing* if it is impossible to leave it (i.e., $p_{ii} = 1$). A Markov chain is absorbing if it has at least one absorbing state, and if from every state it is possible to go to an absorbing state (not necessarily in one step). In an absorbing Markov chain, a state which is not absorbing is called *transient*.

### 1.7.2 QBD Process and Matrix-Geometric Method

A Markov process $P$ with a block-tridiagonal transition matrix such as below is called a quasi-birth-death (QBD) process. $B, C, E, A_2, A_1$ and $A_0$ are matrices in $P$.

$$P = \begin{bmatrix} B & C & & & & \\ E & A_1 & A_0 & & & \\ & A_2 & A_1 & A_0 & & \\ & & A_2 & A_1 & A_0 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix} . \tag{1.10}$$

Transitions are allowed to states within neighboring blocks only. If stochastic matrix $A = A_0 + A_1 + A_2$ is irreducible then it has a stationary vector $\pi$ such that $\pi = \pi A$ , $\pi e = 1$. The stability condition is $\pi A_2 e > \pi A_0 e$. Then we have an invariant vector $x$ such that $x = xP$ and $xe = 1$.

It's shown in [55] that in such systems the stationary probability vector $x_{i+1}$ can be defined in terms of the vector $x_i$ as follows:

$$x_{i+1} = x_i R \tag{1.11}$$

where the rate matrix $R$ is a constant matrix which is the minimal non-negative solution to $R = A_0 + RA_1 + R^2 A_2$. The probability vector $x_i$ is then *matrix-geometric* and given by $x_i = x_0 R^i$.

Although the transform methods in queueing analysis provide exact results using mathematically elegant procedures, their practical use is rather limited because of the highly involved computations required to obtain numerical solutions.

Matrix-geometric method, as developed by Neuts [55] is a powerful alternative for the solution of queueing systems with highly structured multidimensional state spaces. No transform is involved in this method. Therefore, the complex process of transform inversion is avoided.

### 1.7.3 Markovian Arrival Process (MAP)

Markovian arrival process (MAP) was introduced by Neuts [56] as a generalization of the Poisson process which is unable to represent the correlation structure in a stochastic process.

Discrete-time MAP (D-MAP) is an extension of the Markov modulated Bernoulli process (MMBP). In MMBP, Bernoulli arrival is modulated by a discrete-time Markov chain. A Markovian arrival process works in a similar manner, but arrivals may also occur when the Markov process jumps from one state to another. Bernoulli process is a special type of D-MAP.

D-MAP process is defined in [57] as follows: *D-MAP is characterized by two* $m \times m$ *matrices* $D_0$ *and* $D_1$ *where the* $(j_1, j_2)^{th}$ *entry of the matrix* $D_1$ *represents the probability that a customer arrives and the underlying Markov chain makes a transition from state* $j_1$ *to state* $j_2$. *The matrix* $D_0$ *covers the case when there is no arrival. Thus, if the D-MAP is in state* $j_1$ *at time* $n$, *then, with probability* $(D_x)_{j_1,j_2}$, *we have* $x$ *arrival(s) at time* $n$ *and the state at time* $n+1$ *equals* $j_2$. *The matrix* $D$, *defined as* $D = D_0 + D_1$ *represents the stochastic* $m \times m$ *transition matrix of the underlying Markov chain of the arrival process. Let* $\psi$ *be the stationary probability vector of* $D$, *that is,* $\psi D = \psi$ *and* $\psi e = 1$, *where* $e$ *is a column vector with all entries equal to one. The stationary arrival rate is given by* $\psi = \psi D_1 e$.

### 1.7.4 Batch Markovian Arrival Process (BMAP)

MAP is extended to the batch arrival case which is called the batch Markovian arrival process (BMAP) [58]. BMAP arrival process introduced to capture the bursty nature of traffic source behavior, and therefore, well suited to model multimedia data [59].

### 1.7.5 Phase-Type Distribution

Consider an $n+1$ $(n \leq \infty)$ state absorbing Markov chain which can take on any one of the states in the set $\varepsilon = [0, 1, 2, \ldots, n]$. Let the transition matrix of this Markov chain be **P**, where

$$P = \begin{bmatrix} 1 & 0 \\ T^0 & T \end{bmatrix}. \tag{1.12}$$

The matrix $T$ is an $n \times n$ transition matrix corresponding to the transitions within the transient states, and it is sub-stochastic with $0 \leq T_{ij} \leq 1$ and the sum of at least one row of $T$ is strictly less than 1. The column vector $T^0$ is given as $T^0 = e - Te$, where $e$ is a column vector of ones with appropriate dimension. Let $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$, where $\alpha_i$ is the probability that the Markov chain starts from state $i$. We have, by definition, $\alpha_0 + \alpha e = 1$. Let the time to absorption for this Markov chain be $\tau$; then

$$\text{Prob}(\tau = i) = p_i = \alpha T^{i-1}T^0, i \geq 1 \quad \text{and} \quad p_0 = \alpha_0. \tag{1.13}$$

The time to absorption in such a Markov chain is said to have a phase-type distribution [55].

### 1.7.6 Markov Decision Process

We can describe the evolution of a stochastic system by $x_{n+1} = f(x_n, a_n)$, which is called the system equation and $x_n \in \mathcal{S}$ and $a_n \in \mathcal{A}$ denote, respectively, the system state and decision at time $n$. *Markov decision processes* are a class of Markov processes (i.e. stochastic processes with Markov property).

For each state-action pair $(x, a)$, if we let $p_a(x, y)$ represent the probability that the next state is $y$, given that the current state is $x$ and the current action is $a$, then the system dynamics can be represented by a transition probability matrix $P_a$ with $p_a(x, y)$ as its entries. Let $u(x, n)$ be a decision rule (also referred to as a *policy*) that assigns an action from the set of available actions $\mathcal{A}$ based on the current state $x$ and current time $n$.

Each policy is assessed under an objective (e.g., minimizing a cost function). Costs may be accumulated over time. In this case, cost $g_{a_n}(x_n)$ is incurred at each time $n$. An optimality criterion is required to choose the best policy. Given the cost function, a Markov decision process can be represented by $(\mathcal{S}, \mathcal{A}, P_.(., .), g_.(.))$ which consists of a state space, a set of actions, transition probabilities, and costs associated with each state-action pair.

### 1.7.7 Reinforcement Learning

Reinforcement learning [60],[61] provides a framework by which an agent can learn control policies based on experience of interacting with its environment and the rewards receives following its actions. The objective is to find a policy that maximizes these rewards or minimizes the punishments.

There are three different major learning strategies in reinforcement learning. These are dynamic programming, Monte Carlo methods and temporal difference learning.

In dynamic programming, a complete model of the environment is needed. By a model of the environment we mean anything that an agent can use to predict how the environment will respond to its actions. This model can be deterministic or stochastic. MDP is the underlying model for the case that environment model has Markov property.

Both Monte Carlo and temporal difference methods do not need a model of the environment. The difference between them is that temporal difference methods use immediate rewards and the (discounted) value of the current state to update the value of the preceding states. Monte Carlo methods, on the other hand, use samples of state transitions. When applicable, temporal difference methods may show better performance than do Monte Carlo methods [60].

In this section, we introduce MDP as the underlying concept in the model-based reinforcement learning methods. Then, dynamic programming and temporal difference methods are reviewed and a middle ground method, which is used in this work, is introduced.

### 1.7.7.1 Dynamic Programming

Dynamic programming offers a unified approach to solve stochastic decision making/control problems. Central to the methodology is the cost-to-go function, which is obtained via solving Bellman's equation. The domain of the cost-to-go function is the state space of the system to be controlled, and dynamic programming algorithms compute and store a table consisting of one cost-to-go value per state.

Given that the cost $g_{a_n}(x_n)$ is incurred at each time $n$, an infinite horizon discounted cost optimality criterion is defined as follows:

$$E\left[\sum_{n=0}^{\infty} \alpha^t g_{a_n}(x_n)|x_0\right] \tag{1.14}$$

where $\alpha \in (0,1)$ is the discount factor. Finding a policy that minimizes the infinite-horizon cost corresponds to solving the following optimization problem:

$$\min_{u(.,.)} E\left[\sum_{n=0}^{\infty} \alpha^n g_{u(x_n,n)}(x_n)|x_0\right] . \tag{1.15}$$

One approach to solve the optimization problem above is to enumerate all possible policies $u(x, n)$, evaluate the corresponding expected cost, and choose the best policy. However, the number of policies grows exponentially with the number of states and time stages. The computation required to find an optimal policy can be reduced by rewriting the above equation as follows:

$$\min_{a\in\mathcal{A}}\left\{g_a(x_0)+\sum_{y\in\mathcal{S}}p_a(x_0,y)\min_{u(.,.)}E\left[\sum_{n=1}^{\infty}\alpha^{n-1}g_{u(x_n,n)}(x_n)|x_1=y\right]\right\}. \tag{1.16}$$

We define $J(x, n_0)$ as follows:

$$J(x,n_0)=\min_{u(.,.)}E\left[\sum_{n=n_0}^{\infty}\alpha^{n-n_0}g_{u(x_n,n)}(x_n)|x_{n_0}=x\right]. \tag{1.17}$$

If $J(., n_0 + 1)$ is known, $J(x, n_0)$ can be found as follows:

$$J(x,n_0)=\min_{a\in\mathcal{A}}\left\{g_a(x)+\alpha\sum_{y\in\mathcal{S}}p_a(x,y)J(y,n_0+1)\right\}. \tag{1.18}$$

Note that, the computations required to find $J(x, n)$ grow linearly with the number of states and time stages, even though there are exponentially many policies. Moreover, for every $u$ we have $J(x, n) = J(x, n') = J(x)$ for all $n$ and $n'$. Intuitively, it is true as long as the initial state is the same, since transition probabilities $p_a(x, y)$ do not depend on time. Also, $u(x, n)$ does not depend on the current time stage $n$ since $J(x, n) = J(x)$. The policies which do not depend on the time stage are called *stationary* policies. As a result, $J$, which is also called the *cost-to-go function*, must satisfy the following equation which is called the *Bellman's equation*:

$$J(x)=\min_{a\in\mathcal{A}}\left\{g_a(x)+\alpha\sum_{y\in\mathcal{S}}p_a(x,y)J(y)\right\}. \tag{1.19}$$

Classical dynamic programming algorithms are of limited utility in reinforcement learning both because of their assumption of a perfect model and because of their great computational expense, but they are still important theoretically. In fact, other methods can be viewed as attempts to achieve much the same effect as dynamic programming, only with less computation and/or without assuming a perfect model of the environment.

### 1.7.7.2 Q-Learning

Temporal difference learning is a combination of Monte Carlo ideas and dynamic programming ideas. Like Monte Carlo methods, temporal difference methods can learn directly from raw experience without a model of the environment's dynamics. Like dynamic programming method, temporal difference methods update estimates based in part on other learned estimates, without waiting for a final outcome.

Q-learning is a popular model-free, temporal difference reinforcement technique with a strong proof of convergence established in [62]. Let $Q(x_k, a)$ be the expected discounted reinforcement of taking action $a$ in state $x_k$, then continuing by choosing actions optimally. $Q(x_k, a)$ can be written recursively at time $n$ [63]

$$Q_{n+1}(x_n, a_n) = g_{a_n}(x_n) + \alpha \sum_y p_a(x_n, y) \min_{\acute{a}} Q_n(y, \acute{a}) \tag{1.20}$$

However, when we don't know the model or it's expensive to compute it, we consider implicitly estimating the expectation above, based on state transitions observed in system trajectories. Then, $Q(x, a)$ can be obtained as follows.

$$Q_{n+1}(x_n, a_n) = g_{a_n}(x_n) + \alpha \min_{\acute{a}} Q_n(x_{n+1}, \acute{a}) \tag{1.21}$$

In fact, $Q_n(x_{n+1}, \acute{a})$ is a noisy estimate of $\sum_y p_a(x_n, y) \min_{\acute{a}} Q_n(y, \acute{a})$ and the algorithm may not be converged. To make it converge, we attenuate the noise at each step as follows where $\varrho$ is a discounting factor.

$$Q_{n+1}(x_n, a_n) = (1 - \varrho) Q_{x_n, a_n} + \varrho \left[ g_{a_n}(x_n) + \alpha \min_{\acute{a}} Q_n(x_{n+1}, \acute{a}) \right] \tag{1.22}$$

## 1.8 Chapter Summary

In this chapter, we have provided a brief introduction to wireless multimedia sensor networks. Some research issues in these networks, especially in the MAC and the physical layers have been outlined. The research works presented by this thesis have been summarized and an outline of the rest of the thesis has been given. A few mathematical preliminaries have been described to provide some of the basics on the analyses presented in this work.

# Chapter 2

# QoS and Energy Tradeoff in Distributed Energy-Limited Multi-Hop Wireless Networks

## 2.1 Introduction

A queueing analytical framework is proposed in this chapter to study the tradeoff between the energy saving and the QoS at a relay node. We use MAP and phase-type (PH) distribution to model the bursty traffic arrival process and service process respectively. Here, the relayed packets and the node's own packets form two priority classes and the MAC/physical (PHY) layer protocol in the transmission protocol stack acts as the server process. Moreover, we use a phase-type vacation model for the energy-saving mechanism in a node when the MAC/PHY protocol refrains from transmitting in order to save battery power.

Two different power saving mechanisms due to the standard exhaustive and the number-limited exhaustive vacation models (both in multiple vacation cases) are analyzed to study the tradeoff between the QoS performance of the relayed packets and the energy saving at a relay node. Also, an optimization formulation is presented to design an optimal wakeup strategy for the server process under specific QoS constraints.

We use matrix-geometric method to obtain the stationary probability distribution for the system states from which the performance metrics are derived. Using phase-type distribution for both the service and the vacation processes and combining the priority queueing model with the vacation queueing model make the analysis very general and comprehensive.

## 2.2 System Model

A node in a distributed mesh/relay[1] network is modeled as a discrete time queueing system using the theory of discrete-time Markov chains, where time is divided into fixed length intervals called slots. All events which occur between time $t$ and $t+1$ are assumed to become effective at time $t+1$. At least one epoch interval is required to serve one packet. During consecutive slots, packets arriving in a node are stored and served on a first-in first-out (FIFO) basis in two different queues. The arrivals are modeled by MAP which allows correlated and bursty arrivals.

At each node two separate queues are maintained: one for data packets relayed from other nodes which are waiting for processing and transmission to the next node in their route, and the other for the node's own packets. During transmission, the node prioritizes the relayed packets over it's own packets and packets from each queue are served on a first-in first-out (FIFO) basis.

We consider a wireless relay network with stationary nodes where each node is described by two operational modes: *active mode* and *sleep mode*. In active mode, a node is fully operational and is able to transmit/receive data, while in sleep mode it does not take part in any data transmission activity. Since most of the energy consumption in a node is primarily due to radio transmission rather than processing or even radio reception [64], in the system model under consideration, we assume that a node can receive packets while it is in the sleep mode when the transmitter circuitry is turned off. The sleep and wakeup mechanism is modeled by using a vacation queueing model.

The server is, in fact, the MAC/PHY layer protocol which executes back-off procedures, does clear channel assessments, waits for and collects acknowledgements, selects transmission rate and channel coding parameters and is responsible for retransmissions of the packets in case of transmission failure (e.g., due to collisions

[1] From now on, we will use the terms *mesh networks* and *relay networks* interchangeably in this chapter.

and/or channel impairments).

The server process (i.e., the MAC/PHY protocol) goes into the sleep mode either after both the queues have been served exhaustively or a predefined number of packets have been served. These correspond to the *standard exhaustive vacation system* and the *number-limited exhaustive vacation system*, respectively [30]. In the former, the server attends the queues until the system becomes empty (i.e. no waiting packets in the queues) and then goes to vacation (sleep mode). In the latter case, the server serves the queues until it has served $M$ packets or the queues become empty, whichever occurs first.

We include vacation (sleeping) mechanisms to the discrete-time MAP/PH/1 priority queueing model in [29] and obtain the performance measures of the MAC protocol in terms of queue length distribution, packet dropping probability, and access delay in the queue for high-priority packets as well as energy saving performance at that node. Again, we assume a non-preemptive service model where the low-priority queue in service would not be interrupted by new packet arrivals at the high-priority queue. That is, when the MAC/PHY protocol is serving a packet from the low-priority queue, it has to complete transmission of the packet before it can serve any high-priority packet.

### 2.2.1 Markovian Arrival Process (MAP) Traffic Model

Modeling the packet arrivals at a node by using a discrete Markovian arrival process (D-MAP), we allow correlation among the inter-arrival times within each type of packets and between the two types of packets (i.e., high-priority and low-priority packets, which will be referred to as type-1 and type-2 packets, respectively) as well. A MAP is described by four sub-stochastic matrices $D_0, D_{11}, D_{12}$ and $D_2$, with $D_1 = D_{11} + D_{12}$ and $D = D_0 + D_1 + D_2$, where $D$ is stochastic. The element $(D_0)_{ij}$ represents the probability of a transition from phase $i$ to phase $j$ with no arrival, $(D_{11})_{ij}$ represents a transition with an arrival of type-1 packet, $(D_{12})_{ij}$ represents a transition with an arrival of type-2 packet, and $(D_2)_{ij}$ represents a transition with two arrivals (one of each type). The arrival rate $\lambda_i$ for type-$i$ packet ($i = 1, 2$) is given as $\lambda_i = \psi(D_{1i} + D_2)e$, where $\psi$ is the solution of $\psi = \psi D$ and $\psi e = 1$ ($e$ is a column vector of 1s with appropriate dimension).

We consider one special case of this arrival process where there are two independent discrete arrival streams described by the matrices $D_0(i), D_1(i), i = 1, 2$. In this case, $D_0 = D_0(1) \otimes D_0(2), D_{11} = D_1(1) \otimes D_0(2), D_{12} = D_0(1) \otimes D_1(2)$, and $D_2 = D_1(1) \otimes D_1(2)$, where $\otimes$ is the Kronecker product.

### 2.2.2 Phase-Type Service Distribution

We model service time in the node as a phase-type distribution. The service process is determined by the MAC/PHY protocol activity (i.e., active mode, sleep mode) and the corresponding mechanisms (e.g., channel access mechanism, handshaking protocols etc.). The type-1 and type-2 packets are stored in two separate queues (Fig. 2.1). In the active mode, the service process prioritizes type-1 packets over type-2 packets. We study the non-preemptive priority case here in which there is no service interruption upon arrival of a type-1 packet when a type-2 packet is being serviced. Packets from each queue are served on a FIFO basis.

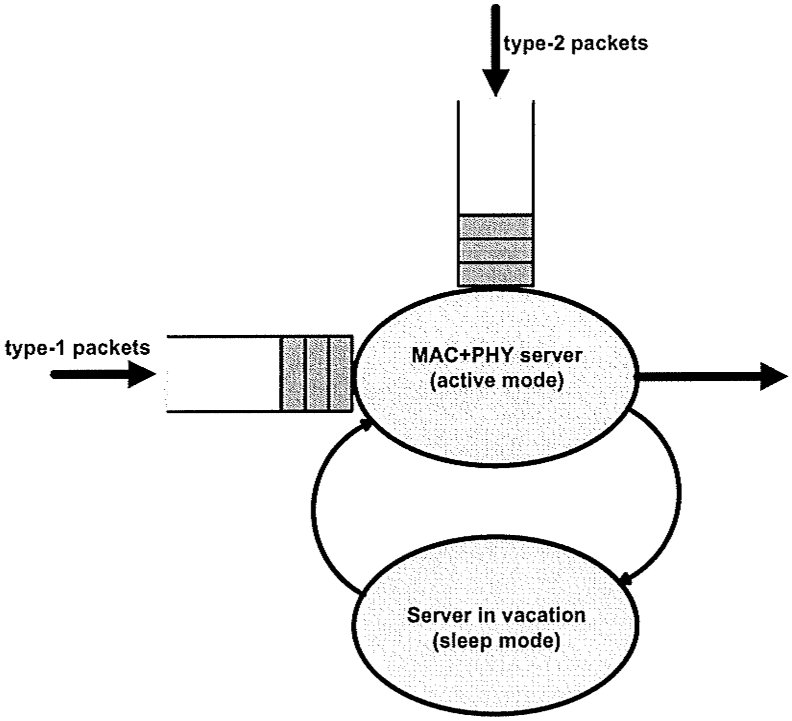


**Figure 2.1.** *Queueing model at a node: MAC/PHY layer as the server.*

The service time for the head-of-line (HOL) packet follows a discrete phase-type distribution represented by $(\beta(i), S(i))$, $i = 1, 2$ for packet type $i$. In this representation, $\beta(i)$ is a row vector, $S(i)$ is a sub-stochastic matrix, and $S^0(i) = e - S(i)e$. Note that, in a practical multi-hop wireless relay network, the service time would account for the delay due to medium access control and link-level error control.

A single server vacation queue with two arrivals and non-batch service is considered in discrete time. For the service process, the distribution of time spent in vacation (i.e., in sleep mode) is also assumed to phase-type with parameters ($\nu$, $V$). To make the system more energy efficient, we assume a multiple vacation model, that is, upon going from the vacation to the active mode, if the server finds the queues empty, it returns to the vacation mode immediately.

We consider two types of vacation systems–standard exhaustive vacation system and number-limited (NL) exhaustive vacation system [30]. In the former case, the server goes to the sleep mode after serving all packets from both of the queues. In the latter case, the server process goes to the sleep mode after it has served $M$ packets or the queues have become empty, whichever occurs first.

## 2.3 Analytical Model for the Standard Exhaustive Vacation Case

In this case, when the server moves from the vacation to the active mode, all the packets from both of the queues are served. The server goes to the vacation mode when the queues are empty. In this section, a Markov model is developed and then analyzed using matrix-geometric method to obtain the steady state probability distribution from which the performance metrics are derived.

### 2.3.1 The Markov Chain

Consider the following state space:

$$\begin{aligned}
&\Omega_0^v = (0, 0, j, l)\\
&\Omega_0^s = (0, i_2, j, k_2), i_2 \geq 1\\
&\Omega^v = (i_1, i_2, , j, l), (i_1 + i_2) \geq 1\\
&\Omega_1^s = (i_1, i_2, j, k_1), i_1 \geq 1; i_2 \geq 0\\
&\Omega_2^s = (i_1, i_2 + 1, j, k_2), i_1 \geq 1; i_2 \geq 0.
\end{aligned} \tag{2.1}$$

Here, $\Omega_0^v$ represents the vacation (i.e., sleep mode) while the queues are empty with the arrival process in phase $j$ and the vacation process in phase $l$. In $\Omega_0^s$, there are only type-2 packets in the system with the arrival process in phase $j$ and a type-2 packet is being served while the service process is in phase $k_2$. In states $\Omega^v$, $\Omega_1^s$ and $\Omega_2^s$, there is at least one type-1 packet in the system. In $\Omega^v$, the server is in vacation (i.e., sleep mode) while the phases of arrival and vacation processes are $j$ and $l$, respectively. $\Omega_1^s$ and $\Omega_2^s$ represent the system in service, where in $\Omega_1^s$ the packet in the server is of type-1 with $k_1$ as the phase of service, and in $\Omega_2^s$ the packet in the server is of type-2 with $k_2$ as the phase of service. The state space for the system is

$$\Omega = \Omega_0^v \bigcup \Omega_0^s \bigcup \Omega^v \bigcup \Omega_1^s \bigcup \Omega_2^s. \tag{2.2}$$

The transition matrix **P** describing this Markov chain has the following form:

$$P = \begin{bmatrix}
B_{00} & B_{01} & & & \\
B_{10} & A_1 & A_0 & & \\
 & A_2 & A_1 & A_0 & \\
 & & A_2 & A_1 & A_0 \\
 & & & \ddots & \ddots & \ddots
\end{bmatrix}. \tag{2.3}$$

All the blocks $B_{00}, B_{01}, B_{10}, B_{11}, A_0, A_1, A_2$ are of infinite dimensions and are described as follows. $B_{00}$ represents the case where the queue for high-priority packets remains empty after the system state transition.

$$B_{00} = \begin{bmatrix} B_{00}^{00} & B_{00}^{01} & & & \\ B_{00}^{10} & B_{00}^{1} & B_{00}^{0} & & \\ & B_{00}^{2} & B_{00}^{1} & B_{00}^{0} & \\ & & B_{00}^{2} & B_{00}^{1} & B_{00}^{0} \\ & & & \ddots & \ddots & \ddots \end{bmatrix} \tag{2.4}$$

Submatrices in $B_{00}$ represent changes in low-priority queue when high-priority queue remains empty. In $B_{00}^{00}$ low-priority queue remains empty and in $B_{00}^{01}$ it will be non-empty by a type-2 packet arrival.

$$B_{00}^{00} = D_0 \otimes (V + V^0\nu) \qquad B_{00}^{01} = \begin{bmatrix} D_{12} \otimes V & D_{12} \otimes V^0\beta(2) \end{bmatrix}$$

In $B_{00}^{10}$, a non-empty low-priority queue becomes empty and since high-priority queue is also empty, server takes a vacation. In $B_{00}^{0}$ not only no packet is served, but also a new packet arrival occurs.

$$B_{00}^{10} = \begin{bmatrix} 0 \\ D_0 \otimes S^0(2)\nu \end{bmatrix} \qquad B_{00}^{0} = \begin{bmatrix} D_{12} \otimes V & D_{12} \otimes V^0\beta(2) \\ 0 & D_{12} \otimes S(2) \end{bmatrix}$$

$B_{00}^{1}$ is the case that the number of packets in low-priority queue is not changed. This happens either if no type-2 packets is served and no arrival of this type of packet occurs or if server serves a packet of this type and a new packet of the same type arrives in the queue. For $B_{00}^{2}$, serving a packet in low-priority queue and no arrival in this queue, decreases the number of packets in the queue. However, queue will not be empty.

$$B_{00}^{1} = \begin{bmatrix} D_0 \otimes V & D_0 \otimes V^0\beta(2) \\ 0 & D_0 \otimes S(2) + D_{12} \otimes S^0(2)\beta(2) \end{bmatrix} \quad B_{00}^{2} = \begin{bmatrix} 0 & 0 \\ 0 & D_0 \otimes S^0(2)\beta(2) \end{bmatrix}$$

In $B_{01}$, high-priority queue becomes non-empty with an arrival of type-1 packet while it was empty in previous time slot.

$$B_{01} = \begin{bmatrix} B_{01}^{00} & B_{01}^{01} & & & \\ B_{01}^{2} & B_{01}^{1} & B_{01}^{0} & & \\ & B_{01}^{2} & B_{01}^{1} & B_{01}^{0} & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \tag{2.5}$$

$B_{01}^{00}$ and $B_{01}^{01}$ in $B_{01}$ represent the case that low-priority queue has been empty and becomes empty and non-empty respectively. In other words, we have just a type-1 arrival in $B_{01}^{00}$. However, in $B_{01}^{01}$, both type-1 and type-2 arrivals occur.

$$B_{01}^{00} = \begin{bmatrix} D_{11} \otimes V & D_{11} \otimes V^0\beta(1) & 0 \end{bmatrix} \qquad B_{01}^{01} = \begin{bmatrix} D_2 \otimes V & D_2 \otimes V^0\beta(1) & 0 \end{bmatrix}$$

$B_{01}^{0}$ is similar to $B_{01}^{01}$. However, low-priority queue has not been empty in $B_{01}^{0}$ before new packet arrival. In $B_{01}^{2}$, a type-2 packet is served and low-priority queue may become empty due to no arrival of this type of packet. However, an arrival in high-priority queue keeps the server busy and node stays in active mode.

$$B_{01}^{0} = \begin{bmatrix} D_2 \otimes V & D_2 \otimes V^0\beta(1) & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B_{01}^{2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_{11} \otimes S^0(2)\beta(1) & D_{11} \otimes S(2) \end{bmatrix}$$

$B_{01}^{1}$ is similar to $B_{00}^{1}$ in which the state of the low-priority queue remains unchanged. However, high-priority queue is not empty in $B_{01}^{1}$.

$$B_{01}^{1} = \begin{bmatrix} D_{11} \otimes V & D_{11} \otimes V^0\beta(1) & 0 \\ 0 & D_2 \otimes S^0(2)\beta(1) & D_2 \otimes S(2) \end{bmatrix}$$

For $B_{10}$, a non-empty high-priority queue becomes empty when its last packet is served.

$$B_{10} = \begin{bmatrix} B_{10}^{00} & B_{10}^{0} & & \\ & B_{10}^{1} & B_{10}^{0} & \\ & & B_{10}^{1} & B_{10}^{0} \\ & & & \ddots & \ddots \end{bmatrix} \tag{2.6}$$

Since there is no type-2 packet arrival in already empty low-priority queue in $B_{10}^{00}$, server takes a vacation. On the other hand, an arrival in low-priority queue in $B_{10}^{0}$, keeps the server busy.

$$B_{10}^{00} = \begin{bmatrix} 0 \\ D_0 \otimes S^0(1)\nu \\ 0 \end{bmatrix}, \; B_{10}^{0} = \begin{bmatrix} 0 & 0 \\ 0 & D_{12} \otimes S^0(1)\beta(2) \\ 0 & 0 \end{bmatrix}$$

For $B_{10}^{1}$, since there are type-2 packets to be served, server remains in active mode to start serving them.

$$B_{10}^{1} = \begin{bmatrix} 0 & 0 \\ 0 & D_0 \otimes S^0(1)\beta(2) \\ 0 & 0 \end{bmatrix}$$

$A_0$ represents an increment in the number of type-1 packets in high-priority queue.

$$A_0 = \begin{bmatrix} A_0^1 & A_0^0 & & \\ & A_0^1 & A_0^0 & \\ & & \ddots & \ddots \end{bmatrix} \tag{2.7}$$

In $A_0^0$, we have an arrival in low-priority queue beside type-1 arrival. However, no type-2 arrival occurs in $A_0^1$.

$$A_0^0 = \begin{bmatrix} D_2 \otimes V & D_2 \otimes V^0\beta(1) & 0 \\ 0 & D_2 \otimes S(1) & 0 \\ 0 & D_2 \otimes S^0(2)\beta(1) & D_2 \otimes S(2) \end{bmatrix}$$

$$A_0^1 = \begin{bmatrix} D_{11} \otimes V & D_{11} \otimes V^0\beta(1) & 0 \\ 0 & D_{11} \otimes S(1) & 0 \\ 0 & D_{11} \otimes S^0(2)\beta(1) & D_{11} \otimes S(2) \end{bmatrix}$$

$A_1$ shows the case when no change is made in the number of type-1 packets in the queue at the end of current time slot comparing to previous time slot.

$$A_1 = \begin{bmatrix} A_1^1 & A_1^0 & & \\ & A_1^1 & A_1^0 & \\ & & \ddots & \ddots \end{bmatrix} \tag{2.8}$$

$A_1^0$ and $A_1^1$ are the same as $A_0^0$ and $A_0^1$ respectively in low-priority queue's perspective.

$$A_1^0 = \begin{bmatrix} D_{12} \otimes V & D_{12} \otimes V^0\beta(1) & 0 \\ 0 & D_{12} \otimes S(1) + D_2 \otimes S^0(1)\beta(1) & 0 \\ 0 & D_{12} \otimes S^0(2)\beta(1) & D_{12} \otimes S(2) \end{bmatrix}$$

$$A_1^1 = \begin{bmatrix} D_0 \otimes V & D_0 \otimes V^0\beta(1) & 0 \\ 0 & D_0 \otimes S(1) + D_{11} \otimes S^0(1)\beta(1) & 0 \\ 0 & D_0 \otimes S^0(2)\beta(1) & D_0 \otimes S(2) \end{bmatrix}$$

In $A_2$, number of high-priority packets is decreased by one.

$$A_2 = \begin{bmatrix} A_2^1 & A_2^0 & & \\ & A_2^1 & A_2^0 & \\ & & \ddots & \ddots \end{bmatrix} \tag{2.9}$$

$A_2^0$ and $A_2^1$ simply represent one arrival and no arrival in low-priority queue respectively while a type-1 packet has been served and server continuing serving high-priority queue.

$$A_2^0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_{12} \otimes S^0(1)\beta(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_2^1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_0 \otimes S^0(1)\beta(1) & 0 \\ 0 & & 0 \end{bmatrix}.$$

### 2.3.2 Matrix-Geometric Analysis and Steady State Distribution

The matrix $\mathbf{P}$ is in the form of a QBD process while its blocks are of infinite dimension. The first step in analyzing the system using matrix-geometric method [55] is to determine the matrix $R$ which is the minimal non-negative solution to the following matrix quadratic equation:

$$R = A_0 + RA_1 + R^2A_2. \tag{2.10}$$

For the QBD case, the matrix $R$ has the following form [29]:

$$R = \begin{bmatrix} R_0 & R_1 & R_2 & R_3 & R_4 & \dots \\ & R_0 & R_1 & R_2 & R_3 & \dots \\ & & R_0 & R_1 & R_2 & \dots \\ & & & \ddots & \ddots & \ddots \end{bmatrix} \tag{2.11}$$

and the blocks of matrices $R_i$ $(i \geq 0)$ are given as follows:

$$\begin{aligned} R_0 &= A_0^1 + R_0A_1^1 + R_0^2A_2^1 && (2.12) \\ R_1 &= A_0^0 + R_0A_1^0 + R_1A_1^1 + R_0^2A_2^0 + (R_0R_1 + R_1R_0)A_2^1 && (2.13) \\ R_j &= R_{j-1}A_1^0 + R_jA_1^1 + \sum_{k=0}^{j-1} R_{j-k-1}R_kA_2^0 + \sum_{k=0}^{j} R_{j-k}R_kA_2^1. && (2.14) \end{aligned}$$

The matrix blocks $R_i$ $(i \geq 0)$ can be further partitioned as follows:

$$R_i = \begin{bmatrix} R_i(vv) & R_i(vs_1) & 0 \\ 0 & R_i(s_1s_1) & 0 \\ 0 & R_i(s_2s_1) & R_i(s_2s_2) \end{bmatrix}. \tag{2.15}$$

Since $(R_{k+1})_{ij} \leq (R_k)_{ij}$ $(\forall k \geq 1, \forall i, j)$ [29], we can compute $R_0, R_1, \cdots$ recursively up to the point of truncation.

Let $x = [x_0\ x_1\ x_2\ \cdots]$ be the steady state probability vector corresponding to $\mathbf{P}$, where $x = xP$, and $xe = 1$. It can be shown that [55]

$$x_{i+1} = x_iR, i \geq 1 \tag{2.16}$$

where

$$x_i = [x_{i0}\ x_{i1}\ x_{i2}\ \cdots], i \geq 0, x_{ij} = [x^v_{ij}\ x^s_{ij}(1)\ x^s_{i,j+1}(2)], i \geq 1, j \geq 0. \tag{2.17}$$

$x^v_{ij}, x^s_{ij}(1)$, and $x^s_{i,j+1}(2)$ are the probabilities that there are $i$ type-1 and $j$ type-2 packets in the node and server is in vacation, serving high-priority, or serving low-priority queue, respectively.

The probabilities of increasing in the number of packets in high-priority queue for three different cases are as follows.

$$\begin{aligned} x^v_{i+1,j} &= \sum_{k=0}^{j} x^v_{i,j-k} R_k(vv) && (2.18) \\ x^s_{i+1,j}(1) &= \sum_{k=0}^{j} [x^v_{i,j-k} R_k(vs_1) + x^s_{i,j-k}(1) R_k(s_1 s_1) \\ &\quad + x^s_{i,j-k+1}(2) R_k(s_2 s_1)], \qquad i \geq 1, j \geq 0 && (2.19) \\ x^s_{i+1,j+1}(2) &= \sum_{k=0}^{j} x^s_{i,j-k+1}(2) R_{j-k}(s_2 s_2), i \geq 1, j \geq 0. && (2.20) \end{aligned}$$

We can calculate $[x_0\ x_1]$ as follows:

$$\begin{bmatrix} x_0 & x_1 \end{bmatrix} = \begin{bmatrix} x_0 & x_1 \end{bmatrix} \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & A_1 + RA_2 \end{bmatrix} \tag{2.21}$$

Then

$$x_1 = x_0 B_{01} + x_1(A_1 + RA_2) \tag{2.22}$$

which implies

$$x_1 = x_0 B_{01}(I - A_1 - RA_2)^{-1}. \tag{2.23}$$

The vector $x_0$ is then normalized by

$$x_0 e + x_0[B_{01}(I - A_1 - RA_2)^{-1}(I - R)^{-1}]e = 1. \tag{2.24}$$

The inverses of the matrices in (2.23) and (2.24) exist and can be calculated recursively since they are both upper triangular with the following form:

$$U = \begin{bmatrix} U_0 & U_1 & U_2 & U_3 & U_4 & \dots \\ & U_0 & U_1 & U_2 & U_3 & \dots \\ & & U_0 & U_1 & U_2 & \dots \\ & & & U_0 & U_1 & \dots \\ & & & & \ddots & \ddots \\ & & & & & \ddots \end{bmatrix}. \tag{2.25}$$

If $V$ denotes the inverse of matrix $U$, then $V$ is given as

$$V = \begin{bmatrix} V_0 & V_1 & V_2 & V_3 & V_4 & \dots \\ & V_0 & V_1 & V_2 & V_3 & \dots \\ & & V_0 & V_1 & V_2 & \dots \\ & & & V_0 & V_1 & \dots \\ & & & & \ddots & \ddots \\ & & & & & \ddots \end{bmatrix} \tag{2.26}$$

where

$$V_0 = U_0^{-1} \quad \text{and} \quad V_n = -U_0^{-1} \sum_{j=0}^{n-1} U_{n-j} V_j, \quad n \geq 1. \tag{2.27}$$

To calculate $x_0$, we have the following equation in addition to the normalization equation:

$$\begin{aligned} x_0 &= x_0 B_{00} + x_1 B_{10} \\ &= x_0 B_{00} + x_0 B_{01} (I - A_1 - R A_2)^{-1} B_{10}. \end{aligned} \tag{2.28}$$

This equation above has the following structure:

$$x_0 = x_0 \begin{bmatrix} F_0 & F_1 & F_2 & F_3 & F_4 & \dots \\ J_0 & H_1 & H_2 & H_3 & H_4 & \dots \\ & H_0 & H_1 & H_2 & H_3 & \dots \\ & & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{2.29}$$

where

$$\bar{F}_v = \sum_{i=v}^{\infty} F_i G^{i-v} \quad \text{and} \quad \bar{H}_v = \sum_{i=v}^{\infty} H_i G^{i-v}, v \geq 0 \tag{2.30}$$

and $G$ is the minimal non-negative solution of the non-linear matrix equation

$$G = \sum_{i=0}^{\infty} A_i G^i. \tag{2.31}$$

The vectors $x_i, i \geq 1$ are given by the following recursion formula [65]:

$$x_i = [x_0 \bar{F}_i + \sum_{j=1}^{i-1} x_j \bar{H}_{i+1-j}](I - \bar{H}_1)^{-1}, \quad i \geq 1. \tag{2.32}$$

The computation of (2.32) can be performed efficiently by noting that as $i \longrightarrow \infty$, $\bar{F}_i, \bar{H}_i \longrightarrow 0$. Therefore, one may choose a large index $i$, set $\bar{F}_i = \bar{H}_i = 0$ and compute the other required matrices to compute the following backward recursion: $\bar{F}_k = F_k + \bar{F}_{k+1}G$ and $\bar{H}_k = H_k + \bar{H}_{k+1}G$. The equations $G = H_0 + \bar{H}_1 G$ and $K = F_0 + \bar{F}_1 G$ can be used to confirm that the truncation index has been chosen to be a sufficiently large value.

### 2.3.3 Performance Measures

#### 2.3.3.1 Queue Length Distribution

Let $q_i$ be the probability that there are $i$ high-priority (i.e., type-1) packets in the system. Then, we have $q_i = x_i e$ and the mean number of high-priority packets in the system is as follows.

$$L = x_1 (I - R)^{-2} e. \tag{2.33}$$

#### 2.3.3.2 Energy Saving Factor

This is defined as the probability that the server is in the sleep mode and is given by

$$S = \sum_{i=0}^{\infty} x_i^v. \tag{2.34}$$

#### 2.3.3.3 Packet Loss Probability

Having the steady state probability vector $x_i$, for a maximum queue length of $L$ packets for high-priority packets, the probability of packet loss ($p_l$) can be found as

$$p_l = \sum_{i=L+1}^{\infty} x_i e. \tag{2.35}$$

#### 2.3.3.4 Distribution of Waiting Time (or Access Delay)

The distribution of access delay[2] for high-priority packets is obtained here by using the concept of an absorbing Markov chain. The absorbing state in this case is the state where the target packet arrives at the head of the queue. This is the point where the target packet is ready to be transmitted.

When a high-priority packet arrives, it may find the system to be either empty or in a state where a high-priority or a low-priority packet is being served. Since this is a non-preemptive case, the arriving high-priority packet would have to wait for the completion of processing of all the high-priority packets ahead of it and the completion of the processing of the low-priority packet, if any, which is being served at the time of its arrival.

We first find out the state probability vectors $z_{ij}^s(j+1)$ $(i \geq 1, j = 0, 1)$ representing the probability of an arriving high-priority packet finding $i$ high-priority (i.e., type-1) packets ahead of it and a type-$(j+1)$ packet in processing.

[2] This is the time required for a packet to arrive at the head of the queue since its arrival into the queue.

$$\begin{aligned} z^s_{i0}(1) &= \lambda_1^{-1}\sum_{j=0}^{\infty} x^s_{ij}(1)[(D_{11}+D_2)\otimes S(1)] \\ &+\lambda_1^{-1}\sum_{j=0}^{\infty} x^s_{i,j+1}(2)[(D_{11}+D_2)\otimes S^0(2)\beta(1)] \\ &+\lambda_1^{-1}\sum_{j=0}^{\infty} x^s_{i+1,j}(1)[(D_{11}+D_2)\otimes S^0(1)\beta(1)] \\ &+\lambda_1^{-1}\sum_{j=0}^{\infty} x^v_{ij}[(D_{11}+D_2)\otimes V^0\beta(1)] \qquad (2.36) \\ z^s_{i1}(2) &= \lambda_1^{-1}\sum_{j=1}^{\infty} x^s_{ij}(2)[(D_{11}+D_2)\otimes S(2)] \qquad (2.37) \end{aligned}$$

Let $z^v_{ij}$ denote the same probability as above except that upon arrival the packet finds the server in sleep mode.

$$z^v_{i0} = \lambda_1^{-1}\sum_{j=0}^{\infty} x^v_{ij}[(D_{11}+D_2)\otimes V] \qquad (2.38)$$

Also, $z^s_{01}$ denotes the probability of having no type-1 packet and at least one type-2 packet in the system and $z^s_{00}$ denotes the probability of having the system empty.

$$\begin{aligned} z^s_{01} &= \lambda_1^{-1}\sum_{j=1}^{\infty} x^s_{0j}[(D_{11}+D_2)\otimes S(2)] \qquad (2.39) \\ z^s_{00} &= \lambda_1^{-1}\sum_{j=1}^{\infty} x^s_{0j}[(D_{11}+D_2)\otimes S^0(2)] \\ &+\lambda_1^{-1}\sum_{j=0}^{\infty} x^s_{1j}(1)[(D_{11}+D_2)\otimes S^0(1)] \qquad (2.40) \end{aligned}$$

Moreover, $z^v_0$ denotes the probability for a type-1 packet, upon its arrival, to find the system empty and the server on vacation.

$$\begin{aligned} z^v_0 &= \lambda_1^{-1}x_{00}[(D_{11}+D_2)\otimes(V+V^0\nu)] \\ &+\lambda_1^{-1}\sum_{j=1}^{\infty} x^v_{0j}[(D_{11}+D_2)\otimes(V+V^0\nu)] \qquad (2.41) \end{aligned}$$

We then have

$$\begin{aligned} z_0 &= [z_0^v \; z_{00}^s \; z_{01}^s] \\ z_i &= [z_{i0}^v \; z_{i0}^s(1) \; z_{i1}^s(2)], \quad i \geq 1 \\ z &= [z_0 \quad z_1 \quad \cdots]. \end{aligned} \tag{2.42}$$

We obtain the distribution of access delay for the high-priority packet as the time to absorption for a Markov chain with the following transition matrix:

$$\widetilde{P} = \begin{bmatrix} \widetilde{B}_{00} & & & \\ \widetilde{B}_{10} & \widetilde{A}_1 & & \\ & \widetilde{A}_2 & \widetilde{A}_1 & \\ & & \ddots & \ddots \end{bmatrix} \tag{2.43}$$

where $\widetilde{B}_{00}$ represent the case that the tagged type-1 packet finds no packet ahead of it. In $\widetilde{B}_{10}$, the last type-1 packet ahead of it is served.

$$\widetilde{B}_{00} = \begin{bmatrix} V & V^0 & 0 \\ 0 & I & 0 \\ 0 & I \otimes S^0(2) & I \otimes S(2) \end{bmatrix}, \quad \widetilde{B}_{10} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I \otimes S^0(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

In $\widetilde{A}_1$, there is no change in the number of packets ahead of tagged packet. On the other hand, $\widetilde{A}_2$ represents the case that the number of those packets has decreased by 1.

$$\widetilde{A}_1 = \begin{bmatrix} I \otimes V & I \otimes V^0\beta(1) & 0 \\ 0 & I \otimes S(1) & 0 \\ 0 & I \otimes S^0(2)\beta(1) & I \otimes S(2) \end{bmatrix}, \quad \widetilde{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I \otimes S^0(1)\beta(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then

$$z^{n+1} = z^n \widetilde{P}. \tag{2.44}$$

The following set of equations is used in the computation:

$$z_0 = z_0 \widetilde{B}_{00} + z_1 \widetilde{B}_{10} \tag{2.45}$$

$$z_i = z_i \widetilde{A}_1 + z_{i+1} \widetilde{A}_2 \tag{2.46}$$

$z_0^v, z_{00}^s$, and $z_{01}^s$ in (2.42) are obtained as follows.

$$z_0^v = z_0^v V \tag{2.47}$$

$$z_{00}^s = z_{00}^s + z_{01}^s (I \otimes S^0(2)) + z_0^v V^0 + z_{10}^s (I \otimes S^0(1)) \tag{2.48}$$

$$z_{01}^s = z_{01}^s (I \otimes S(2)) \tag{2.49}$$

Similarly, $z_{i0}^v, z_{i0}^s(1)$, and $z_{i1}^s(2)$ in (2.42) are obtained as follows.

$$z_{i0}^v = z_{i0}^v (I \otimes V) \tag{2.50}$$

$$\begin{aligned} z_{i0}^s(1) &= z_{i0}^v (I \otimes V^0 \beta(1)) + z_{i0}^s(1)(I \otimes S(1)) \\ &+ z_{i1}^s(2)(I \otimes S^0(2)\beta(1)) + z_{i+1,0}^s(1)(I \otimes S^0(1)\beta(1)) \end{aligned} \tag{2.51}$$

$$z_{i1}^s(2) = z_{i1}^s(2)(I \otimes S(2)). \tag{2.52}$$

Let $W_T$ be the probability that the access delay of a high-priority packet is less than or equal to $T$. Then $W_T = z_0^T e$.

## 2.4 Analytical Model for the Number-Limited Exhaustive Vacation Case

In this case the server process goes to vacation mode after serving a total of $M$ packets (even if there are more packets in the system waiting to be transmitted) or when the queues become empty, whichever occurs first. If there are fewer than $M$ high-priority packets, the server will serve low-priority packets as well (to have a total of $M$ packets if there are sufficient number of packets in the system).

### 2.4.1 The Markov Chain

The state space in this case is as follows:

$$\begin{aligned}
\Omega_0^v &= (0,0,j,l) \\
\Omega_0^s &= (0,i_2,u,j,k_2), i_2 \geq 1 \\
\Omega^v &= (i_1,i_2,j,l), (i_1+i_2) \geq 1 \\
\Omega_1^s &= (i_1,i_2,u,j,k_1), i_1 \geq 1; i_2 \geq 0 \\
\Omega_2^s &= (i_1,i_2+1,u,j,k_2), i_1 \geq 1; i_2 \geq 0 \qquad (2.53)
\end{aligned}$$

where $u$ in $\Omega_0^s, \Omega_1^s$, and $\Omega_2^s$ denotes the number of served packets ($u = 0, 1, 2, \cdots, M$). Note that, the state space is the same as the standard exhaustive case in Section 2.3 except that the number of served packets is different. The vacation states are exactly the same as before.

The transition matrix $\mathbf{P}$ describing this Markov chain is the same as (2.3). However, the matrix blocks are different which are given as follows , where $e$ is a column vector of 1s whose length is $M$ and $e_j$ is a column vector of length $M$ containing all zeros except a 1 as its $j^{th}$ element. Here, $\acute{e}$ and $\acute{e}_j$ are obtained by transposing the vectors $e$ and $e_j$, respectively.

Submatrices in $B_{00}$ have the same meaning as their counterparts in (2.4) and are given as follows.

$$B_{00}^{00} = D_0 \otimes (V + V^0\nu), \quad B_{00}^{01} = \begin{bmatrix} D_{12} \otimes V & \acute{e}_1 \otimes D_{12} \otimes V^0\beta(2) \end{bmatrix}$$

$$B_{00}^{10} = \begin{bmatrix} 0 \\ e \otimes D_0 \otimes S^0(2)\nu \end{bmatrix}, \quad B_{00}^{0} = \begin{bmatrix} D_{12} \otimes V & \acute{e}_1 \otimes D_{12} \otimes V^0\beta(2) \\ 0 & I(M) \otimes D_{12} \otimes S(2) \end{bmatrix}$$

$$B_{00}^{1} = \begin{bmatrix} D_0 \otimes V & \acute{e}_1 \otimes D_0 \otimes V^0\beta(2) \\ e_M \otimes D_{12} \otimes S^0(2)\nu & I(M) \otimes D_0 \otimes S(2) + \bar{I}(M-1) \otimes D_{12} \otimes S^0(2)\beta(2) \end{bmatrix}$$

where

$$\bar{I}(M) = \begin{bmatrix} 0 & I(M) \\ 0 & 0 \end{bmatrix}$$

$$B_{00}^{2} = \begin{bmatrix} 0 & 0 \\ e_M \otimes D_0 \otimes S^0(2)\nu & \bar{I}(M-1) \otimes D_0 \otimes S^0(2)\beta(2) \end{bmatrix}$$

Submatrices in $B_{01}$ have the same meaning as their counterparts in (2.5) and are given as follows.

$$B_{01}^{00} = \begin{bmatrix} D_{11} \otimes V & \acute{e}_1 \otimes D_{11} \otimes V^0\beta(1) & 0 \end{bmatrix}$$

$$B_{01}^{01} = \begin{bmatrix} D_2 \otimes V & \acute{e}_1 \otimes D_2 \otimes V^0\beta(1) & 0 \end{bmatrix}$$

$$B_{01}^{0} = \begin{bmatrix} D_2 \otimes V & \acute{e}_1 \otimes D_2 \otimes V^0\beta(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_{01}^{1} = \begin{bmatrix} D_{11} \otimes V & \acute{e}_1 \otimes D_{11} \otimes V^0\beta(1) & 0 \\ e_M \otimes D_2 \otimes S^0(2)\nu & \bar{I}(M-1) \otimes D_2 \otimes S^0(2)\beta(1) & I(M) \otimes D_2 \otimes S(2) \end{bmatrix}$$

$$B_{01}^{2} = \begin{bmatrix} 0 & 0 & 0 \\ e_M \otimes D_{11} \otimes S^0(2)\nu & \bar{I}(M-1) \otimes D_{11} \otimes S^0(2)\beta(1) & I(M) \otimes D_{11} \otimes S(2) \end{bmatrix}$$

For submatrices in $B_{10}$, block matrices are given as follows and have the same meaning as their counterparts in (2.6).

$$B_{10}^{00} = \begin{bmatrix} 0 \\ e \otimes D_0 \otimes S^0(1)\nu \\ 0 \end{bmatrix}$$

$$B_{10}^{0} = \begin{bmatrix} 0 & 0 \\ e_M \otimes D_{12} \otimes S^0(1)\nu & \bar{I}(M-1) \otimes D_{12} \otimes S^0(1)\beta(2) \\ 0 & 0 \end{bmatrix}$$

$$B_{10}^{1} = \begin{bmatrix} 0 & 0 \\ e_M \otimes D_0 \otimes S^0(1)\nu & \bar{I}(M-1) \otimes D_0 \otimes S^0(1)\beta(2) \\ 0 & 0 \end{bmatrix}$$

Submatrices in $A_0$ are given as follows and have the same meaning as their counterparts in (2.7).

$$A_0^0 = \begin{bmatrix} D_2 \otimes V & e_1' \otimes D_2 \otimes V^0\beta(1) & 0 \\ 0 & I(M) \otimes D_2 \otimes S(1) & 0 \\ e_M \otimes D_2 \otimes S^0(2)\nu & \bar{I}(M-1) \otimes D_2 \otimes S^0(2)\beta(1) & I(M) \otimes D_2 \otimes S(2) \end{bmatrix}$$

$$A_0^1 = \begin{bmatrix} D_{11} \otimes V & e_1' \otimes D_{11} \otimes V^0\beta(1) & 0 \\ 0 & I(M) \otimes D_{11} \otimes S(1) & 0 \\ e_M \otimes D_{11} \otimes S^0(2)\nu & \bar{I}(M-1) \otimes D_{11} \otimes S^0(2)\beta(1) & I(M) \otimes D_{11} \otimes S(2) \end{bmatrix}$$

Submatrices in $A_1$ have the same meaning as their counterparts in (2.8) and are given as follows.

$$A_1^0 = \begin{bmatrix} D_{12} \otimes V & e_1' \otimes D_{12} \otimes V^0\beta(1) & 0 \\ e_M \otimes D_2 \otimes S^0(1)\nu & A_1^0(2,2) & 0 \\ e_M \otimes D_{12} \otimes S^0(2)\nu & \bar{I}(M-1) \otimes D_{12} \otimes S^0(2)\beta(1) & I(M) \otimes D_{12} \otimes S(2) \end{bmatrix}$$

where

$$A_1^0(2,2) = I(M) \otimes D_{12} \otimes S(1) + \bar{I}(M-1) \otimes D_2 \otimes S^0(1)\beta(1). \qquad (2.54)$$

$$A_1^1 = \begin{bmatrix} D_0 \otimes V & e_1' \otimes D_0 \otimes V^0\beta(1) & 0 \\ e_M \otimes D_{11} \otimes S^0(1)\nu & A_1^1(2,2) & 0 \\ e_M \otimes D_0 \otimes S^0(2)\nu & \bar{I}(M-1) \otimes D_0 \otimes S^0(2)\beta(1) & I(M) \otimes D_0 \otimes S(2) \end{bmatrix}$$

where

$$A_1^1(2,2) = I(M) \otimes D_0 \otimes S(1) + \bar{I}(M-1) \otimes D_{11} \otimes S^0(1)\beta(1) \qquad (2.55)$$

Submatrices in $A_2$ have the same meaning as their counterparts in (2.9) and are given as follows.

$$A_2^0 = \begin{bmatrix} 0 & 0 & 0 \\ e_M \otimes D_{12} \otimes S^0(1)\nu & \bar{I}(M-1) \otimes D_{12} \otimes S^0(1)\beta(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} 0 & & 0 \\ e_M \otimes D_0 \otimes S^0(1)\nu & \bar{I}(M-1) \otimes D_0 \otimes S^0(1)\beta(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

### 2.4.2 Matrix-Geometric Analysis and Steady State Distribution

The steady state distributions of the packets in the queues for the number-limited exhaustive vacation case are obtained in the same way as that for the standard exhaustive vacation case using the new matrix blocks in **P**.

### 2.4.3 Performance Measures

The queue length distribution, probability of packet loss, and energy saving factor are obtained in the same way as that for the standard case. For delay distribution we proceed in a similar way with the new matrix blocks. This results in the following probability vectors where $D' = D_{11} + D_2$.

$$\begin{aligned} z_0^v &= \lambda_1^{-1} x_{00}[D' \otimes (V + V^0\nu)] + \lambda_1^{-1} \sum_{j=1}^{\infty} x_{0j}^v [D' \otimes (V + V^0\nu)] \\ &+ \lambda_1^{-1} \left( \sum_{j=1}^{\infty} x_{0j}^s [e_M \otimes D' \otimes S^0(2)\nu] + \sum_{j=0}^{\infty} x_{1j}^s(1)[e_M \otimes D' \otimes S^0(1)\nu] \right) \end{aligned} \tag{2.56}$$

$$\begin{aligned} z_{00}^s &= \lambda_1^{-1} \sum_{j=1}^{\infty} x_{0j}^s [\bar{I}(M-1) \otimes D' \otimes S^0(2)] \\ &\quad + \lambda_1^{-1} \sum_{j=0}^{\infty} x_{1j}^s(1)[\bar{I}(M-1) \otimes D' \otimes S^0(1)] \end{aligned} \tag{2.57}$$

$$z^s_{01} = \lambda_1^{-1}\sum_{j=1}^{\infty} x^s_{0j}[I(M)\otimes D'\otimes S(2)] \tag{2.58}$$

$$\begin{aligned} z^v_{i0} &= \lambda_1^{-1}\sum_{j=0}^{\infty} x^v_{ij}[D'\otimes V] + \lambda_1^{-1}\sum_{j=0}^{\infty} x^s_{i,j+1}(2)[e_M\otimes D'\otimes S^0(2)\nu] \\ &\quad +\lambda_1^{-1}\sum_{j=0}^{\infty} x^s_{i+1,j}(1)[e_M\otimes D'\otimes S^0(1)\nu] \end{aligned} \tag{2.59}$$

$$\begin{aligned} z^s_{i0}(1) &= \lambda_1^{-1}(\sum_{j=0}^{\infty} x^s_{ij}(1)[I(M)\otimes D'\otimes S(1)] + \sum_{j=0}^{\infty} x^v_{ij}[\acute{e}_1\otimes D'\otimes V^0\beta(1)] \\ &\quad +\sum_{j=0}^{\infty} x^s_{i,j+1}(2)[\bar{I}(M-1)\otimes D'\otimes S^0(2)\beta(1)] \\ &\quad +\sum_{j=0}^{\infty} x^s_{i+1,j}(1)[\bar{I}(M-1)\otimes D'\otimes S^0(1)\beta(1)]) \end{aligned} \tag{2.60}$$

$$z^s_{i1}(2) = \lambda_1^{-1}\sum_{j=1}^{\infty} x^s_{ij}(2)[I(M)\otimes D'\otimes S(2)] \tag{2.61}$$

$$\begin{aligned} z_0 &= [z^v_0\; z^s_{00}\; z^s_{01}] \\ z_i &= [z^v_{i0}\; z^s_{i0}(1)\; z^s_{i1}(2)], \quad i\geq 1 \\ z &= [z_0\; z_1\cdots]. \end{aligned} \tag{2.62}$$

We obtain the delay distribution for high-priority packets as the time to absorption for a Markov chain with the same transition matrix as (2.43). The matrix blocks are as follows:

$$\widetilde{B}_{00} = \begin{bmatrix} V & \acute{e}_1\otimes V^0 & 0 \\ 0 & I(M)\otimes I & 0 \\ e_M\otimes I\otimes S^0(2)\nu & \bar{I}(M-1)\otimes I\otimes S^0(2) & I(M)\otimes I\otimes S(2) \end{bmatrix}$$

$$\widetilde{B}_{10} = \begin{bmatrix} 0 & 0 & 0 \\ e_M \otimes I \otimes S^0(1)\nu & \bar{I}(M-1) \otimes I \otimes S^0(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\widetilde{A}_1 = \begin{bmatrix} I \otimes V & \acute{e}_1 \otimes I \otimes V^0\beta(1) & 0 \\ 0 & I(M) \otimes I \otimes S(1) & 0 \\ e_M \otimes I \otimes S^0(2)\nu & \bar{I}(M-1) \otimes I \otimes S^0(2)\beta(1) & I(M) \otimes I \otimes S(2) \end{bmatrix}$$

$$\widetilde{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ e_M \otimes I \otimes S^0(1)\nu & \bar{I}(M-1) \otimes I \otimes S^0(1)\beta(1) & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then

$$z^{n+1} = z^n\widetilde{P}, \quad W_T = z_0^T e. \tag{2.63}$$

The following set of equations is used in the computation:

$$\begin{align} z_0 &= z_0\widetilde{B}_{00} + z_1\widetilde{B}_{10} \tag{2.64} \\ z_i &= z_i\widetilde{A}_1 + z_{i+1}\widetilde{A}_2 \tag{2.65} \end{align}$$

$$\begin{align} z_0^v &= z_0^v V + z_{01}^s(e_M \otimes I \otimes S^0(2)) + z_{10}^s(e_M \otimes I \otimes S^0(1)) \tag{2.66} \\ z_{00}^s &= z_{00}^s + z_{01}^s(\bar{I}(M-1) \otimes I \otimes S^0(2)) \nonumber \\ &\quad + z_{10}^s(\bar{I}(M-1) \otimes I \otimes S^0(1)) + z_0^v(\acute{e}_1 \otimes V^0) \tag{2.67} \\ z_{01}^s &= z_{01}^s(I(M) \otimes I \otimes S(2)) \tag{2.68} \end{align}$$

$$\begin{align} z_{i0}^v &= z_{i0}^v(I \otimes V) + z_{i1}^s(2)(e_M \otimes I \otimes S^0(2)\nu) \nonumber \\ &\quad + z_{i+1,0}^s(1)(e)M \otimes I \otimes S^0(1)\nu) \tag{2.69} \\ z_{i0}^s(1) &= z_{i0}^v(\acute{e}_1 \otimes I \otimes V^0\beta(1)) + z_{i0}^s(1)(I(M) \otimes I \otimes S(1)) \nonumber \\ &\quad + z_{i1}^s(2)(\bar{I}(M-1) \otimes I \otimes S^0(2)\beta(1)) \nonumber \\ &\quad + z_{i+1,0}^s(1)(\bar{I}(M-1) \otimes I \otimes S^0(1)\beta(1)) \tag{2.70} \\ z_{i1}^s(2) &= z_{i1}^s(2)(I(M) \otimes I \otimes S(2)). \tag{2.71} \end{align}$$

We can compute the delay distribution in the same way as in the standard exhaustive case.

## 2.5 Application of the Analytical Framework

### 2.5.1 Performance Evaluation of Popular MAC Protocols

We demonstrate how the proposed analytical model can be used for performance evaluation of two popular wireless MAC protocols, namely, S-ALOHA and IEEE 802.11 DCF.

#### 2.5.1.1 Energy-Aware Slotted ALOHA

With S-ALOHA protocol, when a node is in active mode and there are packets available in the transmission queue, a packet successfully departs the queue with probability $p$ (which is given by the probability of channel access multiplied by the probability of successful transmission in presence of collision). Therefore, the service time distribution of packets can be considered to be geometric which is a special case of phase-type distribution having 1 phase with $\beta = 1, S = 1 - \mu$ and $S^\circ = \mu$, where $\mu = p$. Also, a prioritized access scheme based on S-ALOHA is obtained if we assume different probability of access (probability of service) for packets with different priorities.

For energy saving, a node goes to sleep mode occasionally and it returns to active mode with probability $\gamma$. To model this sleep and wakeup dynamics a phase type distribution of order 1 can be considered with $V = 1 - \gamma$ and $V^\circ = \gamma$. We refer to $\mu$ and $\gamma$ as the probability of service and the probability of wakeup, respectively. Moreover, we assume that the probability of service is the same for both type-1 and type-2 packets unless it is explicitly stated.

#### 2.5.1.2 IEEE 802.11 DCF

IEEE 802.11 DCF consists of two main mechanisms: contention window adjustment and back-off mechanisms. Once a station (STA) is turned on, it sets the contention window (CW) to the minimum value ($CW_{min}$). The contention window is doubled for every transmission failure until it reaches the maximum value ($CW_{max}$). If the transmission is successful, the contention window is reset to $CW_{min}$.

After a window adjustment, an STA waits for distributed inter-frame space (DIFS) period of time and calculates a back-off value from $U(0, W)$, where $U(a, b)$ generates a random variable uniformly distributed between $a$ and $b$, and $W$ is the current contention window. After this point, the back-off value is decreased by one every idle[3] time slot. When the channel becomes *busy*, the back-off value is frozen until the channel becomes idle again for DIFS period of time. After that, the STA starts decreasing the back-off value by one for each subsequent idle time slot. When the back-off value is zero, the STA can transmit data in the next time slot.

Setting up a phase-type service process with two phases (one for the case when the server is in *back-off* mode and the other for the case when the server is in *busy* mode), enables us to model the IEEE 802.11 DCF MAC protocol. The back-off mode ends when back-off counter expires depending on the stage of the back-off and the probability of back-off counter expiration is given as follows [41]:

$$Pr(BC = 1) = \sum_{k=0}^{m-1} \frac{p^k}{2^k CW_{min}} + \frac{p^m}{CW_{max}} \tag{2.72}$$

where $p$ is the probability of collision and $m = log_2(\frac{CW_{max}}{CW_{min}})$. On the other hand, a node goes to the back-off mode (from busy mode) when a collision occurs (with probability $p$).

We assume that the packet collision process follows a Bernoulli process (as in [36]) and that the packet collision probability is constant and independent of the number of retransmissions already suffered. This assumption was extended to the finite load case in [66]. Therefore, a phase-type distribution for the service process can be set up as follows:

$$S = \begin{bmatrix} 1 - Pr(BC = 1) & Pr(BC = 1) \\ p & 0 \end{bmatrix} \quad S^{\circ} = \begin{bmatrix} 0 \\ 1 - p \end{bmatrix} \tag{2.73}$$

Similar to the S-ALOHA case, a 1-phase PH distribution can be assumed for the vacation process to model the sleep and wakeup dynamics in a node.

[3] The channel is considered idle when there is no ongoing transmission.

### 2.5.2 Optimal System Parameter Settings

For a given arrival probability and given QoS requirements, the wakeup probability and the service probability at a node are the tuning parameters to control the node's behavior dynamically, where the service probability can be tuned by changing the MAC/PHY layer access control and error control parameters. Having a high probability of service may not guarantee the required QoS when the wakeup probability is low.

We can use the presented analytical model to obtain the optimal system parameter settings. For example, if we want to maximize the battery lifetime, the wakeup probability $\gamma$ should be minimized while maintaining the packet loss rate ($p_l$) below a target threshold $p_l^{tar}$. We can then formulate an optimization problem as follows:

$$\begin{aligned} \text{minimize:} \quad & \gamma \\ \text{s.t.} \quad & p_l(\gamma) < p_l^{tar}. \end{aligned} \tag{2.74}$$

With this formulation, the decision variable is $\gamma$. Similar formulations can be developed under constraints on packet delay and packet loss probability (or a combination of these QoS metrics) as well. We can use the *Golden Section Search* [67] algorithm to find the minima of (2.74) and the corresponding optimal value $\tilde{\gamma}$. Starting at a given initial interval $a_1$ and $b_1$ ($a_1 < b_1$), the algorithm below (Algorithm 2.5.1) proceeds by evaluating the function $f(x) = |p_l(x) - p_l^{tar}|$ where $\tau = 0.61803$ is *golden section* in the algorithm.

**Algorithm 2.5.1:** Golden Section Search(*inputs* : $a_1, b_1, tol$)

```
$b_1 \leftarrow a_1 + (1-\tau)(b_1 - a_1), F_b \leftarrow f(b_1)$
$d_1 \leftarrow b_1 - (1-\tau)(b_1 - a_1), F_d \leftarrow f(d_1)$
$\tilde{\gamma} \leftarrow b_1$
repeat
  if $F_b < F_d$
    then {
      $\tilde{\gamma} \leftarrow b_k$
      $a_{k+1} \leftarrow a_k, b_{k+1} \leftarrow d_k, d_{k+1} \leftarrow b_k$
      $b_{k+1} \leftarrow a_{k+1} + (1-\tau)(b_{k+1} - a_{k+1})$
      $F_d \leftarrow F_b, F_b \leftarrow f(b_{k+1})$
    }
    else {
      $\tilde{\gamma} \leftarrow d_k$
      $a_{k+1} \leftarrow b_k, b_{k+1} \leftarrow b_k, b_{k+1} \leftarrow d_k$
      $d_{k+1} \leftarrow b_{k+1}(1-\tau)(b_{k+1} - a_{k+1})$
      $F_b \leftarrow F_d, F_d \leftarrow f(d_{k+1})$
    }
until $b_{k+1} - a_{k+1} < tol$
return $(\tilde{\gamma})$
```

## 2.6 Performance Evaluation

### 2.6.1 System Parameters

We analyze the queue length and the delay distribution as well as the probability of loss (for finite queues) for high-priority packets. For packet arrival, we assume a Bernoulli process which is a special case of MAP with $\alpha$ as the probability of arrival. The probability of arrival for a high-priority packet is $\alpha_1$ and the probability of arrival for a low-priority packet is $\alpha_2$. Then, the matrices $D_0, D_{11}, D_{12}$ and $D_2$ become scalars and are given as follows:

$$D_0 = (1-\alpha_1)(1-\alpha_2) = d_0, \quad D_2 = \alpha_1\alpha_2 = d_2$$
$$D_{11} = \alpha_1(1-\alpha_2) = d_{11}, \; D_{12} = (1-\alpha_1)\alpha_2 = d_{12}.$$

For numerical analysis and simulation, we study the standard exhaustive vacation case by considering the fact that this is a variant of the number-limited exhaustive vacation case with $M = \infty$. For the number-limited case we assume $M = 1$. Note that, for numerical analysis, we can obtain the performance measures after the $R$ matrix and the boundary values $[x_0, x_1]$ have been computed. To compute the delay distribution for a packet, we need to first compute the probability distribution corresponding to the system states (given as the $z$ vector) upon the arrival of a packet in the system. For the 802.11 DCF MAC protocol, we assume the same set of parameters as specified in [68]. Each node operates under non-saturation condition and we obtain the probability of collision from [66].

### 2.6.2 Simulation Setup

Simulations are performed using MATLAB with the same system parameters for both the standard and the number-limited cases as described above. We simulate two arrival streams (with predefined probability of arrival) to a node. There are two queues in the node and the node processes the packets in the queues in a FIFO fashion. We assume that the tagged node has 10 neighboring nodes and all the nodes are identical. If more than one node attempts transmission in a time slot, collision occurs. Otherwise, a transmission is assumed to be successful. We assume ideal channel condition in which packet errors occur only due to collisions. The nodes with unsuccessful transmission will attempt transmission in the successive time slots based on specific MAC protocol (for example, in case of 802.11 DCF MAC, the node will go through the predefined back-off procedure).

For the two queues in each node, a decision will be made to serve the queues based on their priority at the beginning of the transmission of a packet for the first time. A packet "in service" in a node won't be interrupted by a lower-priority packet in the case of a collision. The selected packet will continue to be "served" until it is successfully transmitted. At the end of the service (after the packet is successfully sent), the decision to continue serving or going to vacation is made based on the pre-selected vacation strategy.

For the nodes in vacation, a probabilistic approach is used to bring them in service and in the case that the transmission queues are empty, a node returns to the vacation (i.e., sleep mode) immediately.

Arrivals to the queues are independent of whether the node is in active mode or sleep mode. Two independent packet generators with pre-defined input arrival distribution, generate the input traffic to the queues in each node. We assume non-batch arrivals. Therefore, at most one packet of each type in each node is generated at each time slot. We assume that if an arrival event and a service completion event coincide, the latter occurs first followed by the former.

The procedure mentioned above is performed at each time slot. While some events such as packet generation need to be executed in all the time slots, other events such as packet transmission may need to be executed only if necessary. We have used *Independent Replications Method (IRM)* with run up time and hard close in our simulation to obtain reliable results. For each data point, we repeat the simulation for 10000 times and consider 1000 time slots in each iteration.

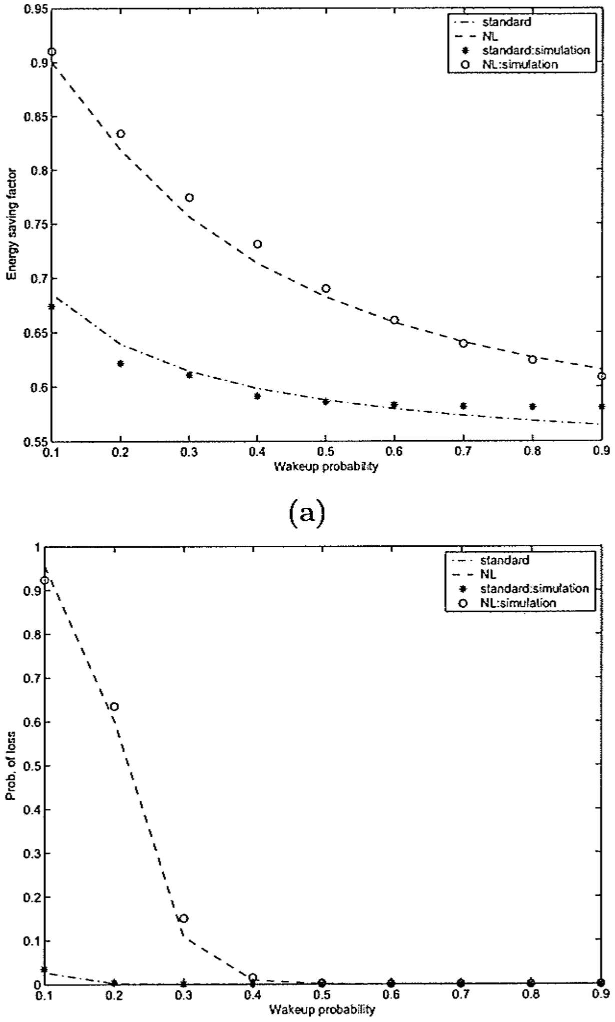
## 2.6.3 Numerical and Simulation Results

### 2.6.3.1 Effect of Wakeup Probability

Figs. 2.2-2.4 show the effect of wakeup probability on queue length distribution, access delay, energy saving factor and probability of packet loss in slotted ALOHA. Moreover, Fig. 2.5 shows access delay and queue length distributions for the 802.11 DCF MAC protocol in the standard exhaustive case when $\gamma$=0.8. The simulation results are plotted against the numerical results obtained from the analysis. It is evident that the simulation results follow the numerical results very closely which validates the accuracy of the analytical model.

The probability of packet arrival and the probability of service (in Figs. 2.2-2.4) are assumed to be 0.2 and 0.9 respectively for both types of packets. Fig. 2.2(a) shows the energy saving factor as a function of wakeup probability for both the standard and the number-limited service cases. Note that, the wakeup probability can be chosen as a function of the battery level in the node.

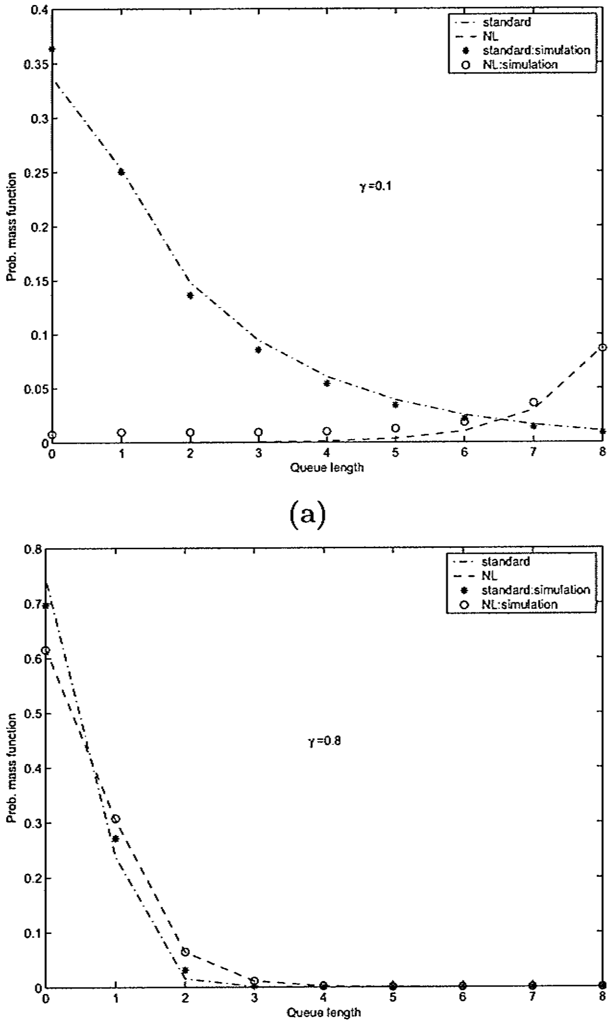
Although the number-limited strategy is more energy efficient, the probability of packet loss is more severe in this case when the wake up probability is low. From Fig. 2.3 we observe that with number-limited service, it is more probable that the queue length will be larger than that for the standard exhaustive case. Also, the queue length distribution for the number-limited case approaches that for the standard case when the wakeup probability becomes higher. However, the distributions differ



(a)

(b)

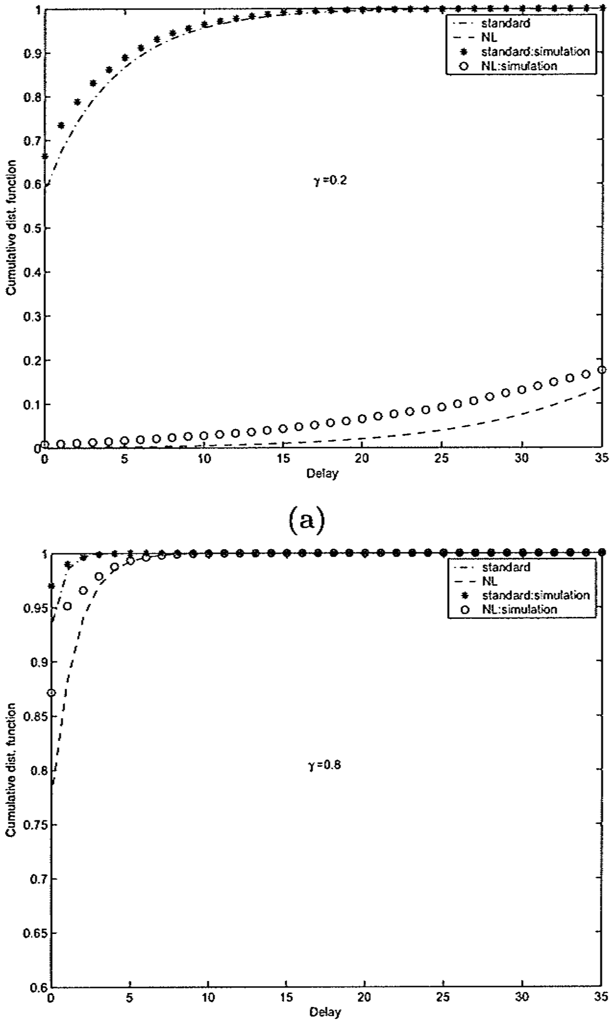
**Figure 2.2.** *Effect of wakeup probability on energy saving factor and packet loss rate.*



(a)

(b)

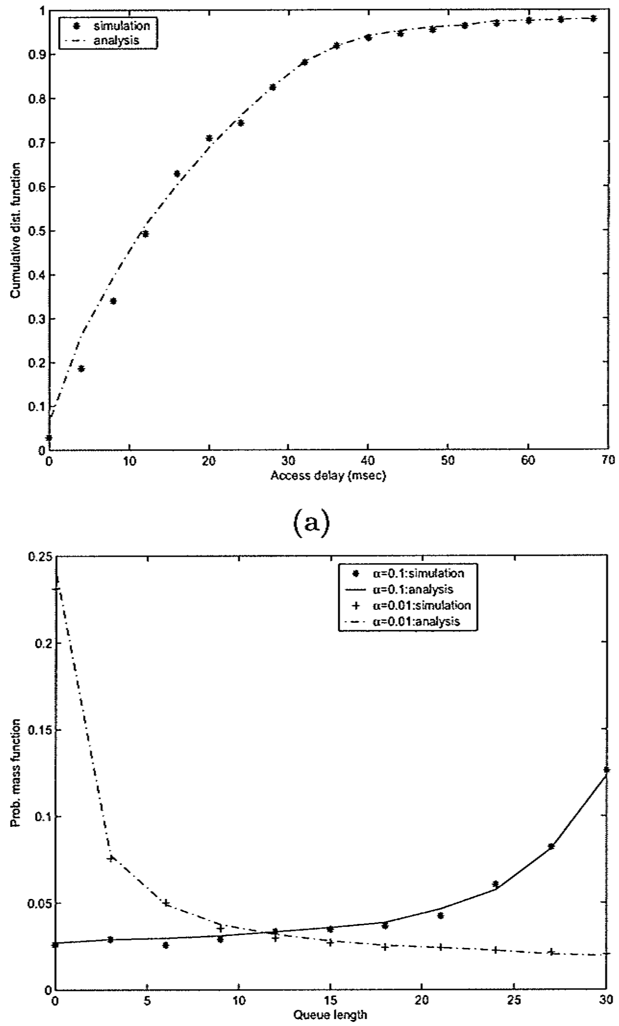
**Figure 2.3.** *Effect of wakeup probability $\gamma$ on queue length distribution.*



(a)

(b)

**Figure 2.4.** *Effect of wakeup probability $\gamma$ on the distribution of access delay.*



(a)

(b)

**Figure 2.5.** *Access delay and queue length distribution for 802.11 DCF MAC (for wakeup probability $\gamma$ =0.8, and different values of arrival probability $\alpha$).*

significantly when the wakeup probability is small.

Fig. 2.4 clearly shows the difference in the access delay performance for the two different service strategies. Specifically, with small wakeup probability, access delay is much higher in the number-limited case compared to that in the standard exhaustive service case.

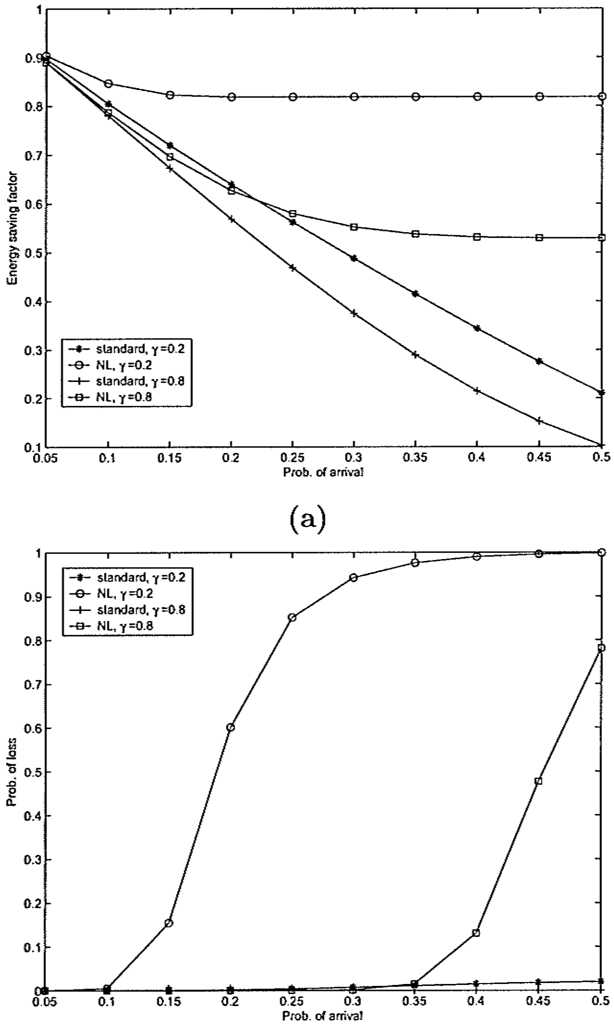
#### 2.6.3.2 Effect of Probability of Arrival

Fig. 2.6 shows typical variations in energy saving factor and packet loss probability under varying packet arrival probability in energy-aware S-ALOHA. For higher values of packet arrival probability the energy efficiency of the standard exhaustive discipline may deteriorate significantly. On the other hand, the number-limited strategy may offer better energy saving performance with reasonable packet loss rate as long as the wakeup probability is not very small. The queue length distribution and the access delay distribution are shown in Figs. 2.7-2.8, respectively, for two different packet arrival rates and wakeup probabilities in slotted ALOHA. It is evident that the number-limited strategy performs reasonably well except for small values of wakeup probabilities.

#### 2.6.3.3 Effect of Probability of Service

Fig. 2.9 shows typical variations in energy saving factor and packet loss probability for different values of service probability for type-1 packets in prioritized S-ALOHA. The probability of service for type-2 packets in this figure is $\mu_2 = 0.2$ and is fixed. While the superiority of the energy saving performance of the number-limited discipline over the standard exhaustive discipline is obvious, the packet loss probability is high when the wakeup probability is small. This implies that when the wakeup mechanism is very conservative (e.g., due to lower level of available battery power), for the number-limited case, even with higher values of service probability, the desired QoS performance for high-priority packets may not be achieved.

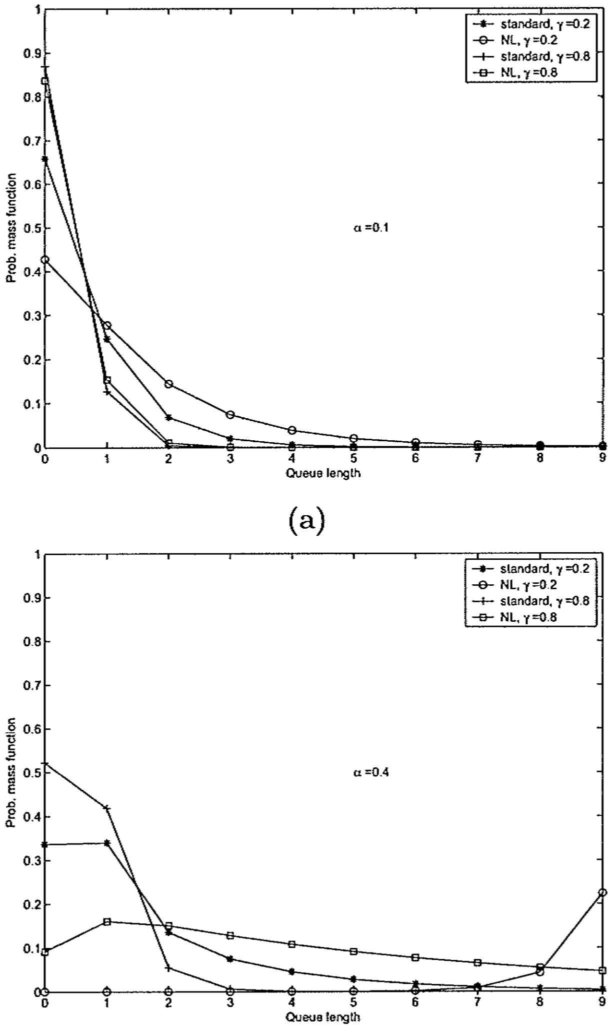
Similar conclusions can be also drawn from Figs. 2.10-2.11 which demonstrate the poor QoS performance (in terms of queue length and access delay distribution) of the number-limited service discipline even with higher values of service probability when the wakeup probability is small. Note that, a higher value of service probability in



(a)

(b)

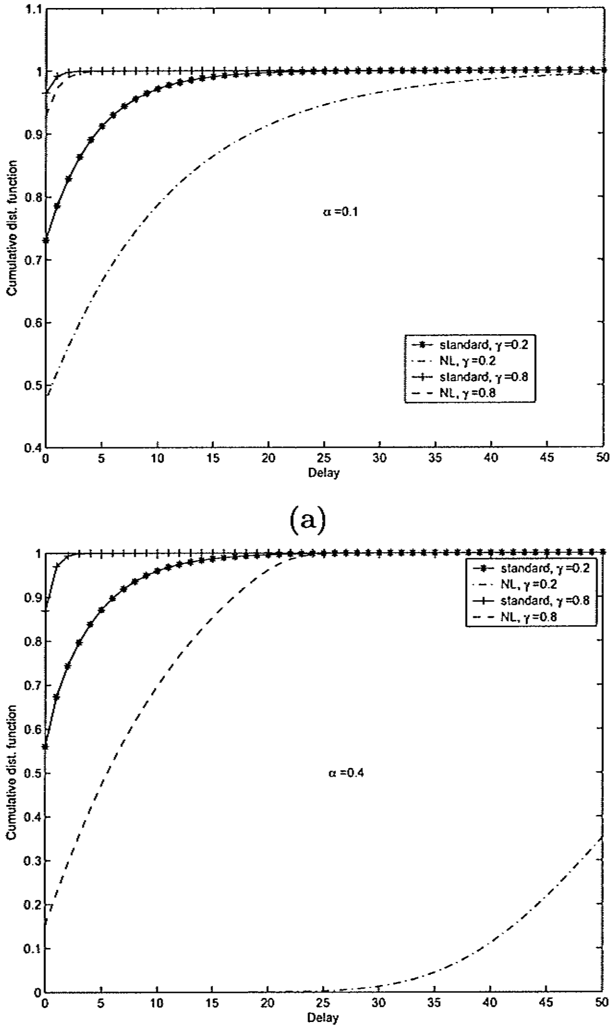
**Figure 2.6.** *Effect of probability of arrival on energy saving factor and packet loss rate (for different values of wakeup probability $\gamma$).*



(a)

(b)

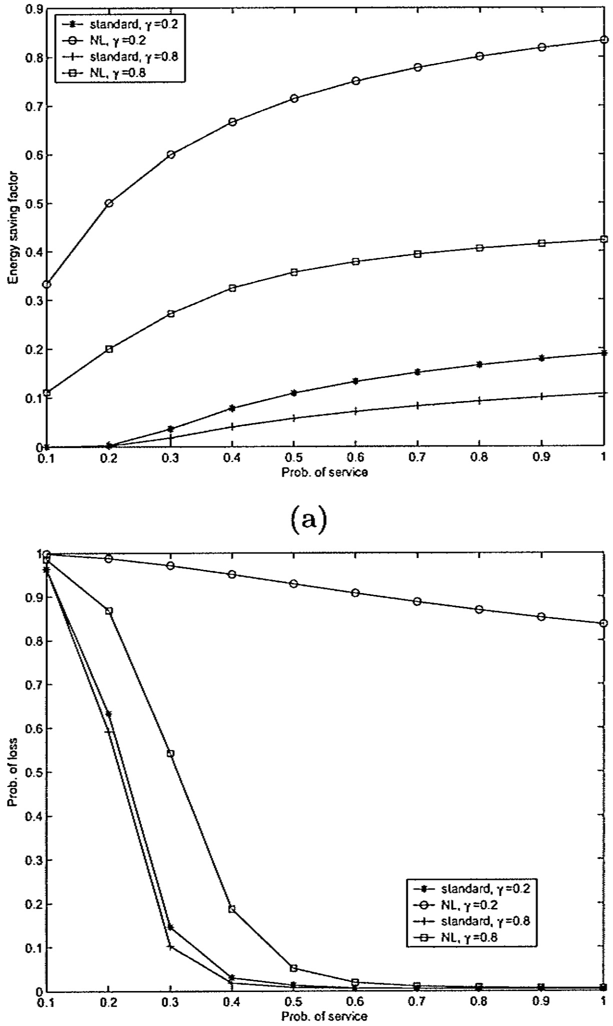
**Figure 2.7.** *Effect of probability of arrival $\alpha$ on queue length distribution ($\gamma$ is the wakeup probability).*



(a)

(b)

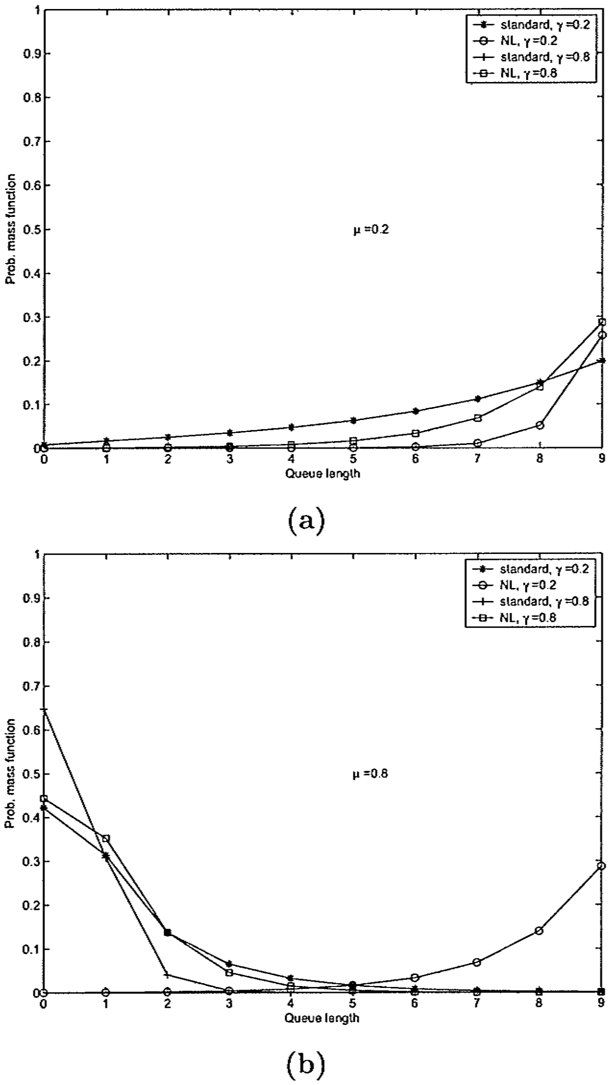
**Figure 2.8.** *Effect of probability of arrival $\alpha$ on the distribution of access delay ($\gamma$ is the wakeup probability).*



(a)

(b)

**Figure 2.9.** *Effect of probability of service on energy saving factor and packet loss probability in prioritized S-ALOHA (probability of service for low-priority packets is $\mu_2$ = 0.2, and $\gamma$ is the wakeup probability).*



(a)

(b)

**Figure 2.10.** *Effect of probability of service $\mu$ on queue length distribution ($\gamma$ is wakeup probability).*

a practical scenario can be achieved through improved channel coding and/or more efficient channel access and error recovery protocols.

#### 2.6.3.4 Optimal Value of Wakeup Probability

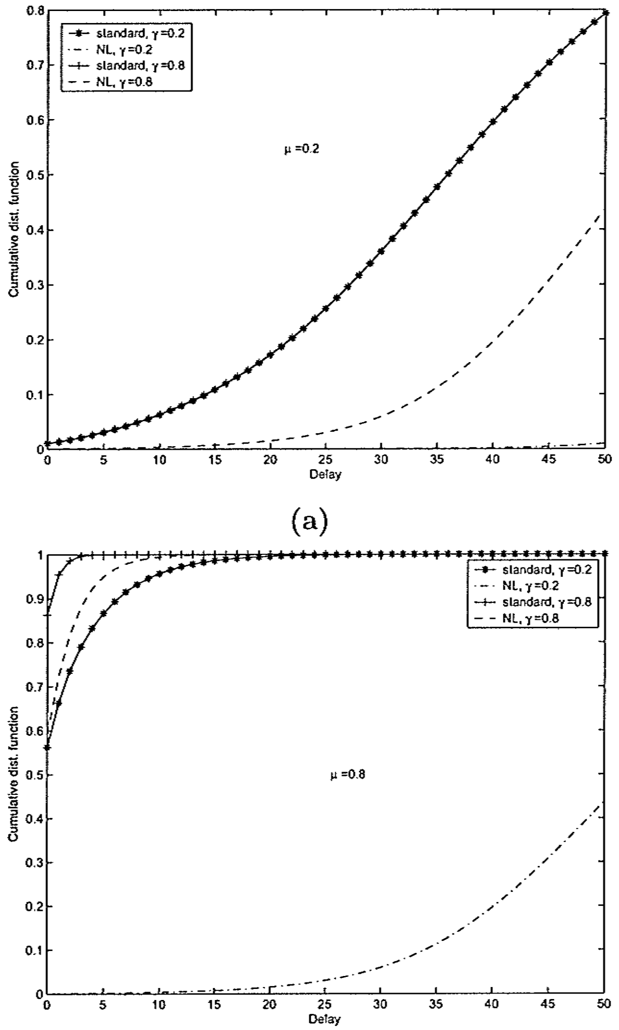
Figs. 2.12-2.13 show typical variations in the optimum value of $\gamma$ with different packet arrival probability under constraints on packet access delay, packet loss rate and energy saving factor in slotted ALOHA. In these cases, probability of arrival for low-priority packets is $\alpha_2 = 0.1$ and probability of service is $\mu_1 = \mu_2 = \mu = 0.9$ for both types of packets.

As expected, when the target packet access delay reduces and/or the packet arrival probability increases, the optimum value of the wake up probability $\tilde{\gamma}$ increases (Fig. 2.12(a)). Similarly, $\tilde{\gamma}$ increases as the target packet loss probability becomes more stringent (Fig. 2.12(b)). Again, with increasing packet arrival rate, wakeup probability needs to be decreased when a target energy saving requirement is to be met (Fig. 2.13).

## 2.7 Chapter Summary

We have presented a novel queueing analytical framework to evaluate the QoS and energy efficiency tradeoff at a node in a multi-hop mesh/relay network. We have combined the priority queueing and the vacation queueing models to describe two important operational aspects of a node in a relay network. While the vacation model has been used to model the sleep mechanism in such a node, the priority queueing model has enabled us to distinguish between the relayed data packets and the node's own packets. The presented analytical framework is very general and comprehensive in that it considers Markovian arrival process, phase-type distribution for service time and phase-type distribution for vacation period with two types of service disciplines, namely, the standard exhaustive and the number-limited exhaustive both in multiple vacation case. The proposed general analytical model has been used for performance evaluation of two different MAC protocols, namely, energy-aware S-ALOHA and 802.11 DCF.

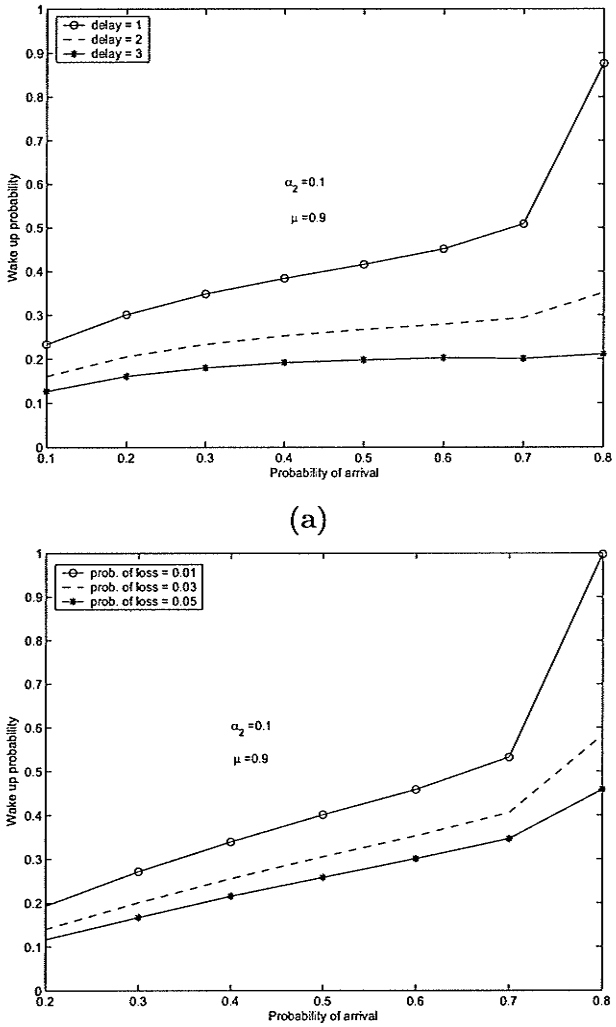
The tradeoff between the QoS performances for the relayed packets (i.e., queue



(a)

(b)

**Figure 2.11.** *Effect of probability of service $\mu$ on distribution of access delay ($\gamma$ is the wakeup probability).*



(a)

(b)

**Figure 2.12.** *Optimum wakeup probability for target access delay and packet loss probability ($\mu$ is the probability of service).*

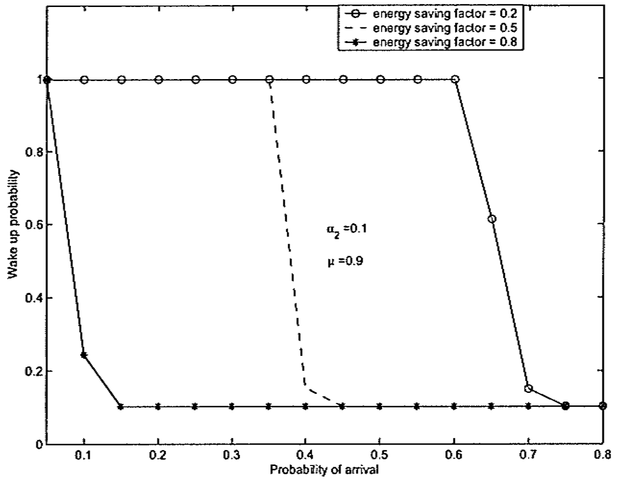


**Figure 2.13.** *Optimum wakeup probability for target energy saving performance ($\mu$ is the probability of service).*

length distribution, distribution of access delay and probability of packet loss) and the energy saving performance at a node has been analyzed under different system parameter settings. Furthermore, we have demonstrated how to obtain the optimum value of the wakeup probability under QoS and energy saving requirements. Also, based on the presented analytical model, the optimum number of packets to be served ($M$) for the number-limited service discipline can be determined as a function of the system parameters.

# Chapter 3

# Energy-Aware MAC for Distributed Energy-Limited Multi-Hop Wireless Networks

## 3.1 Introduction

In this chapter we present a queueing analytical framework for performance evaluation of a distributed and energy-aware MAC protocol for wireless packet data networks with service differentiation. Specifically, we consider a node (both buffer-limited and energy-limited) in the network with two different types of traffic, namely, high-priority and low-priority traffic, and model the node as a MAP/PH (Phase-Type)/1/K non-preemptive priority queue. Finite queue analysis makes the model more realistic than the infinite queue case in Chapter 2. The MAC layer in the node is modeled as a server and a vacation queueing model is used to model the sleep and wakeup mechanism of the server. We study standard exhaustive and number-limited exhaustive vacation models both in multiple vacation case.

Unlike Chapter 2, a *setup time* for the head-of-line packet in the queue is considered which abstracts the contention and the backoff mechanism of the MAC protocol in the node. Rather than assuming ideal channel conditions (as it was done in Chapter 2), a more realistic wireless channel will be consider which enables us to investigate the effects of packet transmission errors on the performance behavior of the system. After obtaining the stationary distribution of the system using matrix-geometric method, we study the performance indices such as packet dropping probability, access delay and queue length distribution for high-priority packets as well as the energy saving factor at the node.

Typical numerical results obtained from the analytical model are presented and validated by extensive simulations. Also, we show how the optimal MAC parameters can be obtained using numerical optimization.

## 3.2 System Model

We model the distributed and energy-aware MAC mechanism at a node in a wireless network as a discrete-time queueing system.

The basic system model in this chapter is similar to the model described in Chapter 2. The service process is determined by the MAC protocol activity described in Section 2.2. The type-1 and type-2 packets are stored in two separate queues (Fig. 3.1(a)). We study the non-preemptive priority case and packets from each queue are served on a FIFO basis.

There are three major differences between the system in our study in this chapter and the system in Chapter 2. First of all, finite queues are studied in this chapter. We also distinguish between the head-of-line (HOL) packet in the queue and other packets. Finally, non-ideal channel affects and infinite-persistent ARQ error control mechanism are considered in this chapter.

To avoid the intricacies of any specific MAC protocol (e.g., IEEE 802.11 DCF MAC [68]), we use a generic distribution for the time required to initiate service (i.e., data packet transmission) from the system's idle state and this is referred to as the *setup time* in this chapter. For example, the setup time can model the overhead due to channel sensing, back-off, and handshaking mechanisms in the MAC protocol before the actual packet transmission starts. Again, to model the packet transmission time (which can vary depending on the packet length) as well as the time required for the sender to receive the acknowledgement (ACK) information, a service time distribution is assumed for each packet[1]. The setup time and the service time are assumed to have phase-type (PH) distribution[2]. While the time required to transmit the first packet

[1] Note that, for MAC protocols such as the IEEE 802.11 DCF, the ACK reception time is in general not deterministic.

[2] Phase-type distribution can virtually model any practical distribution [69].

(when the system switches to the active mode from sleep mode) includes both the setup time and the service time, the time required for transmission of the subsequent packets consists of only the service time (e.g., a reservation type of MAC protocol).

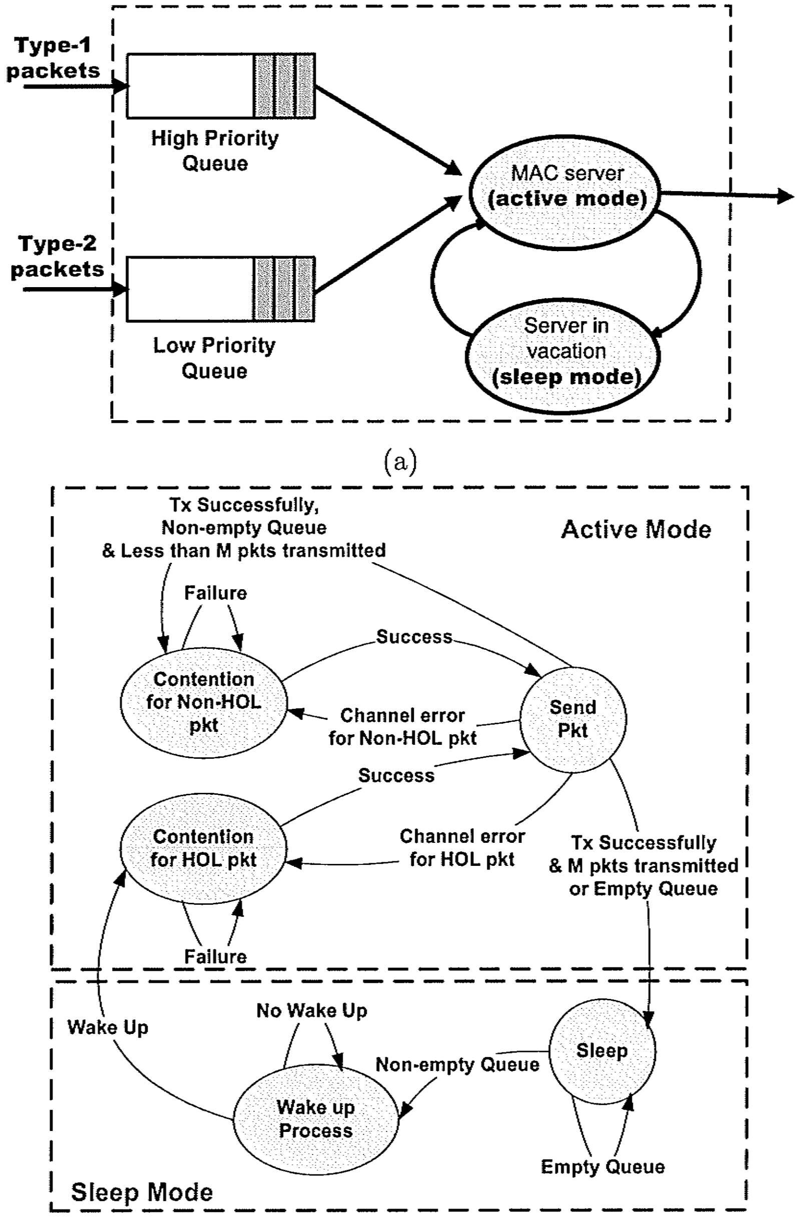
The server state diagram in Fig. 3.1(b) illustrates the service process in more detail. When the server switches to the active mode, a setup time, which includes the overhead due to channel sensing, handshaking, and back-off etc., is required before the actual data packet transmission occurs. We distinguish between the distribution of transmission latency for the head-of-line (HOL) packet in the queue and the distribution of transmission latency for the packets following the HOL packet. While the transmission latency for the HOL packet consists of both the service time and the setup time, the transmission latency for a non-HOL packet consists only of the service time. We assume that when the node acquires channel access, the channel is reserved for the node (e.g., reservation MAC protocol) for a period of time, and therefore, the same service time distribution can be assumed for each of the non-HOL packets.

The decision regarding staying in active mode or switching back to sleep mode is made based on the queue state and selected strategy. As shown in Fig. 3.1(a), an empty queue causes the server to switch to sleep mode. While the queues are not empty, the decision is made based on $M$ (the number of transmitted packets since last wakeup) during the *Send Pkt* state.

The transmission latency for the HOL packet follows a discrete phase-type distribution represented by $(\beta_p(i), S_p(i))$, for packet type $i$ $(i = 1, 2)$. We use $(\beta_c(i), S_c(i))$, $i = 1, 2$ to represent the service time distributions for the non-HOL packets. In this representation, $\beta_p(i)$ and $\beta_c(i)$ are row vectors, $S_p(i)$ and $S_c(i)$ are sub-stochastic matrices, $S_p^0(i) = \mathbf{e} - S_p(i)\mathbf{e}$ and $S_c^0(i) = \mathbf{e} - S_c(i)\mathbf{e}$.

We assume a non-ideal wireless channel where $p$ is the probability of successful transmission of a packet (i.e., $1 - p$ is the probability of transmission failure). For simplicity, we assume that each node knows whether the packet transmission is successful or not at the end of the same slot that transmission occurs. We assume an infinite-persistent ARQ mechanism for retransmission in case of transmission error.

For the vacation process in this chapter, we apply the same vacation strategies as those described in Section 2.2.



(a)

(b)

**Figure 3.1.** *System model: (a) service differentiation and the energy saving mechanism at the MAC layer protocol and (b) server state diagram.*

## 3.3 Standard Exhaustive Vacation System

### 3.3.1 The Markov Chain

We set up our model by considering the following state space:

$$\begin{aligned}
&\Omega_0^v = (0, 0, j, l)\\
&\Omega_0^s = (0, i_2, j, k_2), \quad K \geq i_2 \geq 1\\
&\Omega^v = (i_1, i_2, , j, l), \quad (i_1 + i_2) \geq 1,\ K \geq i_1 \geq 1,\ K \geq i_2 \geq 0\\
&\Omega_1^s = (i_1, i_2, j, k_1), \quad K \geq i_1 \geq 1,\ K \geq i_2 \geq 0\\
&\Omega_2^s = (i_1, i_2 + 1, j, k_2), \quad K \geq i_1 \geq 1; K \geq i_2 \geq 0.
\end{aligned} \tag{3.1}$$

$\Omega_0^v$ represents the vacation while the queues are empty with the arrival in $j^{th}$ phase and the vacation in $l^{th}$ phase. In $\Omega_0^s$, there are only type-2 packets in the system with arrival in the $j^{th}$ phase and a type-2 packet is being served while the service is in phase $k_2$. In states $\Omega^v$, $\Omega_1^s$ and $\Omega_2^s$, there is at least one type-1 packet in the system. In $\Omega^v$, the system is in vacation (i.e, sleep mode) while the phases of arrival and vacation are $j$ and $l$, respectively. $\Omega_1^s$ and $\Omega_2^s$ represent the system in service when the packet in the server is type-1 with $k_1$ being the phase of service and the packet in the server is type-2 with $k_2$ being the phase of service, respectively.

Considering the setup time, state $\Omega_0^s$ is partitioned to $\Omega_0^{s,p}$ and $\Omega_0^{s,c}$ that represent the state for head-of-line packet and other packets, respectively. The same type of partitioning is applicable to $\Omega_1^s$ and $\Omega_2^s$ and they will be partitioned into $\Omega_1^{s,p}$, $\Omega_1^{s,c}$ and $\Omega_2^{s,p}$, $\Omega_2^{s,c}$, respectively. Consequently, the state space for the system becomes

$$\Omega = \Omega_0^v \bigcup \Omega_0^{s,p} \bigcup \Omega_0^{s,c} \bigcup \Omega^v \bigcup \Omega_1^{s,p} \bigcup \Omega_1^{s,c} \bigcup \Omega_2^{s,p} \bigcup \Omega_2^{s,c}. \tag{3.2}$$

The transition matrix $P$ describing this Markov chain has the form:

$$P = \begin{bmatrix} B_{00} & B_{01} & & & & \\ B_{10} & A_1 & A_0 & & & \\ & A_2 & A_1 & A_0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_2 & A_1 & A_0 \\ & & & & A_2 & A_1 + A_0 \end{bmatrix} \tag{3.3}$$

where the $i^{th}$ row in the above matrix represents $i$ high-priority packets in the system. Moreover, the $j^{th}$ row in the block matrices $B_{00}, B_{01}, B_{10}, B_{11}, A_0, A_1, A_2$ represents $j$ low-priority packets in the system. Here, the state space and the matrices are defined in the same way as in [29]. The block matrices are described as follows.

$$B_{00} = \begin{bmatrix} B_{00}^{00} & B_{00}^{01} & & & & \\ B_{00}^{10} & B_{00}^{1} & B_{00}^{0} & & & \\ & B_{00}^{2} & B_{00}^{1} & B_{00}^{0} & & \\ & & \ddots & \ddots & \ddots & \\ & & & B_{00}^{2} & B_{00}^{1} & B_{00}^{0} \\ & & & & B_{00}^{2} & B_{00}^{1} + B_{00}^{0} \end{bmatrix} \tag{3.4}$$

Submatrices in $B_{00}$ represent changes in low-priority queue when high-priority queue remains empty. In $B_{00}^{00}$ low-priority queue remains empty and in $B_{00}^{01}$ it will be non-empty by a packet arrival. Should sleep process finish in $B_{00}^{01}$, setup time starts for the head-of-line (HOL) packet recently arrived.

$$B_{00}^{00} = D_0 \otimes (V + V^0 \nu) \qquad B_{00}^{01} = \begin{bmatrix} D_{12} \otimes V & D_{12} \otimes V^0 \beta_p(2) & 0 \end{bmatrix}$$

In $B_{00}^{10}$, a non-empty low-priority queue becomes empty by serving the last packet in the queue (which could be HOL or non-HOL packet) and since high-priority queue is also empty, server takes a vacation. For $B_{00}^{2}$, serving a packet in low-priority queue and no arrival in this queue, decreases the number of packets in the queue. However, queue will not be empty. Service time for non-HOL packet starts following serving the previous packet.

$$B_{00}^{10} = \begin{bmatrix} 0 \\ D_0 \otimes pS_p^0(2)\nu \\ D_0 \otimes pS_c^0(2)\nu \end{bmatrix} \quad B_{00}^{2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & D_0 \otimes pS_p^0(2)\beta_c(2) \\ 0 & 0 & D_0 \otimes pS_c^0(2)\beta_c(2) \end{bmatrix}$$

$B_{00}^1$ is the case that the number of packets in low-priority queue is not changed.

$$B_{00}^1 = \begin{bmatrix} D_0 \otimes V & D_0 \otimes V^0\beta_p(2) & 0 \\ 0 & D_0 \otimes S_p(2) & D_0 \otimes (1-p)S_p^0(2)\beta_c(2) + D_{12} \otimes pS_p^0(2)\beta_c(2) \\ 0 & 0 & B_{00}^1(s_2^c s_2^c) \end{bmatrix}$$

where

$$B_{00}^1(s_2^c s_2^c) = D_0 \otimes S_c(2) + D_0 \otimes (1-p)S_c^0(2)\beta_c(2) + D_{12} \otimes pS_c^0(2)\beta_c(2) \qquad (3.5)$$

In $B_{00}^0$ not only no packet is served, but also a new packet arrival occurs.

$$B_{00}^0 = \begin{bmatrix} D_{12} \otimes V & D_{12} \otimes V^0\beta_p(2) & 0 \\ 0 & D_{12} \otimes S_p(2) & D_{12} \otimes (1-p)S_p^0(2)\beta_c(2) \\ 0 & 0 & D_{12} \otimes S_c(2) + D_{12} \otimes (1-p)S_c^0(2)\beta_c(2) \end{bmatrix}$$

In $B_{01}$, high-priority queue becomes non-empty with a packet arrival while it was empty in previous time slot.

$$B_{01} = \begin{bmatrix} B_{01}^{00} & B_{01}^{01} & & & \\ B_{01}^{2} & B_{01}^{1} & B_{01}^{0} & & \\ & \ddots & \ddots & \ddots & \\ & & B_{01}^{2} & B_{01}^{1} & B_{01}^{0} \\ & & & B_{01}^{2} & B_{01}^{1} + B_{01}^{0} \end{bmatrix}$$

$B_{01}^{00}$ and $B_{01}^{01}$ in $B_{01}$ represent the case that low-priority queue has been empty and becomes empty and non-empty respectively.

$$B_{01}^{00} = \begin{bmatrix} D_{11} \otimes V & D_{11} \otimes V^0\beta_p(1) & 0 & 0 & 0 \end{bmatrix}$$

$$B_{01}^{01} = \begin{bmatrix} D_2 \otimes V & D_2 \otimes V^0\beta_p(1) & 0 & 0 & 0 \end{bmatrix}$$

$B_{01}^{0}$ is similar to $B_{01}^{01}$. However, low-priority queue has not been empty in $B_{01}^{0}$ before new packet arrival.

$$B_{01}^{0} = \begin{bmatrix} D_2 \otimes V & D_2 \otimes V^0\beta_p(1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$B_{01}^{1}$ is similar to $B_{00}^{1}$ in which the state of the low-priority queue remains unchanged. However, high-priority queue is not empty in $B_{01}^{1}$.

$$B_{01}^{1} = \begin{bmatrix} D_{11} \otimes V & D_{11} \otimes V^0\beta_p(1) & 0 & 0 & 0 \\ 0 & 0 & D_2 \otimes pS_p^0(2)\beta_c(1) & D_2 \otimes S_p(2) & B_{01}^1(s_2^p s_2^c) \\ 0 & 0 & D_2 \otimes pS_c^0(2)\beta_c(1) & 0 & B_{01}^1(s_2^c s_2^c) \end{bmatrix}$$

$$B_{01}^1(s_2^p s_2^c) = D_2 \otimes (1-p)S_p^0(2)\beta_c(2)$$

$$B_{01}^1(s_2^c s_2^c) = D_2 \otimes (1-p)S_c^0(2)\beta_c(2) + D_2 \otimes S_c(2)$$

In $B_{01}^2$, a low-priority packet is served and low-priority queue may become empty due to no arrival of this type of packet. However, an arrival in high-priority queue keeps the server busy and node stays in active mode.

$$B_{01}^{2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_{11} \otimes pS_p^0(2)\beta_c(1) & D_{11} \otimes S_p(2) & D_{11} \otimes (1-p)S_p^0(2)\beta_c(2) \\ 0 & 0 & 0 & 0 & B_{01}^2(s_2^c s_2^c) \end{bmatrix}$$

$$B_{01}^2(s_2^c s_2^c) = D_{11} \otimes S_c(2) + D_{11} \otimes (1-p)S_c^0(2)\beta_c(2)$$

For $B_{10}$, a non-empty high-priority queue becomes empty when its last packet is served.

$$B_{10} = \begin{bmatrix} B_{10}^{00} & B_{10}^{0} & & & \\ & B_{10}^{1} & B_{10}^{0} & & \\ & & \ddots & \ddots & \\ & & & B_{10}^{1} & B_{10}^{0} \\ & & & & B_{10}^{1} + B_{10}^{0} \end{bmatrix} \tag{3.6}$$

Since there is no low-priority packet arrival in already empty low-priority queue in $B_{10}^{00}$, server takes a vacation. On the other hand, an arrival in low-priority queue in $B_{10}^{0}$, keeps the server busy.

$$B_{10}^{00} = \begin{bmatrix} 0 \\ D_0 \otimes pS_p^0(1)\nu \\ D_0 \otimes pS_c^0(1)\nu \\ 0 \\ 0 \end{bmatrix} \quad B_{10}^{0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & D_{12} \otimes pS_p^0(1)\beta_c(2) \\ 0 & 0 & D_{12} \otimes pS_c^0(1)\beta_c(2) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

For $B_{10}^{1}$, since there are low-priority packets to be served, server remains in active mode to start serving them.

$$B_{10}^{1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & D_0 \otimes pS_p^0(1)\beta_c(2) \\ 0 & 0 & D_0 \otimes pS_c^0(1)\beta_c(2) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$A_0$ represents an increment in the number of high-priority packets in the queue.

$$A_0 = \begin{bmatrix} A_0^1 & A_0^0 & & \\ & \ddots & \ddots & \\ & & A_0^1 & A_0^0 \\ & & & A_0^1 + A_0^0 \end{bmatrix} \tag{3.7}$$

In $A_0^0$, we have an arrival in low-priority queue beside a high-priority arrival. However, no low-priority packet arrival occurs in $A_0^1$.

$$A_0^0 = \begin{bmatrix} D_2 \otimes V & D_2 \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ 0 & D_2 \otimes S_p(1) & D_2 \otimes (1-p) S_p^0(1) \beta_c(1) & 0 & 0 \\ 0 & 0 & A_0^0(s_1^c s_1^c) & 0 & 0 \\ 0 & 0 & D_2 \otimes p S_p^0(2) \beta_c(1) & D_2 \otimes S_p(2) & A_0^0(s_2^p s_2^c) \\ 0 & 0 & D_2 \otimes p S_c^0(2) \beta_c(1) & 0 & A_0^0(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_0^0(s_1^c s_1^c) &= D_2 \otimes S_c(1) + D_2 \otimes (1-p) S_c^0(1) \beta_c(1) \\ A_0^0(s_2^p s_2^c) &= D_2 \otimes (1-p) S_p^0(2) \beta_c(2) \\ A_0^0(s_2^c s_2^c) &= D_2 \otimes S_c(2) + D_2 \otimes (1-p) S_c^0(2) \beta_c(2) \end{aligned}$$

$$A_0^1 = \begin{bmatrix} D_{11} \otimes V & D_{11} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ 0 & D_{11} \otimes S_p(1) & D_{11} \otimes (1-p) S_p^0(1) \beta_c(1) & 0 & 0 \\ 0 & 0 & A_0^1(s_1^c s_1^c) & 0 & 0 \\ 0 & 0 & D_{11} \otimes p S_p^0(2) \beta_c(1) & D_{11} \otimes S_p(2) & A_0^1(s_2^p s_2^c) \\ 0 & 0 & D_{11} \otimes p S_c^0(2) \beta_c(1) & 0 & A_0^1(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_0^1(s_1^c s_1^c) &= D_{11} \otimes S_c(1) + D_{11} \otimes (1-p) S_c^0(1) \beta_c(1) \\ A_0^1(s_2^p s_2^c) &= D_{11} \otimes (1-p) S_p^0(2) \beta_c(2) \\ A_0^1(s_2^c s_2^c) &= D_{11} \otimes S_c(2) + D_{11} \otimes (1-p) S_c^0(2) \beta_c(2) \end{aligned}$$

$A_1$ shows the case when no change is made in the number of high-priority packets in the queue at the end of current time slot comparing to previous time slot.

$$A_1 = \begin{bmatrix} A_1^1 & A_1^0 & & \\ & \ddots & \ddots & \\ & & A_1^1 & A_1^0 \\ & & & A_1^1 + A_1^0 \end{bmatrix} \tag{3.8}$$

$A_1^0$ and $A_1^1$ are the same as $A_0^0$ and $A_0^1$ respectively in low-priority queue's perspective.

$$A_1^0 = \begin{bmatrix} D_{12} \otimes V & D_{12} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ 0 & D_{12} \otimes S_p(1) & A_1^0(s_1^p s_1^c) & 0 & 0 \\ 0 & 0 & A_1^0(s_1^c s_1^c) & 0 & 0 \\ 0 & 0 & D_{12} \otimes pS_p^0(2)\beta_c(1) & D_{12} \otimes S_p(2) & A_1^0(s_2^p s_2^c) \\ 0 & 0 & D_{12} \otimes pS_c^0(2)\beta_c(1) & 0 & A_1^0(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_1^0(s_1^p s_1^c) &= D_2 \otimes pS_p^0(1)\beta_c(1) + D_{12} \otimes (1-p)S_p^0(1)\beta_c(1) \\ A_1^0(s_1^c s_1^c) &= D_{12} \otimes S_c(1) + D_2 \otimes pS_c^0(1)\beta_c(1) + D_{12} \otimes (1-p)S_c^0(1)\beta_c(1) \\ A_1^0(s_2^p s_2^c) &= D_{12} \otimes (1-p)S_p^0(2)\beta_c(2) \\ A_1^0(s_2^c s_2^c) &= D_{12} \otimes S_c(2) + D_{12} \otimes (1-p)S_c^0(2)\beta_c(2) \end{aligned}$$

$$A_1^1 = \begin{bmatrix} D_0 \otimes V & D_0 \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ 0 & D_0 \otimes S_p(1) & A_1^1(s_1^p s_1^c) & 0 & 0 \\ 0 & 0 & A_1^1(s_1^c s_1^c) & 0 & 0 \\ 0 & 0 & D_0 \otimes pS_p^0(2)\beta_c(1) & D_0 \otimes S_p(2) & A_1^1(s_2^p s_2^c) \\ 0 & 0 & D_0 \otimes pS_c^0(2)\beta_c(1) & 0 & A_1^1(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_1^1(s_1^p s_1^c) &= D_{11} \otimes pS_p^0(1)\beta_c(1) + D_0 \otimes (1-p)S_p^0(1)\beta_c(1) \\ A_1^1(s_1^p s_1^c) &= D_0 \otimes S_c(1) + D_{11} \otimes pS_c^0(1)\beta_c(1) + D_0 \otimes (1-p)S_c^0(1)\beta_c(1) \\ A_1^1(s_2^p s_2^c) &= D_0 \otimes (1-p)S_p^0(2)\beta_c(2) \\ A_1^1(s_2^c s_2^c) &= D_0 \otimes S_c(2) + D_0 \otimes (1-p)S_c^0(2)\beta_c(2) \end{aligned}$$

In $A_2$, number of high-priority packets is decreased by one.

$$A_2 = \begin{bmatrix} A_2^1 & A_2^0 & & \\ & \ddots & \ddots & \\ & & A_2^1 & A_2^0 \\ & & & A_2^1 + A_2^0 \end{bmatrix} \tag{3.9}$$

$A_2^0$ and $A_2^1$ simply represent one arrival and no arrival in low-priority queue respectively while a high-priority packet has been served and server continuing serving high-priority queue.

$$A_2^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_{12} \otimes pS_p^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & D_{12} \otimes pS_c^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad A_2^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_0 \otimes pS_p^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & D_0 \otimes pS_c^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

### 3.3.2 Matrix-Geometric Analysis and Steady-State Probability Distribution

For the transition probability matrix $P$ let $x = [x_0 \; x_1 \; x_2 \; \ldots \; x_K]$ (where $K$ is the size of both queues) represent the steady-state probability vector corresponding to the number of packets in the queue, where $x = xP$, and $x\mathrm{e} = 1$. If matrix $R$ is the minimal non-negative solution to the following matrix quadratic equation:

$$R = A_0 + RA_1 + R^2 A_2 \tag{3.10}$$

It can be shown that [55]

$$x_{i+1} = x_i R, \quad 1 \le i < K \tag{3.11}$$

where

$$\begin{aligned} x_i &= [x_{i0} \; x_{i1} \; x_{i2} \; \ldots \; x_{iK}], \quad 0 \le i \le K \\ x_{ij} &= [x_{ij}^v \; x_{ij}^{s,p}(1) \; x_{ij}^{s,c}(1) \; x_{i,j+1}^{s,p}(2) \; x_{i,j+1}^{s,c}(2)], \quad i \ge 1, j \ge 0. \end{aligned} \tag{3.12}$$

Here, $x_{ij}^{s,p}(1)$ and $x_{ij}^{s,c}(1)$ represent the probability of a type-1 packet being in service while the HOL or a non-HOL packet is being transmitted, respectively. Also, $x_{i,j+1}^{s,p}(2)$ and $x_{i,j+1}^{s,c}(2)$ are the same as $x_{ij}^{s,p}(1)$ and $x_{ij}^{s,c}(1)$, respectively; but, for a type-2 packet in service. For $i = 0$ we have $x_{00} = x_{00}^v$ and $x_{0j} = [x_{0j}^v \; x_{0,j+1}^{s,p} \; x_{0,j+1}^{s,c}]$.

### 3.3.3 Performance Measures

#### 3.3.3.1 Queue Length Distribution

Let $q_i$ be the probability that there are $i$ high-priority packets in the system. Then, we have $q_i = x_i e$.

#### 3.3.3.2 Packet Dropping Probability

Using the steady state probability vector $x_i$, for a maximum queue length of $K$ packets for high-priority packets, the probability of packet dropping ($p_d$) can be found as follows:

$$\begin{aligned} p_d \;=\; & \lambda_1^{-1}\big(\sum_{j=0}^{K}\big(x^{v}_{Kj}[D' \otimes (V + V^0\nu)] + x^{s,p}_{Kj}(1)[D' \otimes p' S^0_p(1)\beta_c(1)] \\ & + x^{s,p}_{Kj}(1)[D' \otimes S_p(1)]\big) + \sum_{j=0}^{K}\big(x^{s,c}_{Kj}(1)[D' \otimes p' S^0_c(1)\beta_c(1)] + x^{s,c}_{Kj}(1)[D' \otimes S_c(1)] \\ & + x^{s,p}_{K,j+1}(2)[D' \otimes p' S^0_p(2)\beta_c(1)]\big) + \sum_{j=0}^{K}\big(x^{s,p}_{K,j+1}(2)[D' \otimes S_p(2)] \\ & + x^{s,c}_{K,j+1}(2)[D' \otimes p' S^0_c(2)\beta_c(1)] + x^{s,c}_{K,j+1}(2)[D' \otimes S_c(2)]\big)\big)e. \end{aligned} \tag{3.13}$$

where $D' = D_{11} + D_2$ and $p' = 1 - p$.

Note that, based on the dropping probability defined above, the queue throughput can be obtained as $\lambda_1(1 - p_d)$.

#### 3.3.3.3 Probability of Sleep (Energy Saving Factor)

Probability of sleep is given by $S = \sum_{i=0}^{K} x^v_i$. Since the longer the time that a node stays in the sleep mode, the more energy is saved, this is an indicator of energy saving in the node.

#### 3.3.3.4 Access Delay Distribution

With the same definition for vector $z$ given in Section 2.3.3.4, we have

$$
\begin{aligned}
z_0^v &= \lambda_1^{-1} x_{00}[D' \otimes (V + V^0\nu)] + \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^v [D' \otimes (V + V^0\nu)] && (3.14)\\
z_{00}^s &= \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,p}[D' \otimes pS_p^0(2)] + \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,c}[D' \otimes pS_c^0(2)] \\
&+ \lambda_1^{-1} \sum_{j=0}^{K} x_{1j}^{s,p}(1)[D' \otimes pS_p^0(1)] + \lambda_1^{-1} \sum_{j=0}^{K} x_{1j}^{s,c}(1)[D' \otimes pS_c^0(1)] && (3.15)\\
z_{01}^{s,p} &= \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,p}[D' \otimes S_p(2)] + \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,p}[D' \otimes p'S_p^0(2)\beta_c(2)] && (3.16)\\
z_{01}^{s,c} &= \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,c}[D' \otimes S_c(2)] + \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,c}[D' \otimes p'S_c^0(2)\beta_c(2)] && (3.17)\\
z_{i0}^v &= \lambda_1^{-1} \sum_{j=0}^{K} x_{ij}^v [D' \otimes V] && (3.18)\\
z_{i0}^{s,p}(1) &= \lambda_1^{-1} \sum_{j=0}^{K} x_{ij}^{s,p}(1)[D' \otimes S_p(1)] + \lambda_1^{-1} \sum_{j=0}^{K} x_{ij}^v [D' \otimes V^0\beta_p(1)] && (3.19)
\end{aligned}
$$

$$
\begin{aligned}
z_{i0}^{s,c}(1) &= \lambda_1^{-1} \sum_{j=0}^{K} \big( x_{ij}^{s,c}(1)[D' \otimes S_c(1)] + x_{ij}^{s,p}(1)[D' \otimes p'S_p^0(1)\beta_c(1)] \\
&+ x_{i+1,j}^{s,c}(1)[D' \otimes pS_c^0(1)\beta_c(1)] + x_{i,j+1}^{s,p}(2)[D' \otimes pS_p^0(2)\beta_c(1)] \\
&+ x_{i,j+1}^{s,c}(2)[D' \otimes pS_c^0(2)\beta_c(1)] + x_{i+1,j}^{s,p}(1)[D' \otimes pS_p^0(1)\beta_c(1)] \\
&+ x_{ij}^{s,c}(1)[D' \otimes p'S_c^0(1)\beta_c(1)] \big) && (3.20)
\end{aligned}
$$

$$
\begin{aligned}
z_{i1}^{s,p}(2) &= \lambda_1^{-1} \sum_{j=1}^{K} x_{ij}^{s,p}(2)[D' \otimes S_p(2)] && (3.21)\\
z_{i1}^{s,c}(2) &= \lambda_1^{-1} \sum_{j=1}^{K} \big( x_{ij}^{s,c}(2)([D' \otimes S_c(2)] + [D' \otimes p'S_c^0(2)\beta_c(2)]) \\
&+ x_{ij}^{s,p}(2)[D' \otimes p'S_p^0(2)\beta_c(2)] \big). && (3.22)
\end{aligned}
$$

Then

$$\begin{aligned} z_0 &= [z_0^v \, z_{00}^s \, z_{01}^{s,p} \, z_{01}^{s,c}] \quad (3.23) \\ z_i &= [z_{i0}^v \, z_{i0}^{s,p}(1) \, z_{i0}^{s,c}(1) \, z_{i1}^{s,p}(2) \, z_{i1}^{s,c}(2)] \quad (3.24) \\ z &= [z_0 \, z_1 \, \ldots \, z_K]. \quad (3.25) \end{aligned}$$

We study the delay distribution of type-1 packet as the time to absorption in a Markov chain with the following transition matrix:

$$\widetilde{P} = \begin{bmatrix} \widetilde{B}_{00} & & & \\ \widetilde{B}_{10} & \widetilde{A}_1 & & \\ & \widetilde{A}_2 & \widetilde{A}_1 & \\ & & \ddots & \ddots \end{bmatrix} \quad (3.26)$$

where the block matrices are as follows:

$$\widetilde{B}_{00} = \begin{bmatrix} V & V^0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & I \otimes pS_p^0(2) & I \otimes S_p(2) & I \otimes (1-p)S_p^0(2)\beta_c(2) \\ 0 & I \otimes pS_c^0(2) & 0 & I \otimes S_c(2) + I \otimes (1-p)S_c^0(2)\beta_c(2) \end{bmatrix}$$

$$\widetilde{B}_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I \otimes S_p^0(1) & 0 & 0 \\ 0 & I \otimes S_c^0(1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\widetilde{A}_1 = \begin{bmatrix} I \otimes V & I \otimes V^0\beta_p(1) & 0 & 0 & 0 \\ 0 & I \otimes S_p(1) & I \otimes (1-p)S_p^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & \widetilde{A}_1(s_1^c s_1^c) & 0 & 0 \\ 0 & 0 & I \otimes pS_p^0(2)\beta_c(1) & I \otimes S_p(2) & \widetilde{A}_1(s_2^p s_2^c) \\ 0 & 0 & I \otimes pS_c^0(2)\beta_c(1) & 0 & \widetilde{A}_1(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned}
\widetilde{A}_1(s_2^c s_2^c) &= I \otimes S_c(2) + I \otimes (1-p) S_c^0(2)\beta_c(2) \\
\widetilde{A}_1(s_2^p s_2^c) &= I \otimes (1-p) S_p^0(2)\beta_c(2) \\
\widetilde{A}_1(s_1^c s_1^c) &= I \otimes S_c(1) + I \otimes (1-p) S_c^0(1)\beta_c(1)
\end{aligned}$$

$$\widetilde{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I \otimes pS_p^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & I \otimes pS_c^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The following set of equations are used in the computation:

$$z_0 = z_0 \widetilde{B}_{00} + z_1 \widetilde{B}_{10} \tag{3.27}$$

$$z_i = z_i \widetilde{A}_1 + z_{i+1} \widetilde{A}_2 \tag{3.28}$$

$$\begin{aligned}
z_0^v &= z_0^v V \qquad (3.29) \\
z_{00}^s &= z_0^v V^0 + z_{00}^s + z_{01}^{s,p}(I \otimes pS_p^0(2)) + z_{01}^{s,c}(I \otimes pS_c^0(2)) \\
&\quad + z_{10}^{s,p}(1)(I \otimes pS_p^0(1)) + z_{10}^{s,c}(1)(I \otimes pS_c^0(1)) \qquad (3.30) \\
z_{01}^{s,p} &= z_{01}^{s,p}(I \otimes S_p(2)) \qquad (3.31) \\
z_{01}^{s,c} &= z_{01}^{s,p}(I \otimes (1-p) S_p^0(2)\beta_c(2)) + z_{01}^{s,c}(I \otimes S_c(2) \\
&\quad + I \otimes (1-p) S_c^0(2)\beta_c(2)) \qquad (3.32)
\end{aligned}$$

$$\begin{aligned}
z_{i0}^v &= z_{i0}^v (I \otimes V) \qquad (3.33) \\
z_{i0}^{s,p}(1) &= z_{i0}^v (I \otimes V^0 \beta_p(1)) + z_{i0}^{s,p}(1)(I \otimes S_p(1)) \qquad (3.34)
\end{aligned}$$

$$\begin{aligned} z_{i0}^{s,c}(1) &= z_{i0}^{s,p}(1)(I \otimes (1-p)S_p^0(1)\beta_c(1)) + I \otimes (1-p)S_c^0(1)\beta_c(1)) \\ &+z_{i1}^{s,p}(2)(I \otimes pS_p^0(2)\beta_c(1)) + z_{i0}^{s,c}(1)(I \otimes S_c(1) + z_{i1}^{s,c}(2)(I \otimes pS_c^0(2)\beta_c(1)) \\ &+z_{i+1,0}^{s,p}(1)(I \otimes pS_p^0(1)\beta_c(1)) + z_{i+1,0}^{s,c}(1)(I \otimes pS_c^0(1)\beta_c(1)) \qquad (3.35) \end{aligned}$$

$$\begin{aligned} z_{i1}^{s,p}(2) &= z_{i1}^{s,p}(2)(I \otimes S_p(2)) \qquad (3.36) \\ z_{i1}^{s,c}(2) &= z_{i1}^{s,p}(2)(I \otimes (1-p)S_p^0(2)\beta_c(2)) + z_{i1}^{s,c}(2)(I \otimes S_c(2) \\ &+I \otimes (1-p)S_c^0(2)\beta_c(2)). \qquad (3.37) \end{aligned}$$

If $W_T$ denotes the probability that the delay of a type-1 packet is less than or equal to $T$, then $W_T = z_0^T e$.

## 3.4 Number-Limited Exhaustive Vacation System

We repeat the analysis for number-limited vacation mechanism. As has been mentioned before, the transmission latency for the first packet in transmission after the idle period (i.e., the HOL packet) includes the setup time and is different from the service time of each of the following packets. Moreover, probability of successful transmission is assumed to be $p$.

### 3.4.1 The Markov Chain

The state space in this case is as follows:

$$\begin{aligned} \Omega_0^v &= (0,0,j,l) \\ \Omega_0^s &= (0,i_2,u,j,k_2), \quad K \geq i_2 \geq 1 \\ \Omega^v &= (i_1,i_2,j,l), \quad (i_1+i_2) \geq 1,\ K \geq i_1 \geq 1,\ K \geq i_2 \geq 0 \\ \Omega_1^s &= (i_1,i_2,u,j,k_1), \quad K \geq i_1 \geq 1,\ K \geq i_2 \geq 0 \\ \Omega_2^s &= (i_1,i_2+1,u,j,k_2), \quad K \geq i_1 \geq 1,\ K \geq i_2 \geq 0. \qquad (3.38) \end{aligned}$$

This is the same as the standard exhaustive case in Section 3.3 except that the number of the served packets in the state space is different. The vacation states are exactly the same as before. In $\Omega_0^s, \Omega_1^s$ and $\Omega_2^s$, $u$ denotes the number of served packets ($u = 0, 1, 2, \ldots, M$). The server goes to vacation (sleep mode) after serving $M$ packets even if there are more packets waiting to be served in the system.

The transition matrix $P$ describing this Markov chain is the same as (3.3). However, the matrix blocks are different. Here, $e$ is a column vector of 1s whose length is $M$ and $e_j$ is a column vector of zeros except a 1 as its $j^{th}$ element and its length is $M$, and $\acute{e}_j$ is the transpose of $e_j$. Moreover, state $\Omega_0^s$ is partitioned into $\Omega_0^{s,p}$ and $\Omega_0^{s,c}$ to include and exclude the setup time, respectively. The same concept is true for $\Omega_1^s$ and $\Omega_2^s$ and they are partitioned into $\Omega_1^{s,p}, \Omega_1^{s,c}$ and $\Omega_2^{s,p}, \Omega_2^{s,c}$, respectively. The block matrices are given as follows.

$$B_{00}^{00} = D_0 \quad B_{00}^{01} = \left[ \begin{array}{cc} D_{12} \otimes V & \acute{e}_1 \otimes D_{12} \otimes V^0 \beta_p(2) 0 \end{array} \right]$$

$$B_{00}^{10} = \left[ \begin{array}{c} 0 \\ e \otimes D_0 \otimes pS_p^0(2)\nu \\ e \otimes D_0 \otimes pS_c^0(2)\nu \end{array} \right]$$

$$B_{00}^{2} = \left[ \begin{array}{ccc} 0 & 0 & 0 \\ e_M \otimes D_0 \otimes pS_p^0(2)\nu & 0 & \bar{I}(M-1) \otimes D_0 \otimes pS_p^0(2)\beta_c(2) \\ e_M \otimes D_0 \otimes pS_c^0(2)\nu & 0 & \bar{I}(M-1) \otimes D_0 \otimes pS_c^0(2)\beta_c(2) \end{array} \right]$$

$$B_{00}^{1} = \left[ \begin{array}{ccc} D_0 \otimes V & \acute{e}_1 \otimes D_0 \otimes V^0 \beta_p(2) & 0 \\ e_M \otimes D_{12} \otimes pS_p^0(2)\nu & I(M) \otimes D_0 \otimes S_p(2) & B_{00}^1(s_2^p s_2^c) \\ e_M \otimes D_{12} \otimes pS_p^0(2)\nu & 0 & B_{00}^1(s_2^c s_2^c) \end{array} \right]$$

$$\begin{array}{rcl} B_{00}^1(s_2^p s_2^c) & = & I(M) \otimes D_0 \otimes (1-p)S_p^0(2)\beta_c(2) + \bar{I}(M-1) \otimes D_{12} \otimes pS_p^0(2)\beta_c(2) \\ B_{00}^1(s_2^c s_2^c) & = & I(M) \otimes D_0 \otimes S_c(2) + I(M) \otimes D_0 \otimes (1-p)S_c^0(2)\beta_c(2) \\ & & +\bar{I}(M-1) \otimes D_{12} \otimes pS_c^0(2)\beta_c(2) \end{array}$$

$$B_{00}^{0} = \begin{bmatrix} D_{12} \otimes V & \acute{e}_1 \otimes D_{12} \otimes V^0 \beta_p(2) & 0 \\ 0 & I(M) \otimes D_{12} \otimes S_p(2) & I(M) \otimes D_{12} \otimes (1-p) S_p^0(2) \beta_c(2) \\ 0 & 0 & B_{00}^{0}(s_2^c s_2^c) \end{bmatrix}$$

$$B_{00}^{0}(s_2^c s_2^c) = I(M) \otimes D_{12} \otimes S_c(2) + I(M) \otimes D_{12} \otimes (1-p) S_c^0(2) \beta_c(2)$$

$$B_{01}^{00} = \begin{bmatrix} D_{11} \otimes V & \acute{e}_1 \otimes D_{11} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \end{bmatrix}$$

$$B_{01}^{01} = \begin{bmatrix} D_{2} \otimes V & \acute{e}_1 \otimes D_{2} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \end{bmatrix}$$

$$B_{01}^{0} = \begin{bmatrix} D_{2} \otimes V & \acute{e}_1 \otimes D_{2} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_{01}^{1} = \begin{bmatrix} D_{11} \otimes V & \acute{e}_1 \otimes D_{11} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ e_M \otimes D_2 \otimes p S_p^0(2) \nu & 0 & B_{01}^{1}(s_2^p s_1^c) & & B_{01}^{1}(s_2^p s_2^c) \\ e_M \otimes D_2 \otimes p S_c^0(2) \nu & 0 & B_{01}^{1}(s_2^c s_1^c) & 0 & B_{01}^{1}(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} B_{01}^{1}(s_2^p s_1^c) &= \bar{I}(M-1) \otimes D_2 \otimes p S_p^0(2) \beta_c(1) \\ B_{01}^{1}(s_2^c s_1^c) &= \bar{I}(M-1) \otimes D_2 \otimes p S_c^0(2) \beta_c(1) \end{aligned}$$

$$\begin{aligned} B_{01}^{1}(s_2^p s_2^p) &= I(M) \otimes D_2 \otimes S_p(2) \\ B_{01}^{1}(s_2^p s_2^c) &= I(M) \otimes D_2 \otimes (1-p) S_p^0(2) \beta_c(2) \\ B_{01}^{1}(s_2^c s_2^c) &= I(M) \otimes D_2 \otimes (1-p) S_c^0(2) \beta_c(2) + I(M) \otimes D_2 \otimes S_c(2) \end{aligned}$$

$$B_{01}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ e_M \otimes D_{11} \otimes pS_p^0(2)\nu & 0 & B_{01}^2(s_2^p s_1^c) & B_{01}^2(s_2^p s_2^p) & B_{01}^2(s_2^p s_2^c) \\ e_M \otimes D_{11} \otimes pS_p^0(2)\nu & 0 & B_{01}^2(s_2^c s_1^c) & 0 & B_{01}^2(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} B_{01}^2(s_2^p s_1^c) &= \bar{I}(M-1) \otimes D_{11} \otimes pS_p^0(2)\beta_c(1) \\ B_{01}^2(s_2^c s_1^c) &= \bar{I}(M-1) \otimes D_{11} \otimes pS_c^0(2)\beta_c(1) \\ B_{01}^2(s_2^p s_2^p) &= I(M) \otimes D_{11} \otimes S_p(2) \\ B_{01}^2(s_2^p s_2^c) &= I(M) \otimes D_{11} \otimes (1-p)S_p^0(2)\beta_c(2) \\ B_{01}^2(s_2^c s_2^c) &= I(M) \otimes D_{11} \otimes S_c(2) + I(M) \otimes D_{11} \otimes (1-p)S_c^0(2)\beta_c(2) \end{aligned}$$

$$B_{10}^{00} = \begin{bmatrix} 0 \\ e \otimes D_0 \otimes pS_p^0(1)\nu \\ e \otimes D_0 \otimes pS_c^0(1)\nu \\ 0 \\ 0 \end{bmatrix}$$

$$B_{10}^0 = \begin{bmatrix} 0 & 0 & 0 \\ e_M \otimes D_{12} \otimes pS_p^0(2)\nu & 0 & \bar{I}(M-1) \otimes D_{12} \otimes pS_p^0(1)\beta_c(2) \\ e_M \otimes D_{12} \otimes pS_c^0(2)\nu & 0 & \bar{I}(M-1) \otimes D_{12} \otimes pS_c^0(1)\beta_c(2) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_{10}^1 = \begin{bmatrix} 0 & 0 & 0 \\ e_M \otimes D_0 \otimes pS_p^0(2)\nu & 0 & \bar{I}(M-1) \otimes D_0 \otimes pS_p^0(1)\beta_c(2) \\ e_M \otimes D_0 \otimes pS_p^0(2)\nu & 0 & \bar{I}(M-1) \otimes D_0 \otimes pS_c^0(1)\beta_c(2) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_0^0 = \begin{bmatrix} D_2 \otimes V & e'_1 \otimes D_2 \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ 0 & I(M) \otimes D_2 \otimes S_p(1) & A_0^0(s_1^p s_1^c) & 0 & 0 \\ 0 & 0 & A_0^0(s_1^c s_1^c) & 0 & 0 \\ e_M \otimes D_2 \otimes pS_p^0(2)\nu & 0 & A_0^0(s_2^p s_1^c) & A_0^0(s_2^p s_2^p) & A_0^0(s_2^p s_2^c) \\ e_M \otimes D_2 \otimes pS_c^0(2)\nu & 0 & A_0^0(s_2^c s_1^c) & 0 & A_0^0(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_0^0(s_2^p s_2^p) &= I(M) \otimes D_2 \otimes S_p(2) \\ A_0^0(s_2^p s_2^c) &= I(M) \otimes D_2 \otimes (1-p) S_p^0(2) \beta_c(2) \\ A_0^0(s_2^c s_2^c) &= I(M) \otimes D_2 \otimes S_c(2) + I(M) \otimes D_2 \otimes (1-p) S_c^0(2) \beta_c(2) \end{aligned}$$

$$\begin{aligned} A_0^0(s_1^p s_1^c) &= I(M) \otimes D_2 \otimes (1-p) S_p^0(1) \beta_c(1) \\ A_0^0(s_1^c s_1^c) &= I(M) \otimes D_2 \otimes S_c(1) + I(M) \otimes D_2 \otimes (1-p) S_c^0(1) \beta_c(1) \\ A_0^0(s_2^p s_1^c) &= \bar{I}(M-1) \otimes D_2 \otimes pS_p^0(2) \beta_c(1) \\ A_0^0(s_2^c s_1^c) &= \bar{I}(M-1) \otimes D_2 \otimes pS_c^0(2) \beta_c(1) \end{aligned}$$

$$A_0^1 = \begin{bmatrix} D_{11} \otimes V & e'_1 \otimes D_{11} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ 0 & I(M) \otimes D_{11} \otimes S_p(1) & A_0^1(s_1^p s_1^c) & 0 & 0 \\ 0 & 0 & A_0^1(s_1^c s_1^c) & 0 & 0 \\ e_M \otimes D_{11} \otimes pS_p^0(2)\nu & 0 & A_0^1(s_2^p s_1^c) & A_0^1(s_2^p s_2^p) & A_0^1(s_2^p s_2^c) \\ e_M \otimes D_{11} \otimes pS_c^0(2)\nu & 0 & A_0^1(s_2^c s_1^c) & 0 & A_0^1(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_0^1(s_1^p s_1^c) &= I(M) \otimes D_{11} \otimes (1-p) S_p^0(1) \beta_c(1) \\ A_0^1(s_1^c s_1^c) &= I(M) \otimes D_{11} \otimes S_c(1) + I(M) \otimes D_{11} \otimes (1-p) S_c^0(1) \beta_c(1) \\ A_0^1(s_2^p s_1^c) &= \bar{I}(M-1) \otimes D_{11} \otimes pS_p^0(2) \beta_c(1) \\ A_0^1(s_2^c s_1^c) &= \bar{I}(M-1) \otimes D_{11} \otimes pS_c^0(2) \beta_c(1) \end{aligned}$$

$$\begin{aligned} A_0^1(s_2^p s_2^p) &= I(M) \otimes D_{11} \otimes S_p(2) \\ A_0^1(s_2^p s_2^c) &= I(M) \otimes D_{11} \otimes (1-p) S_p^0(2) \beta_c(2) \\ A_0^1(s_2^c s_2^c) &= I(M) \otimes D_{11} \otimes S_c(2) + I(M) \otimes D_{11} \otimes (1-p) S_c^0(2) \beta_c(2) \end{aligned}$$

$$A_1^0 = \begin{bmatrix} D_{12} \otimes V & \acute{e}_1 \otimes D_{12} \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ e_M \otimes D_2 \otimes pS_p^0(1)\nu & I(M) \otimes D_{12} \otimes S_p(1) & A_1^0(s_1^p s_1^c) & 0 & 0 \\ e_M \otimes D_2 \otimes pS_c^0(1)\nu & 0 & A_1^0(s_1^c s_1^c) & 0 & 0 \\ e_M \otimes D_{12} \otimes pS_p^0(2)\nu & 0 & A_1^0(s_2^p s_1^c) & A_1^0(s_2^p s_2^p) & A_1^0(s_2^p s_2^c) \\ e_M \otimes D_{12} \otimes pS_p^0(2)\nu & 0 & A_1^0(s_2^c s_1^c) & 0 & A_1^0(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_1^0(s_1^p s_1^c) &= \bar{I}(M-1) \otimes D_2 \otimes pS_p^0(1)\beta_c(1) + I(M) \otimes D_{12} \otimes (1-p)S_p^0(1)\beta_c(1) \\ A_1^0(s_1^c s_1^c) &= I(M) \otimes D_{12} \otimes S_c(1) + \bar{I}(M-1) \otimes D_2 \otimes pS_c^0(1)\beta_c(1) \\ &\quad + I(M) \otimes D_{12} \otimes (1-p)S_c^0(1)\beta_c(1) \\ A_1^0(s_2^p s_1^c) &= \bar{I}(M-1) \otimes D_{12} \otimes pS_p^0(2)\beta_c(1) \\ A_1^0(s_2^c s_1^c) &= \bar{I}(M-1) \otimes D_{12} \otimes pS_c^0(2)\beta_c(1) \end{aligned}$$

$$\begin{aligned} A_1^0(s_2^p s_2^p) &= I(M) \otimes D_{12} \otimes S_p(2) \\ A_1^0(s_2^p s_2^c) &= I(M) \otimes D_{12} \otimes (1-p)S_p^0(2)\beta_c(2) \\ A_1^0(s_2^c s_2^c) &= I(M) \otimes D_{12} \otimes S_c(2) + I(M) \otimes D_{12} \otimes (1-p)S_c^0(2)\beta_c(2) \end{aligned}$$

$$A_1^1 = \begin{bmatrix} D_0 \otimes V & \acute{e}_1 \otimes D_0 \otimes V^0 \beta_p(1) & 0 & 0 & 0 \\ e_M \otimes D_{11} \otimes pS_p^0(1)\nu & I(M) \otimes D_0 \otimes S_p(1) & A_1^1(s_1^p s_1^c) & 0 & 0 \\ e_M \otimes D_{11} \otimes pS_c^0(1)\nu & 0 & A_1^1(s_1^c s_1^c) & 0 & 0 \\ e_M \otimes D_0 \otimes pS_p^0(2)\nu & 0 & A_1^1(s_2^p s_1^c) & A_1^1(s_2^p s_2^p) & A_1^1(s_2^p s_2^c) \\ e_M \otimes D_0 \otimes pS_c^0(2)\nu & 0 & A_1^1(s_2^c s_1^c) & 0 & A_1^1(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} A_1^1(s_1^p s_1^c) &= \bar{I}(M-1) \otimes D_{11} \otimes pS_p^0(1)\beta_c(1) + I(M) \otimes D_0 \otimes (1-p)S_p^0(1)\beta_c(1) \\ A_1^1(s_1^c s_1^c) &= I(M) \otimes D_0 \otimes S_c(1) + \bar{I}(M-1) \otimes D_{11} \otimes pS_c^0(1)\beta_c(1) \\ &\quad + I(M) \otimes D_0 \otimes (1-p)S_c^0(1)\beta_c(1) \\ A_1^1(s_2^p s_1^c) &= \bar{I}(M-1) \otimes D_0 \otimes pS_p^0(2)\beta_c(1) \\ A_1^1(s_2^c s_1^c) &= \bar{I}(M-1) \otimes D_0 \otimes pS_c^0(2)\beta_c(1) \end{aligned}$$

$$
\begin{aligned}
A_1^1(s_2^p s_2^p) &= I(M) \otimes D_0 \otimes S_p(2) \\
A_1^1(s_2^p s_2^c) &= I(M) \otimes D_0 \otimes (1-p) S_p^0(2) \beta_c(2) \\
A_1^1(s_2^c s_2^c) &= I(M) \otimes D_0 \otimes S_c(2) + I(M) \otimes D_0 \otimes (1-p) S_c^0(2) \beta_c(2)
\end{aligned}
$$

$$
A_2^0 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
e_M \otimes D_{12} \otimes pS_p^0(1)\nu & 0 & \bar{I}(M-1) \otimes D_{12} \otimes pS_p^0(1)\beta_c(1) & 0 & 0 \\
e_M \otimes D_{12} \otimes pS_c^0(1)\nu & 0 & \bar{I}(M-1) \otimes D_{12} \otimes pS_c^0(1)\beta_c(1) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$
A_2^1 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
e_M \otimes D_0 \otimes pS_p^0(1)\nu & 0 & \bar{I}(M-1) \otimes D_0 \otimes pS_p^0(1)\beta_c(1) & 0 & 0 \\
e_M \otimes D_0 \otimes pS_c^0(1)\nu & 0 & \bar{I}(M-1) \otimes D_0 \otimes pS_c^0(1)\beta_c(1) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}.
$$

The matrix-geometric analysis for obtaining the steady state distribution for the number-limited exhaustive vacation system (using the new block matrices of **P**) is the same as that for the standard system.

### 3.4.2 Performance Measures

The performance measures are obtained in the same way as that for the standard exhaustive case. For access delay distribution we proceed in the same way with the new matrices and obtain the probability vectors as follows where $D' = D_{11} + D_2$.

$$
\begin{aligned}
z_0^v &= \lambda_1^{-1} x_{00}[D' \otimes (V + V^0\nu)] + \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^v [D' \otimes (V + V^0\nu)] \\
&+ \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,p} [e_M \otimes D' \otimes pS_p^0(2)\nu] + \lambda_1^{-1} \sum_{j=1}^{K} x_{0j}^{s,c} [e_M \otimes D' \otimes pS_c^0(2)\nu] \\
&+ \lambda_1^{-1} \sum_{j=1}^{K} x_{1j}^{s,p} [e_M \otimes D' \otimes pS_p^0(1)\nu] + \lambda_1^{-1} \sum_{j=1}^{K} x_{1j}^{s,c} [e_M \otimes D' \otimes pS_c^0(1)\nu] (3.39)
\end{aligned}
$$

$$\begin{aligned} z_{00}^{s} &= \lambda_1^{-1}\sum_{j=1}^{K} x_{0j}^{s,p}[\bar{I}(M-1)\otimes D'\otimes pS_p^0(2)] + \lambda_1^{-1}\sum_{j=1}^{K} x_{0j}^{s,c}[\bar{I}(M-1)\otimes D'\otimes pS_c^0(2)] \\ &+ \lambda_1^{-1}\sum_{j=0}^{K} x_{1j}^{s,p}(1)[\bar{I}(M-1)\otimes D'\otimes pS_p^0(1)] \\ &+ \lambda_1^{-1}\sum_{j=0}^{K} x_{1j}^{s,c}(1)[\bar{I}(M-1)\otimes D'\otimes pS_c^0(1)] \end{aligned} \tag{3.40}$$

$$\begin{aligned} z_{01}^{s,p} &= \lambda_1^{-1}\sum_{j=1}^{K} x_{0j}^{s,p}[I(M)\otimes D'\otimes S_p(2)] \\ z_{01}^{s,c} &= \lambda_1^{-1}\sum_{j=1}^{K}(x_{0j}^{s,c}[I(M)\otimes D'\otimes S_c(2)] + x_{0j}^{s,p}[I(M)\otimes D'\otimes p'S_p^0(2)\beta_c(2)] \\ &+ x_{0j}^{s,c}[I(M)\otimes D'\otimes p'S_c^0(2)\beta_c(2)]) \end{aligned} \tag{3.41}$$

$$\begin{aligned} z_{i0}^{v} &= \lambda_1^{-1}\sum_{j=0}^{K}\big(x_{ij}^{v}[D'\otimes V] + x_{ij+1}^{s,p}(2)[e_M\otimes D'\otimes pS_p^0(2)\nu] \\ &\quad +x_{ij+1}^{s,c}(2)[e_M\otimes D'\otimes pS_c^0(2)\nu] + x_{i+1j}^{s,p}(1)[e_M\otimes D'\otimes pS_p^0(1)\nu] \\ &\quad +x_{i+1j}^{s,c}(1)[e_M\otimes D'\otimes pS_c^0(1)\nu]\big) \end{aligned} \tag{3.42}$$

$$\begin{aligned} z_{i0}^{s,p}(1) &= \lambda_1^{-1}\sum_{j=0}^{K} x_{ij}^{s,p}(1)[I(M)\otimes D'\otimes S_p(1)] \\ &\quad +\lambda_1^{-1}\sum_{j=0}^{K} x_{ij}^{v}[e\acute{}_1\otimes D'\otimes V^0\beta_p(1)] \end{aligned} \tag{3.43}$$

$$\begin{aligned}
z_{i0}^{s,c}(1) &= \lambda_1^{-1}\sum_{j=0}^{K}(x_{ij}^{s,c}(1)[I(M)\otimes D'\otimes S_c(1)] \\
&+\lambda_1^{-1}\sum_{j=0}^{K}x_{ij}^{s,p}(1)[I(M)\otimes D'\otimes p'S_p^0(1)\beta_c(1)] \\
&+\lambda_1^{-1}\sum_{j=0}^{K}x_{ij}^{s,c}(1)[I(M)\otimes D'\otimes p'S_c^0(1)\beta_c(1)] \\
&+\lambda_1^{-1}\sum_{j=0}^{K}x_{ij+1}^{s,p}(2)[\bar{I}(M-1)\otimes D'\otimes pS_p^0(2)\beta_c(1)] \\
&+\lambda_1^{-1}\sum_{j=0}^{K}x_{ij+1}^{s,c}(2)[\bar{I}(M-1)\otimes D'\otimes pS_c^0(2)\beta_c(1)] \\
&+\lambda_1^{-1}\sum_{j=0}^{K}x_{i+1j}^{s,p}(1)[\bar{I}(M-1)\otimes D'\otimes pS_p^0(1)\beta_c(1)] \\
&+\lambda_1^{-1}\sum_{j=0}^{K}x_{i+1j}^{s,c}(1)[\bar{I}(M-1)\otimes D'\otimes pS_c^0(1)\beta_c(1)] \qquad (3.44)
\end{aligned}$$

$$\begin{aligned}
z_{i1}^{s,p}(2) &= \lambda_1^{-1}\sum_{j=1}^{K}x_{ij}^{s,p}(2)[I(M)\otimes D'\otimes S_p(2)] \\
z_{i1}^{s,c}(2) &= \lambda_1^{-1}\sum_{j=1}^{K}x_{ij}^{s,c}(2)[I(M)\otimes D'\otimes S_c(2)] \\
&+\lambda_1^{-1}\sum_{j=1}^{K}x_{ij}^{s,c}(2)[I(M)\otimes D'\otimes(1-p)S_c^0(2)\beta_c(2)] \\
&+\lambda_1^{-1}\sum_{j=1}^{K}x_{ij}^{s,p}(2)[I(M)\otimes D'\otimes(1-p)S_p^0(2)\beta_c(2)]. \qquad (3.45)
\end{aligned}$$

Then

$$\begin{aligned}
z_0 &= [z_0^v\ z_{00}^s\ z_{01}^{s,p}\ z_{01}^{s,c}], \\
z_i &= [z_{i0}^v\ z_{i0}^{s,p}(1)\ z_{i0}^{s,c}(1)\ z_{i1}^{s,p}(2)\ z_{i1}^{s,c}(2)], \\
z &= [z_0\ z_1\ \ldots\ z_K]. \qquad (3.46)
\end{aligned}$$

We evaluate the delay distribution for a type-1 packet as the time to absorption in a Markov chain with the same transition matrix as in (3.26). The block matrices are described as follows.

$$\widetilde{B}_{00} = \begin{bmatrix} V & \acute{e}_1 \otimes V^0 & 0 & 0 \\ 0 & I(M) \otimes I & 0 & 0 \\ e_M \otimes I \otimes pS_p^0(2)\nu & \bar{I}(M-1) \otimes I \otimes pS_p^0(2) & I(M) \otimes I \otimes S_p(2) & \widetilde{B}_{00}(s_2^p s_2^c) \\ e_M \otimes I \otimes pS_c^0(2)\nu & \bar{I}(M-1) \otimes I \otimes pS_c^0(2) & 0 & \widetilde{B}_{00}(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} \widetilde{B}_{00}(s_2^p s_2^c) &= I(M) \otimes I \otimes (1-p)S_p^0(2)\beta_c(2) \\ \widetilde{B}_{00}(s_2^c s_2^c) &= I(M) \otimes I \otimes S_c(2) + I(M) \otimes I \otimes (1-p)S_c^0(2)\beta_c(2) \end{aligned}$$

$$\widetilde{B}_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ e_M \otimes I \otimes pS_p^0(1)\nu & \bar{I}(M-1) \otimes I \otimes pS_p^0(1) & 0 & 0 \\ e_M \otimes I \otimes pS_p^0(1)\nu & \bar{I}(M-1) \otimes I \otimes pS_c^0(1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\widetilde{A}_1 = \begin{bmatrix} I \otimes V & \acute{e}_1 \otimes I \otimes V^0\beta_p(1) & 0 & 0 & 0 \\ 0 & I(M) \otimes I \otimes S_p(1) & \widetilde{A}_1(s_1^p s_1^c) & 0 & 0 \\ 0 & 0 & \widetilde{A}_1(s_1^c s_1^c) & 0 & 0 \\ e_M \otimes I \otimes pS_p^0(2)\nu & 0 & \widetilde{A}_1(s_2^p s_1^c) & \widetilde{A}_1(s_2^p s_2^p) & \widetilde{A}_1(s_2^p s_2^c) \\ e_M \otimes I \otimes pS_c^0(2)\nu & 0 & \widetilde{A}_1(s_2^c s_1^c) & 0 & \widetilde{A}_1(s_2^c s_2^c) \end{bmatrix}$$

$$\begin{aligned} \widetilde{A}_1(s_1^p s_1^c) &= I(M) \otimes I \otimes (1-p)S_p^0(1)\beta_c(1) \\ \widetilde{A}_1(s_1^c s_1^c) &= I(M) \otimes I \otimes S_c(1) + I(M) \otimes I \otimes (1-p)S_c^0(1)\beta_c(1) \\ \widetilde{A}_1(s_2^p s_1^c) &= \bar{I}(M-1) \otimes I \otimes pS_p^0(2)\beta_c(1) \\ \widetilde{A}_1(s_2^c s_1^c) &= \bar{I}(M-1) \otimes I \otimes pS_c^0(2)\beta_c(1) \end{aligned}$$

$$\begin{aligned} \widetilde{A}_1(s_2^p s_2^c) &= I(M) \otimes I \otimes (1-p)S_p^0(2)\beta_c(2) \\ \widetilde{A}_1(s_2^p s_2^p) &= I(M) \otimes I \otimes S_p(2) \\ \widetilde{A}_1(s_2^c s_2^c) &= I(M) \otimes I \otimes S_c(2) + I(M) \otimes I \otimes (1-p)S_c^0(2)\beta_c(2) \end{aligned}$$

$$\widetilde{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ e_M \otimes I \otimes pS_p^0(1)\nu & 0 & \bar{I}(M-1) \otimes I \otimes pS_p^0(1)\beta_c(1) & 0 & 0 \\ e_M \otimes I \otimes pS_p^0(1)\nu & 0 & \bar{I}(M-1) \otimes I \otimes pS_c^0(1)\beta_c(1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Then

$$z^{n+1} = z^n \widetilde{P}, \quad W_T = z_0^T e. \tag{3.47}$$

The following set of equations are used in the computation.

$$z_0 = z_0 \widetilde{B}_{00} + z_1 \widetilde{B}_{10} \tag{3.48}$$

$$z_i = z_i \widetilde{A}_1 + z_{i+1} \widetilde{A}_2 \tag{3.49}$$

$$\begin{aligned} z_0^v &= z_0^v V + z_{01}^{s,p}(e_M \otimes I \otimes pS_p^0(2)\nu) + z_{01}^{s,c}(e_M \otimes I \otimes pS_c^0(2)\nu) \\ &\quad + z_{10}^{s,p}(1)(e_M \otimes I \otimes pS_p^0(1)\nu) + z_{10}^{s,c}(1)(e_M \otimes I \otimes pS_c^0(1)\nu) \qquad (3.50) \\ z_{00}^s &= z_0^v(e_1' \otimes V^0) + z_{00}^s(I(M) \otimes I) + z_{01}^{s,p}(\bar{I}(M-1) \otimes I \otimes pS_p^0(2)) \\ &\quad + z_{01}^{s,c}(\bar{I}(M-1) \otimes I \otimes pS_c^0(2)) + z_{10}^{s,p}(1)(\bar{I}(M-1) \otimes I \otimes pS_p^0(1)) \\ &\quad + z_{10}^{s,c}(1)(\bar{I}(M-1) \otimes I \otimes pS_c^0(1)) \qquad (3.51) \end{aligned}$$

$$\begin{aligned} z_{01}^{s,p} &= z_{01}^{s,p}(I(M) \otimes I \otimes S_p(2)) \\ z_{01}^{s,c} &= z_{01}^{s,p}(I(M) \otimes I \otimes p'S_p^0(2)\beta_c(2)) + z_{01}^{s,c}(I(M) \otimes I \otimes S_c(2) \\ &\quad + I(M) \otimes I \otimes p'S_c^0(2)\beta_c(2)) \qquad (3.52) \end{aligned}$$

$$\begin{aligned} z_{i0}^v &= z_{i0}^v(I \otimes V) + z_{i1}^{s,p}(2)(e_M \otimes I \otimes pS_p^0(2)\nu) + z_{i1}^{s,c}(2)(e_M \otimes I \otimes pS_c^0(2)\nu) \\ &+ \; z_{i+1,0}^{s,p}(1)(e_M \otimes I \otimes pS_p^0(1)\nu) + z_{i+1,0}^{s,c}(1)(e_M \otimes I \otimes pS_c^0(1)\nu) \qquad (3.53) \\ z_{i0}^{s,p}(1) &= z_{i0}^v(e_1' \otimes I \otimes V^0\beta_p(1)) + z_{i0}^{s,p}(1)(I(M) \otimes I \otimes S_p(1)) \qquad (3.54) \end{aligned}$$

$$\begin{aligned}
z_{i0}^{s,c}(1) &= z_{i0}^{s,p}(1)(I(M) \otimes I \otimes (1-p)S_p^0(1)\beta_c(1)) \\
&+ z_{i0}^{s,c}(1)(I(M) \otimes I \otimes S_c(1) + I \otimes (1-p)S_c^0(1)\beta_c(1)) \\
&+ z_{i1}^{s,p}(2)(\bar{I}(M-1) \otimes I \otimes pS_p^0(2)\beta_c(1)) \\
&+ z_{i1}^{s,c}(2)(\bar{I}(M-1) \otimes I \otimes pS_c^0(2)\beta_c(1)) \\
&+ z_{i+1,0}^{s,p}(1)(\bar{I}(M-1) \otimes I \otimes pS_p^0(1)\beta_c(1)) \\
&+ z_{i+1,0}^{s,c}(1)(\bar{I}(M-1) \otimes I \otimes pS_c^0(1)\beta_c(1)) \qquad (3.55)
\end{aligned}$$

$$\begin{aligned}
z_{i1}^{s,p}(2) &= z_{i1}^{s,p}(2)(I(M) \otimes I \otimes S_p(2)) \qquad (3.56) \\
z_{i1}^{s,c}(2) &= z_{i1}^{s,p}(2)(I(M) \otimes I \otimes p'S_p^0(2)\beta_c(2)) + z_{i1}^{s,c}(2)(I(M) \otimes I \otimes S_c(2) \\
&+ I(M) \otimes I \otimes p'S_c^0(2)\beta_c(2)). \qquad (3.57)
\end{aligned}$$

## 3.5 Application of the Analytical Model

As an application of the presented analytical model, we use it for performance evaluation of predictive p-persistent CSMA protocol [70]. This protocol is a variant of p-persistent CSMA with the difference that when the channel is idle, the probability of packet transmission varies according to the traffic condition (instead of being constant as in p-persistent CSMA).

The service-time distribution in this protocol is geometric (a special case of phase-type distribution) with parameter $p_x$ as the successful access to shared medium. In other words, when a node is in active mode and it has data packets to be transmitted and the channel is idle, it can successfully transmit with probability $p_x$. Therefore, $S_p = S_c = p_x$ and $S_p^\circ = S_c^\circ = 1 - p_x$ .

When either the channel is busy or there is no packet for transmission, the server goes to sleep mode. This is similar to adopting a back-off mechanism to avoid collision as well as to save energy. We model this back-off mechanism in a node as a geometric distribution in which the back-off ends in each node with probability $\gamma$ (i.e., the node returns to active mode with probability $\gamma$).

The value of $p_x$ is not the same for all nodes in the network. The nodes compute the appropriate values of $p_x$ and transmit with different probabilities depending on the number of the nodes present in the system. At the beginning of each contention period, node $i$ decides to transmit with probability $p_i$.

To obtain the access probability, we follow [71] and define utility function of node $i$ as follows:

$$u_i = p_i \times \prod_{j \neq i} p_{w,j} \tag{3.58}$$

where $p_{w,j}$ is the probability that node $j$ does not transmit a packet. It happens when either the node has no packet to transmit or it has a packet and decides not to transmit. Defining $p_{p,j}$ as the probability of having packet for transmission at the beginning of a time slot in user $j$, the probability $p_{w,j}$ is given as follows [71]:

$$p_{w,j} = p_{p,j} \times (1 - p_j) + (1 - p_{p,j}) = 1 - p_{p,j} p_j. \tag{3.59}$$

In a network with $N$ nodes we have

$$u_i = \frac{p_i}{1 - p_{p,i} p_i} \prod_{j=1}^{N} p_{w,j} \tag{3.60}$$

and the probability that a node $i$ accesses the shared medium successfully is

$$p_{x,i} = \frac{p_{p,i} p_i}{1 - p_{p,i} p_i} \prod_{j=1}^{N} p_{w,j}. \tag{3.61}$$

To model this sleep and wakeup dynamics a 1-phase PH distribution can be considered with $V = 1 - \gamma$ and $V^{\circ} = \gamma$. We refer to $\mu$ and $\gamma$ as the probability of service and the probability of wakeup, respectively. Moreover, we assume that the probability of service is the same for both type-1 and type-2 packets unless it is explicitly stated.

We also apply the model to IEEE 802.11 DCF (Similar to the approach taken in Section 2.2, except for the method of computing the probability of success as shown above).

# 3.6 Numerical Analysis and Simulation Results

## 3.6.1 System Parameters and Simulation Setup

Simulations are performed using MATLAB and in the same way as Chapter 2. For numerical analysis, we consider simpler versions of MAP and PH distributions as we did in Section 2.6. We consider a geometric distribution as a special case of phase-type distribution with one phase: $\beta_p = \beta_c = 1$, $S_p = \mu_p$, $S_c = \mu_c$, $S_p^\circ = \mu_p' = 1 - \mu_p$ and $S_c^\circ = \mu_c' = 1 - \mu_c$ for service time distribution of both types of packets. The vacation period is also modeled as a PH distribution having one phase with $V = \gamma'$ and $V^\circ = \gamma$.

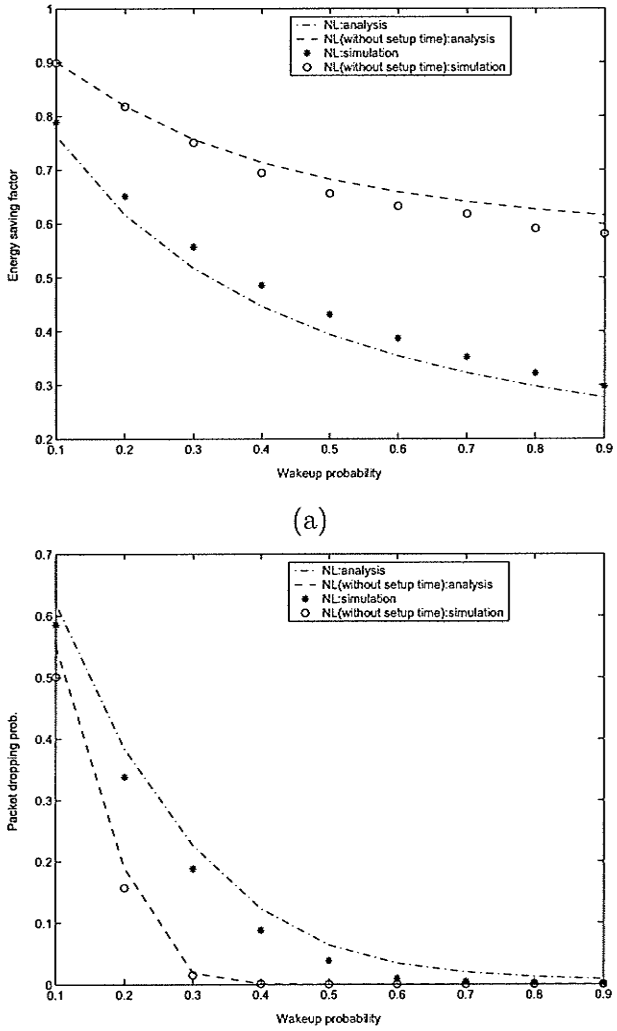
For numerical analysis and simulations, we study the standard exhaustive vacation case as a special case of the number-limited exhaustive vacation case with $M = \infty$. For the number-limited case we assume $M = 1$. We use the parameter $N$ to denote node density (the number of neighboring nodes around the 'tagged' node is $N - 1$).

For predictive p-persistent CSMA protocol, we setup the simulation for different values of node density $N$ and compute the channel access probability for each node at the beginning of each contention period. We assume $p_i = 0.2$ for a high node density scenario ($N \geq 20$) and $p_i = 0.8$ for a low node density scenario ($N \leq 5$). We assume that $p_i$ decreases linearly when the number of nodes increases from 5 to 20.

## 3.6.2 Numerical Results

### 3.6.2.1 Effect of Wakeup Probability

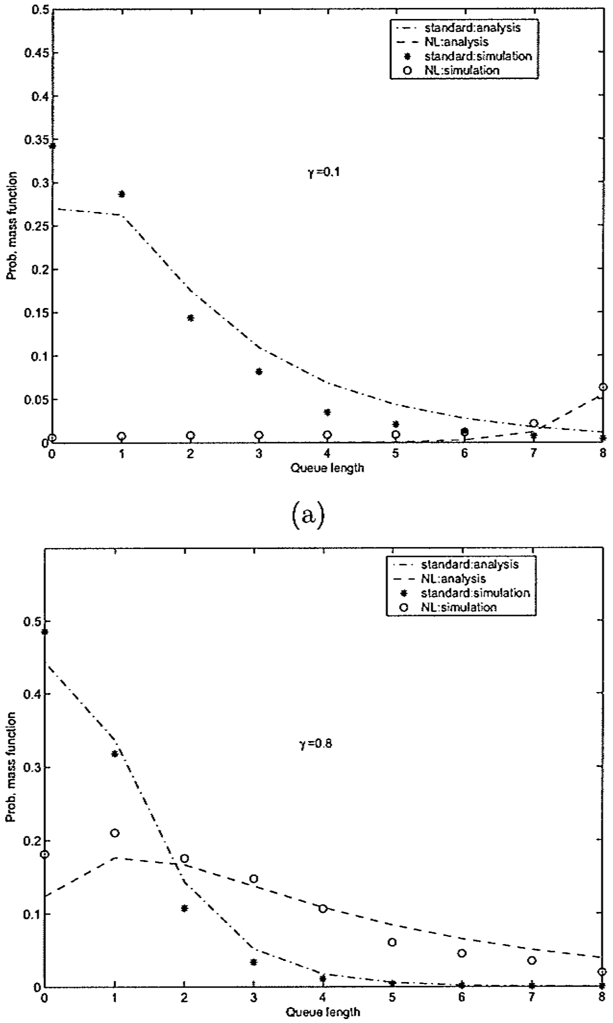
Figs. 3.2-3.3 show the effects of wakeup probability on energy saving factor, packet dropping probability and queue length distribution. Here, the probability of successful transmission is 0.5, the probability of service completion and the probability of arrival for both types of arrivals are $\mu_p = 0.5$, $\mu_c = 0.9$ and $\alpha = 0.2$, respectively. The simulation results are plotted against the numerical results obtained from the analysis. It is evident that the simulation results follow the numerical results very closely which validates the accuracy of the analytical model. The maximum difference between simulation and analysis results in Figs. 3.2(a)-3.3(b) is 0.04, 0.05, 0.07 and 0.04, respectively. Moreover, Fig. 3.5 shows access delay and queue length distributions for



(a)

(b)

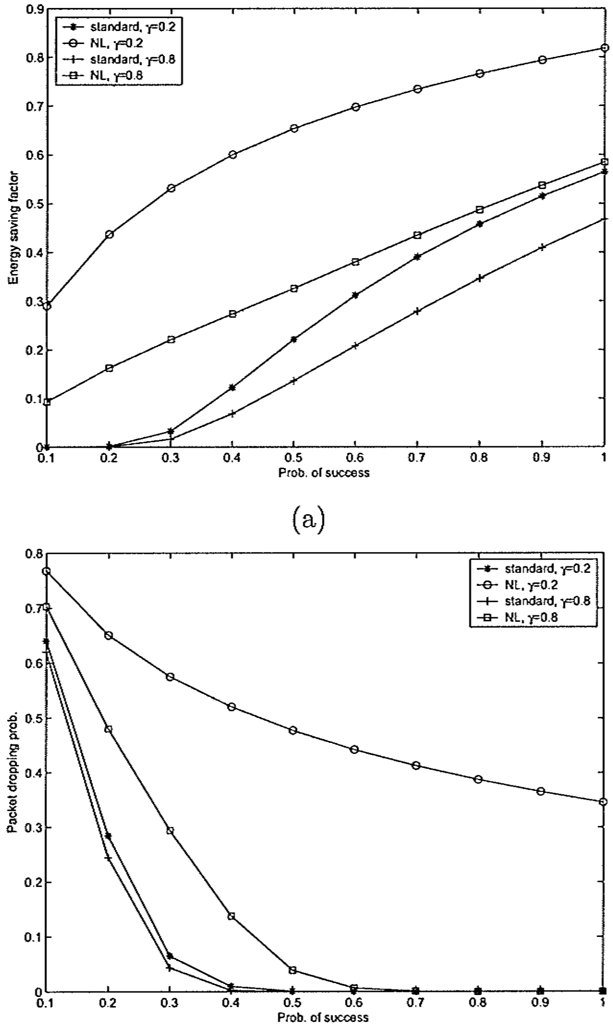
**Figure 3.2.** *Effect of wakeup probability on energy saving factor and packet dropping (for probability of successful transmission p =0.5, probability of arrival $\alpha$ =0.2, probability of service $\mu_p = 0.5, \mu_c = 0.9$).*



(a)

(b)

**Figure 3.3.** *Effect of wakeup probability* $\gamma$ *on queue length distribution* $(p = 0.5, \alpha = 0.2, \mu_p = 0.5, \mu_c = 0.9)$.



(a)

(b)

**Figure 3.4.** *Effect of probability of success on energy saving factor and packet dropping probability for fixed wakeup probability $\gamma$ and for $\alpha = 0.25, \mu_c = 0.9$.*

the 802.11 DCF MAC protocol in the standard exhaustive case when $\gamma = 0.8$.

For comparison, the results on energy saving performance and packet dropping probability for the ideal channel case (i.e., no transmission error) and zero setup time are also shown in Fig. 3.2. The degradation in the energy saving performance due to the non-zero setup time (e.g., due to the contention and back-off) and non-ideal channel conditions (when compared to the more ideal system characteristics) is evident from this figure. Figs. 3.2-3.3 basically reveal the tradeoff between the energy saving and the QoS performance. Higher wakeup probability leads to QoS improvement; however, at the expense of more energy consumption and vice versa.

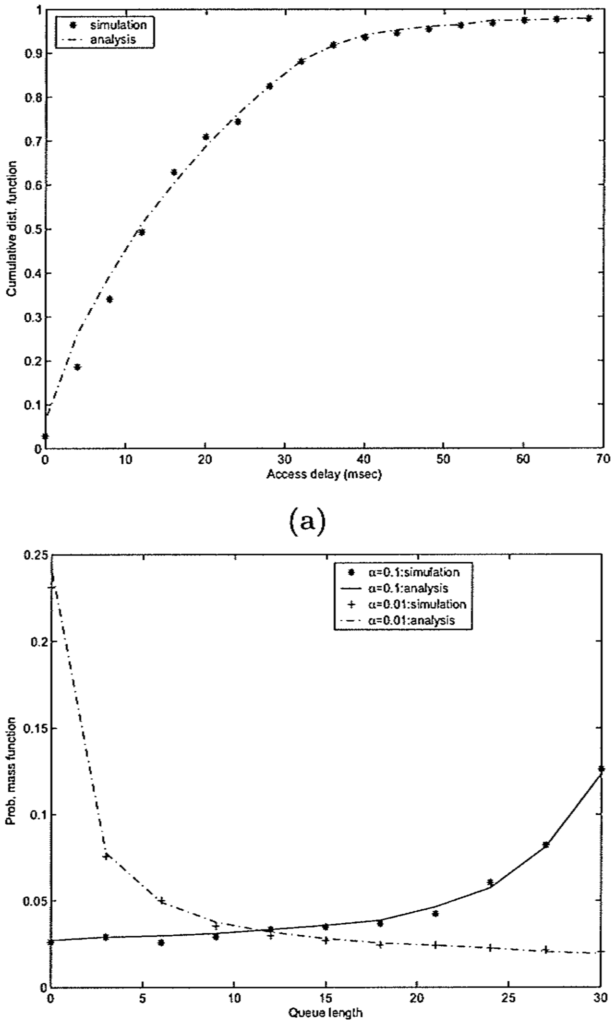
#### 3.6.2.2 Effect of Transmission Error

We study the effect of probability of successful packet transmission (i.e., channel effect) on the performance of the system for different values of wakeup probability. Fig. 3.4 shows the effect of probability of successful transmission on the QoS performances and the energy saving factor. Here, the probability of arrival (traffic load) is $\alpha = 0.25$ for both arrival streams and we study the system for two different wake up probabilities. The channel conditions affect the setup time. Higher probability of channel error tends to destroy the control packets during the setup phase, and therefore, tends to increase the setup time. Hence, increasing the value of $p$, decreases the value of $\mu_p$. To capture this fact, we set $\mu_p = \mu_c - a \times p$ (with $\mu_c = 0.9$ and $a = 0.5$) to obtain the numerical results.

Fig. 3.6 shows queue length distribution and access delay when $p = 0.8$. Among all the cases shown in Fig. 3.6(a), the number-limited case with $\gamma = 0.2$ tends to result in the worst packet dropping probability. In this case, the access delay performance may be unacceptable for delay-sensitive applications (Fig. 3.6(b)).

#### 3.6.2.3 Effect of Probability of Arrival (Traffic Load)

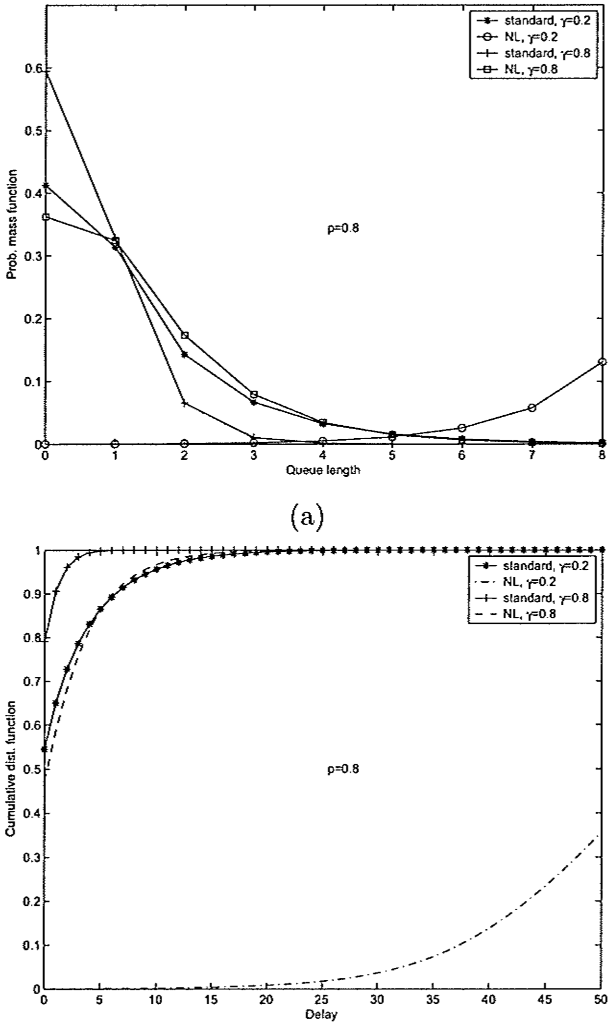
Figs. 3.7-3.8 demonstrate the effect of packet arrival probability (i.e., traffic load) on the system performance. For both these figures, the value of the wakeup probability is set to be the same as the successful packet transmission probability (i.e., channel-dependent wakeup). We study two different traffic loads, $\alpha = 0.1$ and $\alpha = 0.4$, which are the same for both of streams. Moreover, the setup time is assigned according to



(a)

(b)

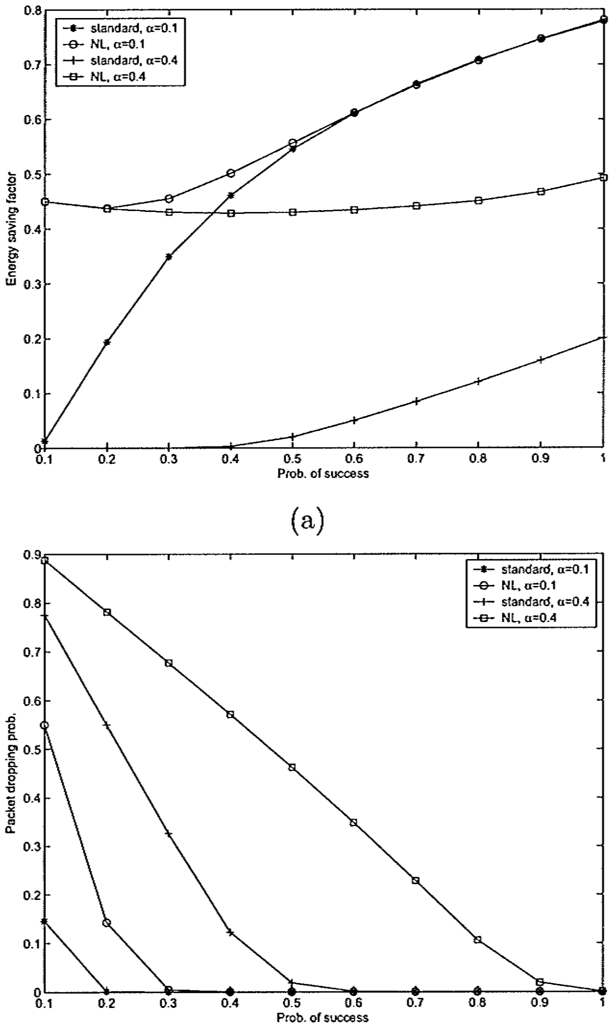
**Figure 3.5.** *Access delay and queue length distribution for 802.11 DCF MAC (for wakeup probability* $\gamma$ *=0.8 and different values of arrival probability* $\alpha$*).*



(a)

(b)

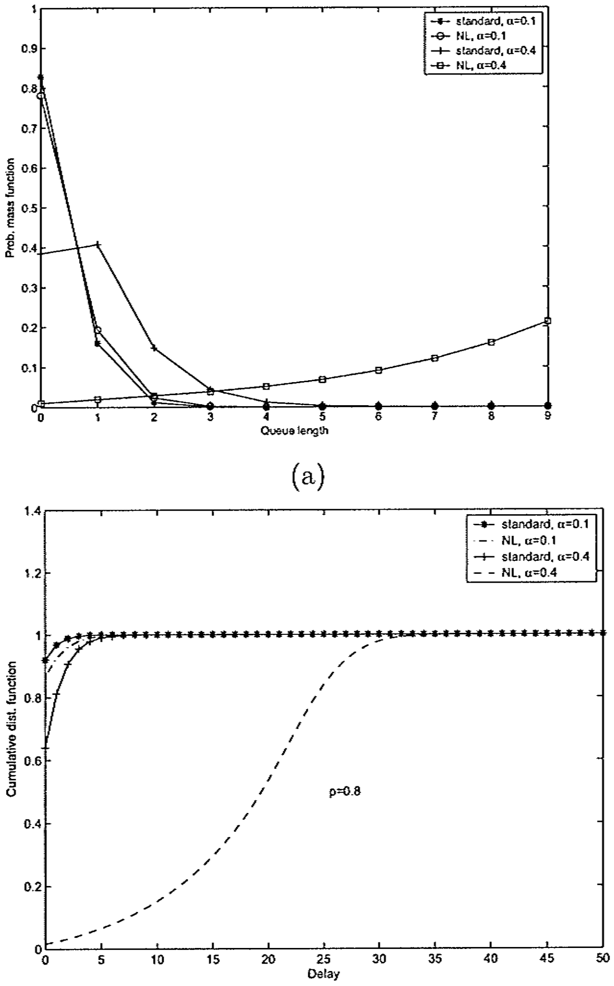
**Figure 3.6.** *Effect of probability of success on queue length and delay for fixed wakeup probability* $\gamma$ *and for* $p = 0.8, \mu_c = 0.9, \mu_p = 0.8$.



(a)

(b)

**Figure 3.7.** *Effect of probability of success on energy saving factor and packet dropping for channel-dependent wakeup ($\alpha$ is the probability of arrival).*



(a)

(b)

**Figure 3.8.** *Effect of probability of success on queue length and delay channel-dependent wakeup ($p = 0.8, \mu_c = \mu_p = 0.8$, $\alpha$ is the probability of arrival).*

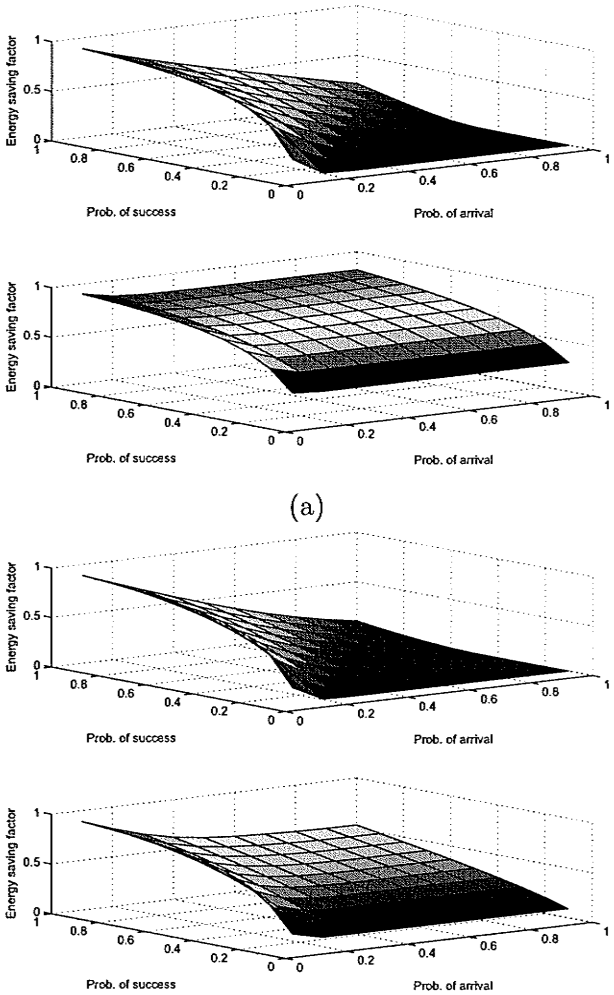
the channel condition (i.e., the probability of successful packet transmission).

Under low traffic load condition (e.g., $\alpha_1 = \alpha_2 = 0.1$) and with higher probability of success, both the standard and the number-limited strategies achieve similar gain in energy saving factor. However, the performance of the standard exhaustive strategy degrades when the probability of transmission error increases (Fig. 3.7(a)). For high traffic load condition (e.g., $\alpha_1 = \alpha_2 = 0.4$), the difference in the energy saving performance with these two strategies is quite significant. For the number-limited case, we observe that, with relatively high traffic intensity, the energy saving factor is almost independent of the channel condition (Fig. 3.7(a)). While the energy saving performance is better in the number-limited case compared to that in the standard case, degradation in the QoS performance is more significant in the former case (Figs. 3.7-3.8). Using the analytical model, the trade-off between the QoS and energy saving can be analyzed quantitatively under different channel conditions.

In Fig. 3.9, the energy saving factor is plotted as a function of probability of arrival and probability of success. Here, the probability of service is $\mu_c = 0.8$ for both types of arrivals. As expected, the energy saving performance improves with increasing probability of successful transmission and/or decreasing probability of arrival. Given a particular required energy saving performance and a given channel condition, the packet arrival rate at the node can be controlled in order to achieve the desired performance objectives.

#### 3.6.2.4 Effect of Number of Neighboring Nodes

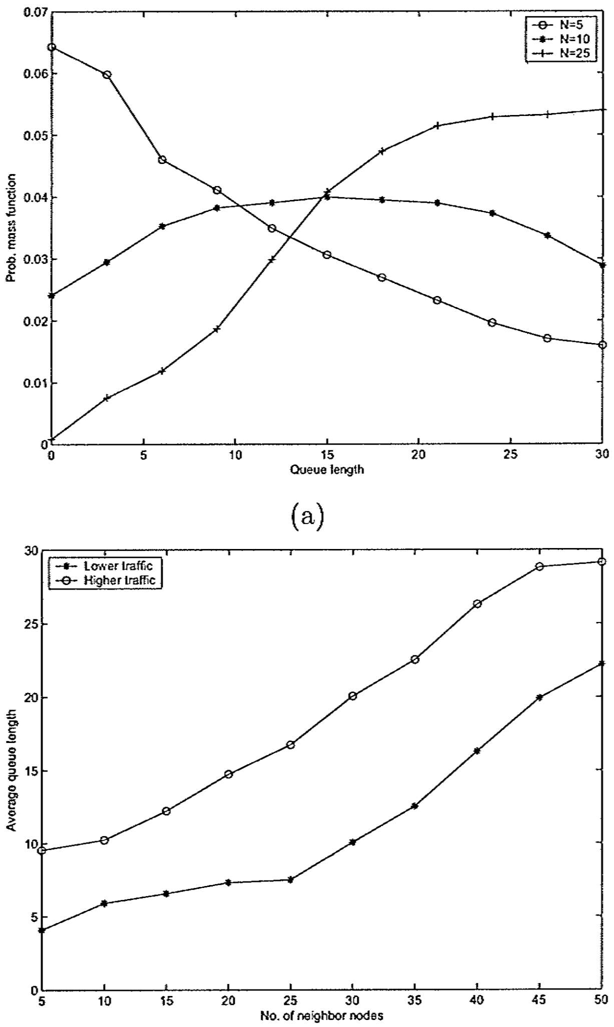
For the predictive p-persistent CSMA protocol, we investigate the effect of the number of neighboring nodes on queue length distribution and average queue length when the size of each of the queues at the tagged node is 30. While the probability mass function for the high and low node density scenarios varies monotonically (which is intuitive), it follows a 'convex' variation in a medium node density scenario (Fig. 3.10(a), for $\alpha_1 = \alpha_2 = \alpha = 0.15$). This is due to the adaptive variation in $p_{x,i}$ in the predictive p-persistent CSMA protocol. Note that, given a particular node density, the probability mass function can be utilized for traffic shaping (to adjust the traffic arrival rate) when the queue length (and hence the access delay) is required to be statistically bounded.



(a)

(b)

**Figure 3.9.** *Effect of probability of arrival and probability of success on energy saving: (a) for wakeup probability, $\gamma = 0.2$ and (b) for wakeup probability $\gamma = 0.8$.*



(a)

(b)

**Figure 3.10.** *Effect of number of neighbor nodes $N$ on (a) queue length distribution and (b) average queue length ($p_i$ = 0.9,$\gamma$ = 0.8).*

Fig. 3.10(b) shows the variation of the average queue length (which is as an index of available resources) when the node density varies. In this case we consider two different arrival rates ($\alpha_a = \alpha_2 = \alpha = 0.1$ and $\alpha_1 = \alpha_2 = \alpha = 0.4$) which correspond to low and high traffic cases, respectively. As expected, the average queue-length increases with increasing traffic arrival rate.

### 3.6.3 Optimal Parameter Setting

The proposed analytical framework can be used to set the system parameters optimally. Here we are interested in finding the optimum wakeup probability and optimum traffic load simultaneously under constrained packet dropping probability performance. We can formulate an optimization problem for optimum wakeup probability as follows:

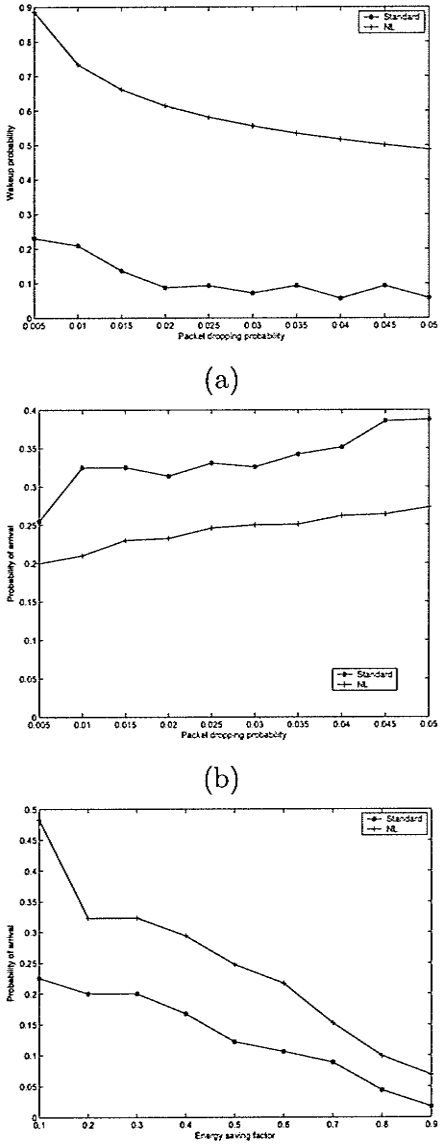
$$\begin{aligned} &\text{minimize} \quad \gamma \\ \text{s.t.} \quad & p_d(\gamma) \leq p_d^{tar}. \end{aligned} \tag{3.62}$$

Note that, similar formulations can be developed under constraints on energy or other QoS factors.

We use Hooke-Jeeves direct search method [72] to solve this optimization problem. This heuristic direct search technique is used to find the minimum of the cost function using the smallest number of iterations. This is done by using a succession of simple moves known as *exploratory searches* and *pattern moves* [73].

Fig. 3.11 shows typical variations in the optimum value of $\gamma$ and $\alpha$ under constraints on packet dropping probability and energy saving factor, respectively. In these cases, we set $p = 0.5$, $\mu_c = 0.8$ and $\mu_p = 0.6$ for both types of arrivals. We observe that (in Figs. 3.11(a)-3.11(b)), for lower packet dropping probability, probability of wakeup needs to be increased. Meanwhile, lower traffic arrival rate is required to meet the target packet dropping probability requirement. As is evident from Fig. 3.11(c), the traffic load needs to be reduced when we need more energy saving in the node.

Results for both the standard and the number-limited cases are shown in Fig. 3.11 from where the relative performances of these two strategies become evident. The frequency of state switching (between the active and the sleep mode) is higher for the number-limited case since in this case the server is not allowed to serve more



(a)

(b)

(c)

**Figure 3.11.** *(a) Optimum wakeup probability for target packet dropping probability, (b) optimum arrival probability for target packet dropping probability, and (c) optimum arrival probability for target energy saving factor (for probability of success p =0.5, and probability of service* $\mu_p = 0.6, \mu_c = 0.8$*).*

than a packet during each cycle. However, the traffic load that can be served under a constrained packet loss probability is lower compared to that for the standard strategy. On the other hand, regarding the energy saving performance, the optimum traffic load in the number-limited case is higher than that in the standard case (Fig. 3.11(c)) under different values for QoS performance such as the packet dropping rate.

## 3.7 Chapter Summary

A novel queueing analytical framework has been presented for performance evaluation of distributed and energy-aware wireless MAC protocols under non-saturation condition and with differentiated services between two types of services. The presented analytical framework is very general and comprehensive in that it considers Markovian arrival process, phase-type distribution for service time and phase-type distribution for vacation period with two types of service disciplines, namely, the standard exhaustive and the number-limited exhaustive both in multiple vacation. In this general analytical framework, the MAC layer access delay (e.g., time required for channel sensing and contention resolution) is taken into account through setup time, the distribution of which is phase-type. Impact of non-ideal wireless channel conditions has been also taken into account.

The trade-off between the QoS performances for the high-priority packets (i.e., queue length distribution, distribution of access delay, packet dropping probability and throughput) and the energy saving performance at a node has been analyzed under different system parameter settings. Although the analyses presented in this paper are for the high-priority packets, similar analyses can also be performed for the low-priority packets.

# Chapter 4

# A Dynamic Programming Approach for QoS-Aware Power Management in Wireless Video Sensor Networks

## 4.1 Introduction

In this chapter, we present a dynamic power management framework for transporting MPEG-coded video traffic over a distributed MAC protocol in a wireless video sensor network. The distributed dynamic power management problem is formulated as a Markov decision process (MDP). The MDP formulation is developed based on a *vacation queue* model [53] with MAP, phase-type service distribution in the transmitter and phase-type sleep mechanism. The traffic source model here in the sensor node captures both short-range correlation and long-range dependence in MPEG video stream. The performance of the proposed framework is analyzed in terms of different QoS metrics considering a FSMC model [54] and multi-rate transmission feature along with automatic repeat request (ARQ)-based radio link level error recovery for wireless transmission. Multi-rate transmission is assumed to be achieved through AMC to adjust the transmission rate according to the channel condition.

## 4.2 Background and Motivation

While there exist a significant amount of research results on issues related to data gathering, routing, and medium access control [8] in wireless sensor networks, QoS provisioning for multimedia communication in these networks has remained vastly unexplored. Many of the existing solutions proposed for multimedia communications in wireless and Internet environments cannot be directly applied to the sensor networks due to the unique characteristics and resource constraints in sensor networks [43]. High and variable error rates in wireless channels and the intrinsic limitations in sensors due to limited energy and simple hardware are significant obstacles for providing QoS support for multimedia applications in sensor networks. In particular, multimedia applications impose significant resource requirements on bandwidth and energy-constrained wireless sensor networks. The problem of dynamic resource management in a sensor node in such a network in a distributed manner is basically a sequential decision making (and optimization) problem.

MDP is a cornerstone in the study of sequential optimization problems that arise in a wide range of fields, where the results of actions taken may be uncertain. An MDP is characterized by mappings for a set of states, actions, Markovian transition probabilities, and real-valued rewards or costs within the process. An optimal planning solution seeks to maximize the sum of rewards (or minimize the costs) over states under some decision policy for state-action pairs given the updated transition probabilities. Dynamic programming is a classical solution method for MDPs. However, in practice, the applicability of dynamic programming may be prohibited due to the size of the underlying state space for a real-world problem. One approach to deal with this difficulty is to generate an approximation to the dynamic programming [74]. In [75], a policy evaluation algorithm was proposed for optimal control of Markov decision processes with quasi birth-death structure. A simulation-based stochastic approximation algorithm was presented in [76] for computing the locally optimal policy of a constrained average cost finite state Markov decision process. The stochastic approximation algorithm requires computation of the gradient of the cost function with respect to the parameter that characterizes the randomized policy.

Dynamic programming [77] is a computational approach to find an optimal policy and has been widely used in the area of wireless networks. A dynamic power control method for a queueing system in a fading channel subject to energy constraint was proposed in [78] to find the optimal transmission rate for each user. Here a central controller chooses a state-dependent transmission rate for each user in a fading downlink channel by varying the transmission power over time. Instead of using a standard dynamic programming approach the authors proposed an approach based on *constrained Markov decision process* [79] to solve the optimization problem. In [80], adaptive control policies, which use information on both queue state and channel state, were explored. The authors used a dynamic programming technique with a discrete-time Markov chain model to study the dynamic power control problem in the context of time-varying wireless channels. A dynamic programming optimization method was used in [81] to obtain the optimal scheduling policy which explores the channel dynamics to obtain a reasonable trade-off between the communication throughput and the packet transmission delay. A general dynamic programming framework was presented in [82] to obtain the optimal power and rate control policies which satisfy deadline-based QoS constraints.

Dynamic programming approach was used in [83] to find a smart policy for energy efficient tracking in wireless sensor networks. Based on the dynamic programming approach, an algorithm was proposed in [84] to analyze the delay-energy tradeoff for a data collecting sensor network, where the energy consumption includes both the transmission energy and the energy consumed in the electronic circuitry. However, the above works did not consider the queueing dynamics at a sensor node which arises due to the bursty data arrival process, the sleep and wakeup dynamics, and the multi-rate packet transmission over a fading wireless channel.

## 4.3 System Model

### 4.3.1 Dynamic Power Management

Dynamic power management (DPM) [85] is a general design methodology aiming at controlling performance and power levels of a system dynamically by exploiting the idleness of different parts of the system and performing selective shutdown of idle system resources. The ultimate goal of DPM is to achieve the maximum performance while minimizing the energy consumption. However, this objective is achievable only by an ideal power management scheme with complete knowledge of present, past, and future workloads. Due to the uncertainty in the communication/computation load in a wireless sensor node, a stochastic approach can be used to describe the system and to find the optimum policy to be used by the power manager.

The block diagram of a wireless video sensor node with a dynamic power management model is shown in Fig. 4.1. Three major parts of the system are camera, data queue, and transmitter, among which, the camera and the wireless transmitter are responsible for major part of power consumption in the sensor node. The power manager monitors the system and observes the states of all the system components and controls the power state of each component through its command for sleep or wakeup. Obviously, the sleep and wakeup dynamics at a sensor node affects the performance of video transmission by increasing the delay when the transmitter is shut down and by increasing the probability of missing some events when the camera is turned off. Based on a *policy*, the power manager issues a command to control the system by turning on/off the controllable parts that include the camera and the wireless transmitter in the sensor node in Fig. 4.1.

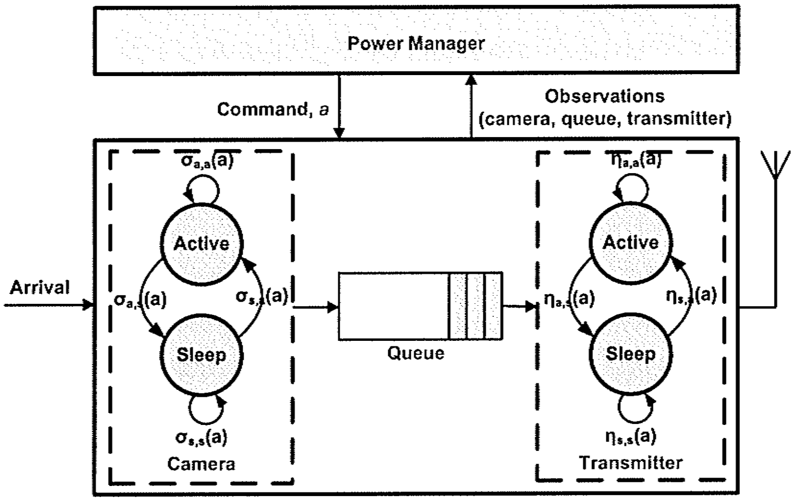


**Figure 4.1.** *Dynamic power management in a wireless video sensor node.*

### 4.3.2 Video Source Model

We assume an MPEG-coded video source model in which three different types of coding schemes are used for a video frame in order to improve the coding efficiency [86]. An I-frame is coded independently of other frames using transform coding. A P-frame is predictively coded based on a previously coded frame and a B-frame is predictively coded using both the previous frame and the successive frame.

Frames are organized in small sets called group of pictures (GoP) each of which starts with an I-frame and consists of B- and P-frames before the next I-frame. Each GoP is characterized by two parameters, $G$ and $H$ in MPEG video streams where GoP patterns are considered to be regular and repetitive. $G$ represents the distance (in number of frames) between two consecutive I-frames. On the other hand, $H$ represents the distance between an I-frame and the first P-frame inside a GoP.

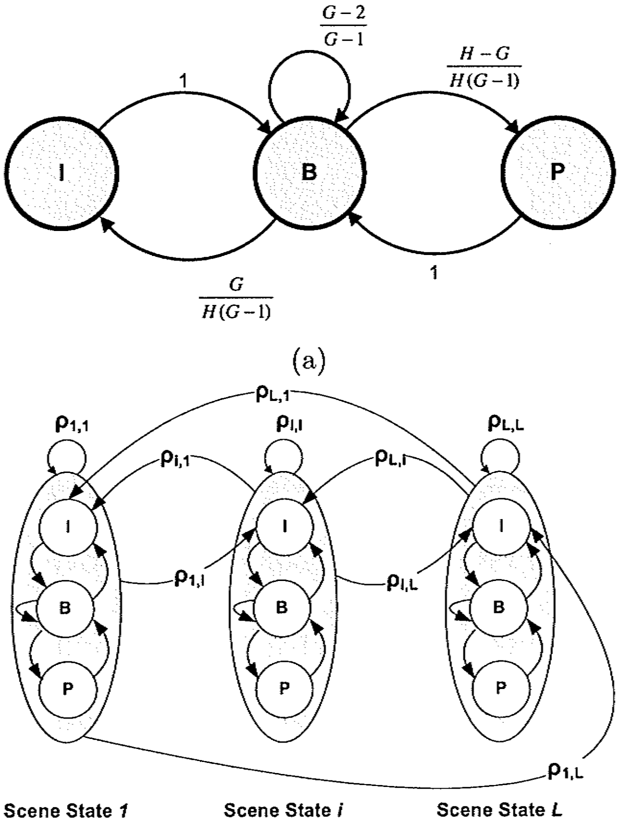
A significant change in the size of two consecutive GoPs indicates a scene change. To model the scene changes, we use a discrete Markov chain where each state or each scene of the MPEG video sequence is represented by the average frame size of the scene. Many video models make use of the concept of scene and introduce several criteria for partitioning a video trace in scenes [87]. The transition of frame types as well as scene changes in video source can be described by a stochastic model as in Fig. 4.2 [86]. The value $\rho_{i,j}$ represents the probability that the next frame will belong to scene state $j$, when the current frame belongs to scene state $i$. Since a new scene always starts with an I-frame, an I-frame is selected with probability 1 whenever a scene state transition occurs. In this model, $\rho_{i,j}$ is obtained as follows:

$$\rho_{i,j} \;=\; \begin{cases} \frac{N_{sc_{ij}}}{N_{f_i}}, & j \neq i \\ 1 - \sum_{k \neq i} \rho_{i,k}, & j = i \end{cases} \tag{4.1}$$

where $N_{sc_{ij}}$ is the number of scene changes from scene state $i$ to scene state $j$ and $N_{f_i}$ represents the number of frames in scene state $i$.

### 4.3.3 Wireless Channel Model and Multi-rate Transmission

We consider a slowly varying Nakagami-$m$ fading channel represented by a FSMC model, which is a useful model for analyzing radio channel with non-independent fad-



(a)

(b)

**Figure 4.2.** *(a) Frame transition and (b) scene transition.*

ing (and hence bursty channel errors) [54]. Also, multi-rate transmission is considered which is assumed to be achieved through AMC. Each state of the FSMC corresponds to one transmission mode for AMC. With a $K$ state FSMC, the signal-to-noise ratio (SNR) at the receiver ($\gamma$) can be partitioned into $K+1$ non-overlapping intervals by thresholds $\Gamma_k$ ($k \in \{0, 1, \ldots, K\}$), where $\Gamma_0 = 0 < \Gamma_1 < \ldots < \Gamma_{K+1} = \infty$. The channel is said to be in state $k$ if $\Gamma_k \leq \gamma < \Gamma_{k+1}$. In this state, $f_k$ bits can be transmitted per symbol using $2^k$-QAM (quadrature amplitude modulation) which corresponds to transmission rate $F_k$ (in packets per time slot). To avoid possible transmission error, no packet is transmitted when channel state $k = 0$. The method to calculate the average packet error rate $\overline{PER}_k$ corresponding to transmission rate $k$ can be found in [54]. The channel transition matrix $\mathbf{T}$ is given as follows.

$$\mathbf{T} = \begin{bmatrix} t_{0,0} & t_{0,1} & \cdots & 0 \\ t_{1,0} & t_{1,1} & t_{1,2} & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & t_{K-1,K-2} & t_{K-1,K-1} & t_{K-1,K} \\ 0 & \cdots & t_{K,K-1} & t_{K,K} \end{bmatrix} \tag{4.2}$$

where $\mathbf{T} \in \mathbb{R}^{K+1 \times K+1}$ denotes the channel state transition matrix and $t_{k,k'}$ denotes the transition probability from state $k$ to $k'$ ($k' \in \{k-1, k, k+1\}$).

Assuming that a transmission error recovery protocol (e.g., based on a reliable automatic repeat request (ARQ) mechanism) is used, for an independent packet error process, the probability that $i$ out of $j$ packets (i.e., $i \in \{0, 1, \ldots, j\}$) are successfully transmitted (i.e., depart the transmission queue) in one time slot can be obtained as follows:

$$\xi_{i,j}^{(k)} = \begin{pmatrix} j \\ i \end{pmatrix} \theta_k^{j-i}(1-\theta_k)^i \tag{4.3}$$

where $\theta_k$ is the probability of successful transmission of a packet (i.e., $\theta_k = 1 - \overline{PER}_k$) when the transmitter uses modulation level $k$. We also assume that the transmission status for the packet(s) transmitted in the previous transmission interval[1] becomes available to the sensor node before transmissions in the current transmission interval start.

## 4.4 Queueing Dynamics at a Sensor Node Under Dynamic Power Management

We model the queueing dynamics at a sensor node by using the theory of discrete-time Markov chains, where time is divided into fixed length intervals called slots. At

[1] One transmission interval corresponds to the time required to perform the necessary handshaking and transmit packet(s), and the time spent in waiting to receive the acknowledgement(s) from the receiver. For simplicity, from now on, we will refer to this transmission interval as a time slot.

least one time slot is required to serve one packet. After capturing scenes, MPEG-coded video frames are generated in the camera. The distribution of the size of frames is given by a phase-type distribution $(\vartheta_x, Y_x)$ for frame type $x$ (where $x \in \{I, B,\ P\}$). The transition matrix $Y_x$ is a substochastic matrix and it corresponds to the transitions within the transient states. Here $\vartheta_x$ is a vector and its element $\vartheta_{x_i}$ denotes the probability that the frame generation process starts from state $i$. These frames are stored in the buffer of the sensor node for transmission. The service process in the wireless transmitter is determined by a distributed MAC protocol and the physical layer model (i.e., the FSMC model described before). The matrix $D$ below captures scene-level, frame-level, and packet-level evolutions for the encoded MPEG traffic.

$$D = \begin{bmatrix} \rho_{1,1}\Lambda_1 & \rho_{1,2}\Psi & \ldots & \rho_{1,L-1}\Psi & \rho_{1,L}\Psi \\ \rho_{2,1}\Psi & \rho_{2,2}\Lambda_2 & \ldots & \rho_{2,L-1}\Psi & \rho_{2,L}\Psi \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ \rho_{L-1,1}\Psi & \rho_{L-1,2}\Psi & \ldots & \rho_{L-1,L-1}\Lambda_{L-1} & \rho_{L-1,L}\Psi \\ \rho_{L,1}\Psi & \rho_{L,2}\Psi & \ldots & \rho_{L,L-1}\Psi & \rho_{L,L}\Lambda_L \end{bmatrix} \qquad (4.4)$$

where

$$\Lambda_i = \begin{bmatrix} Y_I & Y_I^0\vartheta_B & 0 \\ \frac{G}{H(G-1)}Y_B^0\vartheta_I & Y_B + \frac{G-2}{G-1}Y_B^0\vartheta_B & \frac{H-G}{H(G-1)}Y_B^0\vartheta_P \\ 0 & Y_P^0\vartheta_B & Y_P \end{bmatrix},\ \Psi = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \qquad (4.5)$$

The states of the video sensor node can be described by a Markov chain the state space for which, in general, can be described as $\mathcal{X} = \mathcal{N} \times \mathcal{R} \times \mathcal{S} \times \mathcal{T} \times \mathcal{V}_1 \times \mathcal{V}_2$, where $\mathcal{N}$ is the number of packets in the queue, $\mathcal{R}$ is the arrival process, $\mathcal{T}$ is the channel state, $\mathcal{S}$ is the service process in transmitter, and $\mathcal{V}_1$ and $\mathcal{V}_2$ are sleep/wakeup processes in camera and transmitter, respectively. Transition matrix $P$ in (4.6) denotes the transition matrix for this Markov chain where $N$ is the queue size. In general, matrix $P$ is a level-dependent matrix.

$$P = \left[ \begin{array}{cccccccc} A_{0,0} & C_0 & & & & & & \\ A_{1,1} & A_{1,0} & C_1 & & & & & \\ A_{2,2} & A_{2,1} & A_{2,0} & C_2 & & & & \\ \vdots & \vdots & \vdots & \vdots & & & & \\ A_{K-1,K-1} & A_{K-1,K-2} & A_{K-1,K-3} & \cdots & C_{K-1} & & & \\ A_{K,K} & A_{K,K-1} & A_{K,K-2} & \cdots & A_{K,0} & C_K & & \\ \hline & A_K^{(K+1)} & A_{K-1}^{(K+1)} & \cdots & A_1^{(K+1)} & A_0^{(K+1)} & E_0^{(K+1)} & \\ & & & & & & & \\ & & A_K^{(K+i)} & A_{K-1}^{(K+i)} & \cdots & A_1^{(K+i)} & A_0^{(K+i)} & E_0^{(K+i)} \\ & & & & & & & \\ & & & A_K^{(N)} & A_{K-1}^{(N)} & \cdots & A_1^{(N)} & A_0^{(N)} + E_0^{(N)} \end{array} \right] \quad (4.6)$$

The block matrices in (4.6) are explained below where $T_i (i = 0, 1, \ldots, K)$ is obtained by keeping only $i + 1^{st}$ row of the channel transition matrix $T$ and setting all other rows to $\mathbf{0}$. The sleep and wakeup processes in the camera are phase-type processes and are represented by $(u, U)$ and $(\omega, W)$ respectively. $(\beta, S)$ and $(\nu, V)$ represent, respectively, the service process and the sleep mechanism in the wireless transmitter.

We also model the activity of the camera including the sleep/capture dynamics and encoding by a discrete Markovian arrival process (D-MAP) [56] for the queue. The MAP in our model is described by two substochastic matrices $D_0^*$ and $D_1^*$ below

$$D_0^* = \begin{bmatrix} U & U^0 w \\ 0 & 0 \end{bmatrix} \otimes I_D, \qquad D_1^* = \begin{bmatrix} 0 & 0 \\ W^0 u & W \end{bmatrix} \otimes D$$

which represent no arrival and arrival to the queue, respectively, where $I_D$ is an identity matrix with dimension $D$.

In (4.6), $A_{0,0}$ is given as follows and represents no arrival in the queue while the queue is empty and the channel evolution is considered.

$$A_{0,0} = D_0^* \otimes T \otimes \begin{bmatrix} V + V^0 \nu & 0 \\ 1 & 0 \end{bmatrix}$$

$C_0$ considers that the MAC protocol starts serving the packet recently arrived in the empty queue.

$$C_0 = D_1^* \otimes T \otimes \begin{bmatrix} V & V^0\beta \\ 0 & 1 \end{bmatrix}$$

$A_{k,k}$ denotes the case that all packets in the queue are served successfully and no arrival occurs during the time slot (and hence, at the end of the slot, the queue is empty).

$$A_{k,k} = D_0^* \otimes \sum_{i=k}^{K} \xi_{k,k}^{(i)} T_i \otimes \begin{bmatrix} 0 & 0 \\ S^0\nu & 0 \end{bmatrix}$$

In $A_k, k \geq 1$, some packets are served; however, the queue is not empty at the end of the slot. The number of served packets is $k$ in the case of no arrival and $k+1$ if an arrival occurs.

$$A_k = D_0^* \otimes (\sum_{i=k}^{K} \xi_{k,i}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} + D_1^* \otimes (\sum_{i=k+1}^{K} \xi_{k+1,i}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \quad k \geq 1$$

$A_0$ represents the case that either packet transmission is still in progress or it has failed and needs to be retransmitted. It also represents that there is no change in the number of packets in the queue either due to the fact that the channel is completely faded (no transmission is possible) or only one successful transmission and one arrival have occurred during the time slot.

$$\begin{aligned} A_0 &= D_0^* \otimes \left( T \otimes \begin{bmatrix} V & V^0\beta \\ 0 & S \end{bmatrix} + (\sum_{i=1}^{K} \xi_{0,i}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} + T_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \right) \\ &+ D_1^* \otimes (\sum_{i=1}^{K} \xi_{1,i}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \end{aligned}$$

$C_k, k \geq 1$ represents that the number of packets in the queue increases by one when there is an arrival and no successful transmission occurs.

$$C_n = D_1^* \otimes T \otimes \begin{bmatrix} V & V^0\beta \\ 0 & S \end{bmatrix} + D_1^* \otimes (\sum_{i=0}^{n-1} \xi_{0,i}^{(i)} T_i + \sum_{i=n}^{K} \xi_{0,n}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix}$$

The interpretation of $E_0$ for $K+1 \leq j \leq N$ is the same as that of $C_k$.

$$\begin{aligned} E_0 &= D_1^* \otimes T \otimes \begin{bmatrix} V & V^0\beta \\ 0 & S \end{bmatrix} + D_1^* \otimes T_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \\ &+ D_1^* \otimes (\sum_{i=1}^{K} \xi_{0,i}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \end{aligned}$$

$A_{k',k}, k' > k$ represents successful transmission of $k$ packets when there are $k'$ packets in the queue.

$$\begin{aligned} A_{n,k} &= D_0^* \otimes (\sum_{i=k}^{n-1} \xi_{k,i}^{(i)} T_i + \sum_{i=n}^{K} \xi_{k,n}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \\ &+ D_1^* \otimes (\sum_{i=k+1}^{n-1} \xi_{k+1,i}^{(i)} T_i + \sum_{i=n}^{K} \xi_{k+1,n}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \quad k \geq 1,\ n > k \end{aligned}$$

$$\begin{aligned} A_{n,0} &= D_0^* \otimes (\sum_{i=0}^{n-1} \xi_{0,i}^{(i)} T_i + \sum_{i=n}^{K} \xi_{0,n}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} + D_0^* \otimes T \otimes \begin{bmatrix} V & V^0\beta \\ 0 & S \end{bmatrix} \\ &+ D_1^* \otimes (\sum_{i=1}^{n-1} \xi_{1,i}^{(i)} T_i + \sum_{i=n}^{K} \xi_{1,n}^{(i)} T_i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0\beta \end{bmatrix} \quad n \geq 1 \end{aligned}$$

## 4.5 Analysis and Optimization of the Power Management Model

A dynamic power management policy can be designed where the sleep and the wakeup transitions in the Markov chain are controlled by a command $a \in$ {sleep, active} issued by the power manager in the node. This results in a controllable Markov chain which can be described as a Markov decision process [88]. The transition matrix $P$ in (4.6) for this controllable Markov chain is $P(a)$ which is a function of the command $a$ issued by the power manager. Specifically, in the system model shown in Fig. 4.1, based on the command $a$, the probability of transition from state $i$ to state $j$ (i.e., $\sigma_{i,j}(a)$ and $\eta_{i,j}(a)$ in the camera and the wireless transmitter, respectively (where

$i, j \in$ {sleep, active}), can be varied as functions of $a$, based on the system information in the power manager. This will result in a probabilistic sleep and wakeup mechanism for energy saving in the sensor node.

In general, the power manager monitors the state of the sensor node and issues command $a$ with probability $p_a$. Such a decision is made based on the observation of the system including the state of the camera (sleep and capture), state of the transmitter (sleep and active), and the state of the queue in the node up to time $t_n$. The decision at time $t_n$ is $\phi_n$ which associates a probability $p_a$ with each command $a$. The transition matrix of the system for decision $\phi_n$ at time $t_n$ is $P_{\phi_n} = \sum_{p_a \in \phi_n} p_a P(a)$. The infinite sequence $\pi = [\phi_1, \phi_2, \dots)$ describes the policy of the power manager in the node and if policy $\pi$ is adopted, transition matrices under policy $\pi$ and up to time $t_n$ are defined as $P_\pi^n = P_{\phi_1}, \dots, P_{\phi_n}$. In the case that there is no power manager in the system, the Markov chain is not controllable, that is, $P_{\phi_i} = P, \forall i$. This is referred to as the *static power management (SPM)* scheme. We use dynamic programming to obtain a simple procedure for optimization using the special structure of matrix $P$. Then, based on this procedure we solve the general problem of dynamic power management.

### 4.5.1 Optimal Control in Quasi Birth-Death Decision Processes

A Quasi Birth-Death (QBD) process $X(t)$ is a discrete-state Markov process where the transition probability matrix for $X(t)$ has the following block tridiagonal form due to the limitation on allowable transitions:

$$P = \begin{bmatrix} A_1^{(0)} & A_0^{(0)} & & & & \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & & & \\ & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_2^{(S-1)} & A_1^{(S-1)} & A_0^{(S-1)} \\ & & & & A_2^{(S)} & A_1^{(S)} \end{bmatrix}. \tag{4.7}$$

For an MDP, the transition probability matrix $P$ is a function of policy or control function $u$. Therefore, each block $A_i^{(s)}$ is also a function of $u$ and we have

$$Pr\{X(t) = k'|X(t-1) = k, a \in u\} = \begin{cases} A_0^{(k)}(a), & k' = k+1 \\ A_1^{(k)}(a), & k' = k \\ A_2^{(k)}(a), & k' = k-1. \end{cases} \tag{4.8}$$

The cost-to-go function for state $i$ under policy $u$ is defined by

$$J_u(i) = E\left\{\sum_{t=0}^{\infty} \alpha^t g(X(t), X(t+1), a_t)|X(0) = i\right\} \tag{4.9}$$

where $g(.,.,.)$ denotes the cost incurred from a transition from $X(t)$ to $X(t+1)$ under control $a_t = u(X(t))$. Bellman's equation provides a solution for the optimal cost, and the associated policy and optimal cost-to-go are given by the solution of

$$\begin{aligned} J(i) &= \min_{a_t \in \mathcal{A}} E\{g(i, X(1), a_t) + \alpha J(X(1))|X(0) = i\} \\ &= \min_{a_t \in \mathcal{A}} \left\{\sum_{j=0}^{S} [P_a]_{ij}(g(i,j,a_t) + \alpha J(j))\right\}. \end{aligned} \tag{4.10}$$

We can write the above equation in matrix form as follows:

$$J = \min_a \{g(a) + \alpha P_a J\} \tag{4.11}$$

where the entries for vector $g(a)$ are given by

$$[g(a)]_i = \sum_{j=0}^{S} [P_a]_{ij} g(i,j,a_t) \tag{4.12}$$

and for a given policy $u$, the associated cost-to-go satisfies the following equation:

$$J_u = g(a) + \alpha P_a J_u \tag{4.13}$$

where $a$ is specified by policy $u$ for each state. Solving the cost-to-go under a specified policy $u$ is called *policy evaluation.* To determine the optimum cost-to-go, we follow a *policy iteration* method as follows. At the beginning, an initial policy $u_0$ is specified. That is, we apply $a_0 = u_0(i)$ in state $i$. Then the policy is evaluated by solving the above equation. Next, policy $u_1$ is obtained in *policy improvement* step as follows:

$$u_1 = argmin_u\{g(a) + \alpha P_a J_{u_0}\}. \tag{4.14}$$

In the above equation, $a$ is generated according to the policy $u$ and the minimum is taken over all applicable policies. Because of the finite number of policies, if we repeat the process, the policy iteration leads us to the solution within a limited number of iterations.

For the specific transition matrix $P$ in (4.6), we can use the above procedure by rewriting (4.8) as follows:

$$\begin{aligned} Pr\{X(t) = k' \quad | \quad & X(t-1) = k, a \in u\} \\ = & \begin{cases} C_k(a), & k' = k+1 \\ A_{k,0}(a), & k' = k \qquad\qquad k \leq K \\ A_{k,k-k'}(a), & k' = 1, ..., k-1 \end{cases} & (4.15) \\ = & \begin{cases} E_0^{(k)}(a) & k' = k+1 \\ A_0^{(k)}(a) & k' = k \qquad\qquad K < n < N \\ A_{k'}^{(k)}(a) & k' = 1, ..., K. \end{cases} & (4.16) \end{aligned}$$

### 4.5.2 Policy Evaluation for QBD Processes

We can exploit the QBD structure to reduce the computational complexity of the policy evaluation which incurs most of the computational cost in the above mentioned policy iteration process. From now on we assume that a fixed policy is being evaluated and $g_{i,j}$ is the associated cost from state $i$ to state $j$ under that policy.

Cost-to-go for each state is obtained from (4.13) as follows. For state 0, we have:

$$J_0 = \alpha A_{0,0} J_0 + \alpha C_0 J_1 + g_{0,0} + g_{0,1}. \tag{4.17}$$

Letting $g_0 = g_{0,0} + g_{0,1}$ and $\Gamma_0 = A_{0,0}$, we can rewrite $J_0$ as follows:

$$J_0 = (I - \alpha\Gamma_0)^{-1}(\alpha C_0 J_1 + g_0). \tag{4.18}$$

For $J_k, 1 \leq k \leq K$, (4.13) can be expressed as follows:

$$J_k = \alpha A_{k,k} J_0 + \alpha \sum_{i=0}^{k-1} A_{k,i} J_{k-i} + \alpha C_k J_{k+1} + \sum_{i=0}^{k+1} g_{k,i}. \tag{4.19}$$

For $k = 1$, we can substitute $J_0$ from (4.18) to find $J_1$ as follows:

$$J_1 = (I - \alpha\Gamma_1)^{-1}(\alpha C_1 J_2 + g_1) \tag{4.20}$$

where

$$\begin{aligned} \Gamma_1 &= A_{1,1}(I - \alpha\Gamma_0)^{-1}\alpha C_0 + A_{1,0} && (4.21) \\ g_1 &= \alpha A_{1,1}(I - \alpha\Gamma_0)^{-1} g_0 + \sum_{i=0}^{2} g_{1,i}. && (4.22) \end{aligned}$$

Similarly, we can obtain $J_k$ for $k = 2, ..., K$ as follows:

$$\begin{aligned} J_k &= (I - \alpha\Gamma_k)^{-1}(\alpha C_k J_{k+1} + g_k) && (4.23) \\ \Gamma_k &= \sum_{i=0}^{k} A_{k,i} \prod_{j=k-i}^{k-1} (I - \alpha\Gamma_j)^{-1}\alpha C_j && (4.24) \\ g_k &= \sum_{i=0}^{k-1} \left( \sum_{j=k-i}^{k} \alpha A_{k,j}(I - \alpha\Gamma_{k-j})^{-1} \prod_{l=k-j+1}^{i} \alpha C_{l-1}(I - \alpha\Gamma_l)^{-1} \right) g_i \\ &+ \sum_{i=0}^{k+1} g_{k,i}. && (4.25) \end{aligned}$$

Moreover, $J_k, 0 \leq k \leq K$ can be calculated in terms of $J_{K+1}$ as follows:

$$\begin{aligned} J_k &= R_{K+1-k} J_{K+1} + h_{K+1-k} && (4.26) \\ R_{K+1-k} &= \prod_{i=k}^{K} (I - \alpha\Gamma_i)^{-1}\alpha C_i && (4.27) \\ h_{K+1-k} &= \sum_{j=0}^{K-k} (I - \alpha\Gamma_k)^{-1} \left( \prod_{i=k+1}^{k+j} \alpha C_{i-1}(I - \alpha\Gamma_i)^{-1} \right) g_{k+j}. && (4.28) \end{aligned}$$

For $J_{K+1}$, we have

$$\begin{aligned} J_{K+1} &= \sum_{i=0}^{K} \alpha A_{K-i}^{(K+1)} J_{i+1} + \alpha E_0^{(K+1)} J_{K+2} + \sum_{i=1}^{K+2} g_{K+1,i} \\ &= \alpha A_K^{(K+1)}(R_K J_{K+1} + h_K) + \alpha A_{K-1}^{(K+1)}(R_{K-1} J_{K+1} + h_{K-1}) \\ &\quad + \cdots + \alpha A_1^{(K+1)}(R_1 J_{K+1} + h_1) + \alpha A_0^{(K+1)} J_{K+1} \\ &\quad + \alpha E_0^{(K+1)} J_{K+2} + \sum_{i=1}^{K+2} g_{K+1,i}. \end{aligned} \tag{4.29}$$

$$\Rightarrow J_{K+1} = (I - \alpha\Gamma_{K+1})^{-1}\left(\alpha E_0^{(K+1)} J_{K+2} + g_{K+1}\right) \tag{4.30}$$

$$\Gamma_{K+1} = \sum_{j=1}^{K} A_j^{(K+1)} R_j + A_0^{(K+1)} \tag{4.31}$$

$$g_{K+1} = \sum_{j=1}^{K} \alpha A_j^{(K+1)} h_j + \sum_{i=1}^{K+2} g_{K+1,i}. \tag{4.32}$$

In a similar way, $J_{K+i}$ can be obtained as follows:

$$J_{K+i} = (I - \alpha\Gamma_{K+i})^{-1}\left(\alpha E_0^{(K+i)} J_{K+i+1} + g_{K+i}\right) \tag{4.33}$$

where

$$\begin{aligned} \Gamma_{K+i} &= (\sum_{j=i}^{K} A_j^{(K+i)} R_j) \prod_{l=1}^{i-1} (I - \alpha\Gamma_{K+l})^{-1} \alpha E_0^{(K+l)} \\ &\quad + \sum_{j=0}^{i-1} \left( A_j^{(K+i)} \prod_{l=i-j}^{i-1} (I - \alpha\Gamma_{K+l})^{-1} \alpha E_0^{(K+l)} \right) \end{aligned} \tag{4.34}$$

$$g_{K+i} = (\sum_{j=i}^{K} A_j^{(K+i)} R_j) m_{K+i,K+1} + \alpha \sum_{j=0}^{i-1} A_j^{(K+i)} m_{K+i,K+i-j} + \sum_{j=i}^{K+i+1} g_{K+i,j} \tag{4.35}$$

and

$$m_{r,l} = \sum_{j=0}^{r-l} (I - \alpha\Gamma_l)^{-1} \left( \prod_{i=l+1}^{l+j} \alpha E_0^{(i-1)} (I - \alpha\Gamma_i)^{-1} \right) g_{l+j}. \tag{4.36}$$

Finally, from (4.13) we have

$$J_N = \alpha \sum_{i=0}^{K} (A_{K-i}^{(N)} J_{i+N-K}) + \alpha E_0^{(N)} J_N + \sum_{i=N-K}^{N} g_{N,i}. \tag{4.37}$$

For $N < K$,

$$J_N = (\alpha \sum_{j=N-K}^{K} A_j^{(N)} R_j) J_{K+1} + \alpha \sum_{j=0}^{N-K-1} A_j^{(N)} J_{N-j} + \alpha E_0^{(N)} J_N + \sum_{i=N-K}^{N} g_{N,i}. \tag{4.38}$$

$J_{K+1}$ and $J_{N-j}$ can be written in terms of $J_N$ as follows:

$$J_{K+1} = U_{K+1} J_N + m_{N,K+1} \tag{4.39}$$

$$J_{N-j} = U_{N-j} J_N + m_{N,N-j} \tag{4.40}$$

where

$$U_l \quad = \quad \alpha \prod_{i=l}^{N-1} (I - \alpha \Gamma_i)^{-1} E_0^{(i)}. \tag{4.41}$$

Then, $J_N$ is given as follows:

$$J_N = (I - \alpha \Gamma_N)^{-1} g_N \tag{4.42}$$

where

$$\Gamma_N \quad = \quad \left( \sum_{j=N-K}^{K} A_j^{(N)} R_j \right) U_{K+1} + \sum_{j=0}^{N-K-1} A_j^{(N)} U_{N-j} + E_0^{(N)} \tag{4.43}$$

$$g_N \quad = \quad \alpha \left( \sum_{j=N-K}^{K} A_j^{(N)} R_j \right) m_{N,K+1} + \alpha \sum_{j=0}^{N-K-1} A_j^{(N)} m_{N,N-j} + \sum_{i=N-K}^{N} g_{N,i}. \tag{4.44}$$

For $N \geq K$, (4.37) is given as follows:

$$J_N = \alpha \sum_{j=0}^{K} A_j^{(N)} J_{N-j} + \alpha E_0^{(N)} J_N + \sum_{i=N-K}^{N} g_{N,i} \tag{4.45}$$

and therefore, $\Gamma_N$ and $g_N$ in (4.42) can be expressed as follows:

$$\begin{aligned}\Gamma_N &= \sum_{j=0}^{K} A_j^{(N)} U_{N-j} + E_0^{(N)} &\quad (4.46)\\ g_N &= \alpha \sum_{j=0}^{K} A_j^{(N)} m_{N,N-j} + \sum_{i=N-K}^{N} g_{N,i}. &\quad (4.47)\end{aligned}$$

The procedure above is summarized as an algorithm for policy evaluation as follows.

- Initialization:

$$\begin{aligned}\Gamma_0 &= A_{0,0} &\quad (4.48)\\ g_0 &= g_{0,0} + g_{0,1}. &\quad (4.49)\end{aligned}$$

- Evaluate $\Gamma_k$ and $g_k$ in (4.24) and (4.25), respectively, for $1 \leq k \leq K$.
- Evaluate $\Gamma_k$ and $g_k$ in (4.34) and (4.35), respectively, for $K+1 \leq k < N$.
- Evaluate $\Gamma_N$ and $g_N$ in (4.46) and (4.47), respectively.
- Use (4.45) to solve $J_N$.
- Use (4.33) to solve $J_k$, $\quad K+1 \leq k \leq N$.
- Use (4.23) to solve $J_k$, $\quad 1 \leq k \leq K$.

### 4.5.3 Complexity of Policy Evaluation

For queue length $N$, if the dimension of each of the block matrices in matrix $P$ is $M+1$, the computational complexity for a general (i.e. non-QBD) policy evaluation policy is $O(N^3M^3)$ since the complexity of matrix inversion is $O(n^3)$ when matrix dimension is $n$. However, the computational complexity can be reduced by exploiting the QBD structure. In the following, we find the complexity of the QBD approach assuming that the complexity of matrix multiplication is $O(n^2)$. For simplicity, we consider the complexity of matrix inversions and multiplications only and ignore the complexity of matrix summations and scalar operations.

Following (4.23), (4.24) and (4.25) and counting the required computations, complexity for each $k, k = 0, \cdots, K$ is obtained as follows:

$$\mathbb{C}_k = k^2 O(M+1)^3 + \frac{5k^2+k}{2} O(M+1)^2 \tag{4.50}$$

and the total complexity for all $k, k = 0, \cdots, K$ is

$$\mathbb{X}_1 = \sum_{k=0}^{K} \mathbb{C}_k = \frac{K(K+1)(2K+1)}{6} O(M+1)^3 + \frac{K(K+1)(10K+8)}{12} O(M+1)^2. \tag{4.51}$$

The complexity of evaluating (4.33), (4.34) and (4.35) for each $i, i = 1, \cdots, N-K-1$ is given as follows:

$$\mathbb{C}_{K+i} = (K^2+K+i)O(M+1)^3 + (2K^2+2K+2i-1)O(M+1)^2 \tag{4.52}$$

which results in the following complexity for all $k, k = K+1, \cdots, N-1$:

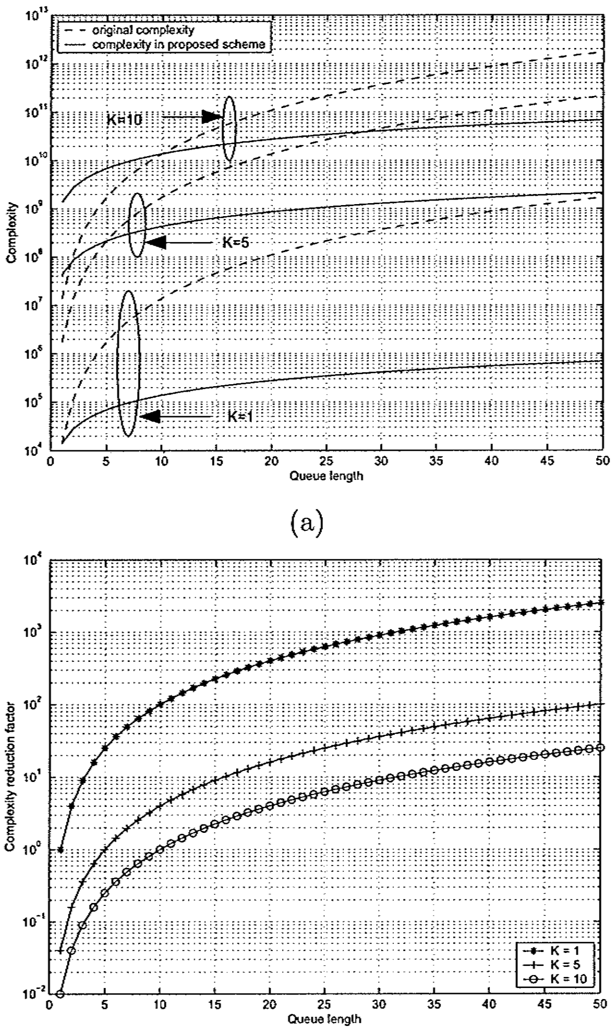
$$\begin{aligned} \mathbb{X}_2 &= \sum_{i=1}^{N-K-1} \mathbb{C}_{K+i} = (N-K-1)\frac{2K^2+K+N}{2} O(M+1)^3 \\ &+(N-K-1)(2K^2+K+N-2)O(M+1)^2. \end{aligned} \tag{4.53}$$

Finally, the complexity of evaluating (4.33), (4.46), and (4.47) is given as follows:

$$\mathbb{X}_N = (K+1)^2 O(M+1)^3 + (K^2+3K+3)O(M+1)^2. \tag{4.54}$$

The total complexity of policy evaluation is obtained as $\mathbb{X}_1 + \mathbb{X}_2 + \mathbb{X}_N$ which results in either $O(M^3K^3)$ or $O(M^3NK^2)$. In practice, $K \ll N$, and therefore, the complexity is $O(M^3NK^2)$ which is much smaller than $O(M^3N^3)$. In other words, the complexity reduces significantly when the QBD structure is exploited, and it increases only linearly with increasing queue length. On the other hand, consideration of multi-rate transmission on a $K$ state FSMC channel increases the complexity of computation by a factor of $K^2$, and since a $K^3$ factor is embedded in $M^3$, it ultimately results in the factor $K^5$ in the computational complexity. Therefore, the complexity would be reduced significantly if we ignore multi-rate transmission.

Fig. 4.3(a) compares the computational complexity between the original and the proposed schemes for different possible transmission rates in the system as well as



(a)

(b)

**Figure 4.3.** *(a) Comparison of computational of complexity between the original and the proposed schemes, and (b) reduction in complexity with the proposed scheme (K is the number of states in FSMC channel model).*

for the single-rate case for different values of queue length. While the complexity increases with increasing queue length for all values of $K$, increasing $K$ adds more to the computational complexity.

Based on the above analysis presented earlier, the complexity reduction factor in Fig. 4.3(b) is obtained by dividing the complexity of the original scheme by the complexity of the proposed scheme for different queue length. This figure shows that the complexity reduction factor increases with queue length. Also, adding multi-rate transmission capability to the system decreases this factor.

Although our approach reduces the order of computational complexity, it may still be significant for a sensor node. In such a case, the policies can be evaluated off-line by a supervisory node and then disseminated to the sensor nodes.

### 4.5.4 Policy Optimization in a Video Sensor Node

An optimum policy is of our interest to minimize the power consumption in a sensor node while achieving the minimum QoS performance requirements. We consider the average power consumption $\bar{\chi}$ given by $\bar{\chi}_{\phi_i} = \sum_{p_a \in \phi_i} \chi(a)$. Therefore, the energy cost vector is given by $\bar{\chi}_{\pi}^{n} = (\bar{\chi}_{\phi_1} \dots \bar{\chi}_{\phi_n})^T$. Given the initial probability distribution of the system $p^{(0)}$ and a policy $\pi$, we have $p^{(n)} = p^{(0)} P_{\pi}^{n}$ and the best estimate of future based on the available information is the expected value of the variable under investigation. We can set up a power optimization problem as shown in (4.55) where the objective is to obtain the optimum policy $\pi$ under a constrained QoS, i.e., $\mathcal{Q}(P, \pi)$ which is a function of transition matrix $P$ and policy $\pi$.

$$\begin{aligned} &\min_{\pi} \sum_{n=1}^{\infty} E_{\pi}[\bar{\chi}_{\pi}^{n}] \\ s.t. \quad &\mathcal{Q}(P, \pi) \leq \Delta. \end{aligned} \tag{4.55}$$

Each sensor node has a finite energy and when the battery is exhausted, there is no operational capability at the node. In the case of renewable batteries such as solar-powered sensor nodes, the operation is resumed when enough energy becomes available again. Hence, we assume that the node operates over a finite time horizon $\tilde{H}$, where $\tilde{H}$ is called the stopping time. The stopping time can be deterministic or random. For a random stopping time, we assume that the operation of the node at the beginning of each time slot continues with probability $0 < \kappa < 1$ (which is called the discount factor) and the lifetime of the node ends up with probability $1 - \kappa$.

Then the probability distribution of the system is given by $p^{(n)} = p^{(0)}(\kappa P)^n_\pi$. The expected value of energy cost is given by $\mathbf{E}_\pi[\bar{\chi}^n_\pi] = \kappa^n p^{(0)} P^n_\pi \bar{\chi}^n_\pi$. We can rewrite this equation as follows where the expectation is taken over all possible initial states in the system.

$$\mathbf{E}_\pi[\bar{\chi}^n_\pi] = E\left[\kappa^n P^n_\pi \bar{\chi}_\pi(x_n) | x_0 = x\right]. \tag{4.56}$$

Now, we replace (4.56) in (4.55) to obtain

$$\begin{aligned} &\min_\pi E\left[\sum_{n=1}^{\infty} \kappa^n P^n_\pi \bar{\chi}_\pi(x_n) | x_0 = x\right] \\ &s.t. \quad \mathcal{Q}(P, \pi) \leq \Delta. \end{aligned} \tag{4.57}$$

This is a discrete-time Markov control problem, and the solution can be formulated as a dynamic program as described before. The optimization problem above is the same as that in (1.15) where $g_\pi(x_n, n)$ is equal to $P^n_\pi \bar{\chi}_\pi(x_n)$ and the problem can be solved by using a dynamic programming approach so that the target level of QoS can be met.

## 4.6 Performance Evaluation

The performance of video transmission from a sensor node is analyzed for the proposed dynamic power management scheme and compared to the simulation results obtained in MATLAB. The performance of static power management is also investigated and compared to that for the dynamic scheme. Note that, for the static case no command is issued to control the sleep and wakeup process in either the camera or the transmitter. In other words, the controllable Markov chain falls back to a static Markov chain.

### 4.6.1 Parameters, Assumptions, and Performance Measures

For simplifying the analysis of the above mentioned general model during performance evaluation, we assume that the command by the power manager is only used to control the sleep mechanism in the wireless transmitter in the sensor node. This assumption limits the command vector issued by power manager to a single command. That is, $\sigma_{a,s}$ and $\sigma_{a,a}$ in Fig. 4.1 are not functions of command $a$ in this case (i.e., the sleep/capture dynamics in the camera are not under control of the power manager).

On the other hand, the transmitter adapts itself with the queue status as well as the channel status. The transmitter uses an exhaustive service discipline [53], where the transmitter remains in active (i.e., wakeup) mode until the queue is empty, in which case the transmitter goes to the sleep mode. It is also assumed that the wireless transceiver in the sensor node is able to estimate the channel state, for example, by using a distributed channel estimation technique. When the channel quality becomes very poor (e.g., state 0 in the FSMC model), to conserve energy, no transmission is scheduled. However, the power manager controls the sleep period of the transmitter and the transmitter stays in the sleep mode for a random period of time which is controlled by the power manager. The multiple vacation strategy [53] in the transmitter saves more energy by putting it again to sleep mode immediately at the end of sleep period when the queue is empty.

For the video traffic source model that is shown in Fig. 4.2, we assume a GoP pattern with $H = 12$ and $G = 3$. To find the probability of transition to another scene ($\rho_{i,j}$) in (4.1), we assume that the number of frames at each scene is a random number with maximum number of 5 frames in each scene. Moreover, the number of scene changes is also a random variable and the probabilities of scene change follow a stochastic process. The probability of packet arrival for different types of frames at each time slot is 0.10, 0.15, and 0.12 for I-, B- and P-frames, respectively. The queue length in the sensor node is assumed to be 20 and the frame length is distributed geometrically (a special case of phase-type distribution) with parameter $\vartheta_x$ which is assumed to be 0.15, 0.20 and 0.10 for frame types I, B, and P, respectively.

The camera remains in its current state with probability 0.2 (i.e., changes its state with probability 0.8). For the transmitter, the wakeup probabilities are chosen as 0.1

and 0.9 for the *sleep* and *wakeup* commands, respectively, and for the SPM scheme, this probability is 0.5. The value of the discount factor $\kappa$ is 0.9 and we assume that the camera and the transmitter consume the same amount of energy when they are in active mode and there is no energy consumption when they are in sleep mode. Therefore, the energy cost is 0 at each time slot when both of them are in sleep mode. Energy cost increases by one for each of the units to be in active mode. Moreover, we consider an energy cost equal to 0.1 for switching between sleep and active modes. The energy costs are properly applied to $g_{.,.}$ during policy evaluation.

For the FSMC channel, we use the same parameters as in [54] with average packet error rate in each of the transmission modes being equal to $P_0 = 0.05$. The value of the Nakagami parameter is $m = 1$ and the Doppler frequency is assumed to be $f_d = 10$ Hz for the fading channel.

For simulation-based performance analysis, the methodology for finding the proper action at each state and obtaining the performance measures is shown in Fig. 4.4. To obtain the best policy, we select a state (e.g. target state) and assign a deterministic action to the target state. Then we run the simulation 5000 times while a random action is assigned to each state except the target state. We repeat the procedure for all possible actions and at the end of this stage we assign the best action to the target state by comparing the performance measures. Repeating the same procedure for all states, we obtain the best policy. Having the best policy, we can do performance analysis by running the simulation again with the pre-assigned actions to each state.

For multimedia transmission over a sensor network, the subjective QoS requirements must be translated into objective QoS parameters such as the packet delay, loss, and throughput at the lower layers in the protocol stack. For performance evaluation, we consider the following QoS performance measures: (i) *energy saving factor*, which is defined as the sum of probabilities of the camera or transmitter to be in sleep mode so that energy is saved in the node, (ii) *frame dropping probability*, which is defined as the probability that a frame is dropped. It happens when the packets within the frame are dropped since the queue is full. (iii) *delay*, which is measured from the time each packet enters the queue till it is transmitted successfully, and (iv) *effective throughput*, which is defined as the ratio between the number of useful frames in the receiver and the total number of transmitted frames.

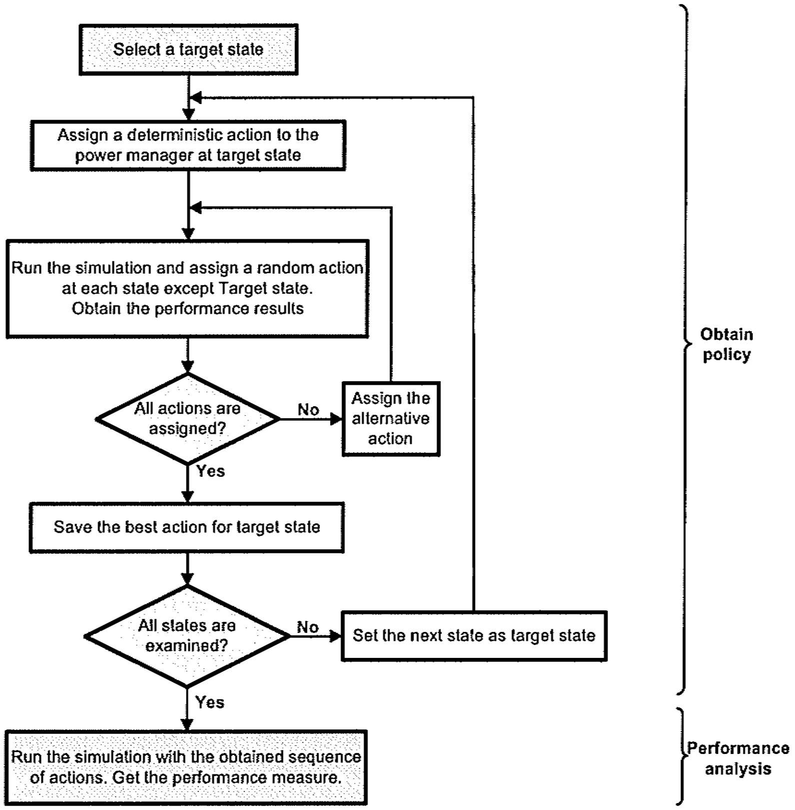

Select a target state
Assign a deterministic action to the power manager at target state
Run the simulation and assign a random action at each state except Target state. Obtain the performance results
All actions are assigned?
No
Assign the alternative action
Yes
Save the best action for target state
All states are examined?
No
Set the next state as target state
Yes
Run the simulation with the obtained sequence of actions. Get the performance measure.
Obtain policy
Performance analysis

**Figure 4.4.** *Methodology for simulation-based performance analysis.*

## 4.6.2 Numerical and Simulation Results

### 4.6.2.1 Energy Saving Factor

Fig. 4.5(a) shows optimum energy saving factor in a sensor node to meet the target frame dropping probability under different SNR for transmission channel when probability of service is assumed to be $\mu = 0.75$. As has been mentioned before, for the numerical results presented here, we assume that the camera is not under control of the power manager. The power manager monitors camera, transmitter, and queue length and it only controls the sleep period at the transmitter. With high SNR, it is more likely that the transmitter will be able to successfully transmit more packets from the queue and it will be in sleep mode for a relatively shorter period of time. As the number of successful transmissions increases, it provides more room in the queue for the new arrivals, and therefore, the dropping probability decreases.

Typical variations in energy saving factor are shown in Fig. 4.5(a) under different dropping probabilities for both the DPM and SPM schemes. For the SPM case, when the sleep and wakeup processes in the camera are independent of the state of the queue and the wireless transmitter, packet dropping probability increases (especially with low SNR). This is due to the fact that, as SNR decreases, the probability of successful transmission decreases (and hence the probability that the queue tends to be full becomes higher). Moreover, an increase in the dropping rate increases the chance of saving more energy. This clearly represents the fact that there exists a trade off between performance and energy saving in energy-limited wireless sensor nodes.

### 4.6.2.2 Delay

The average packet delay is obtained by simulation under different SNR and for different values of probability of service when frame dropping probability is 0.1 (Fig. 4.5(b)). For a persistent medium access protocol (as being considered here), a higher probability of service implies less energy consumption for transmission of same number of packets as well as smaller delay due to the higher chance of getting access to the shared medium. Therefore, the average delay decreases as the value of service probability increases. It also results in more energy saving in the system due to the shorter duration of active mode of transmission.

The DPM scheme outperforms the SPM scheme (Fig. 4.5(b)), which is due to the fact that the sleep and wakeup decisions in SPM are made at random without considering the state of the system.

#### 4.6.2.3 Frame Dropping Probability

For the DPM scheme, typical variations in average frame dropping probability with wireless channel quality for different types of frames are shown in Fig. 4.6(a). The average is taken on the results for policies with at least 25% energy saving factor. The numerical results are compared with the simulation results as well as with the results for the SPM scheme. We observe that lower dropping rates are achieved for the DPM scheme for all types of frames. However, dropping rate for I-frame is lower than other two types of frames. This is because an I-frame is dropped when any one of the packets in that frame is dropped. However, for the other frame types, a frame is dropped when the corresponding reference I-frame as well as any packet of its own are lost. The dropping probability of B-frames is higher compared to that of P-frames due to the larger lengths of B-frames in this case.

Here we assume that camera is not aware of the queue status and probable dropping of some packets due to insufficient space in the queue. The dynamic power management model can be enhanced where the camera in the sensor node is queue-aware (e.g., it goes to the sleep mode when the queue is full). Incorporating this feature would make the model more complex. However, it would increase the energy efficiency in the sensor node.

#### 4.6.2.4 Throughput and Effective Throughput

Throughput is another performance metric that must be compromised with the energy efficiency in a system. However, throughput, which is defined as the ratio of successfully received packets to all packets sent by the transmitter, is not a useful index for performance evaluation for correlated traffic like video. Instead, *effective throughput* is a more useful measure since it represents the useful information received at the receiver/data sink. In our system model, the received packets are useful only if they are received as a full frame. Moreover, frames without reference frame are wasted. In other words, effective throughput ($\tau$) is defined as follows:

$$\tau = \frac{\text{Packets belonging to fully successfully received frames with valid reference frame}}{\text{All received packets}}.$$

In Fig. 4.6(b), effective throughput is compared with *throughput* via simulation, for the case when no ARQ mechanism is applied. Probability of service is assumed to be $\mu = 0.9$ and the same policy as in Fig. 4.5(b) is adopted here for the DPM scheme. We observe that, for bursty video traffic, effective throughput is quite different from the raw throughput. Proper decision making can improve the throughput, and consequently, the effective throughput in the system.

## 4.7 Chapter Summary

A dynamic power management scheme based on a Markov decision process formulation has been presented in this chapter. The main objective of the proposed scheme is to save as much energy as possible in a sensor node while the specified QoS performance is achieved. Performance evaluation results have revealed that the proposed scheme is capable of improving the energy saving performance (and hence increasing the lifetime) while satisfying the QoS requirement of a video sensor node when compared to a static energy management scheme.

Several avenues for further research on this problem can be pursued. In this chapter, the effect of the battery has been represented by the discount factor in the analysis. This assumption has led us to the Markov stationary policies where off-line calculations are enough for decision making. However, the optimum decision making must be taken dynamically when the remaining battery power in a sensor node changes. Since the complexity of this approach may be huge for a sensor node, an adaptive learning approach with lower complexity would be preferable. A reinforcement learning approach is a promising approach which is investigated in Chapter 5.

Since I-frames are more important than other frames in an MPEG-coded video stream, a prioritization scheme for the I-frames should be considered to enhance the QoS performance. With prioritization among the different types of frames, the energy consumption in a wireless video sensor node can be optimized by selectively dropping the low priority data frames.

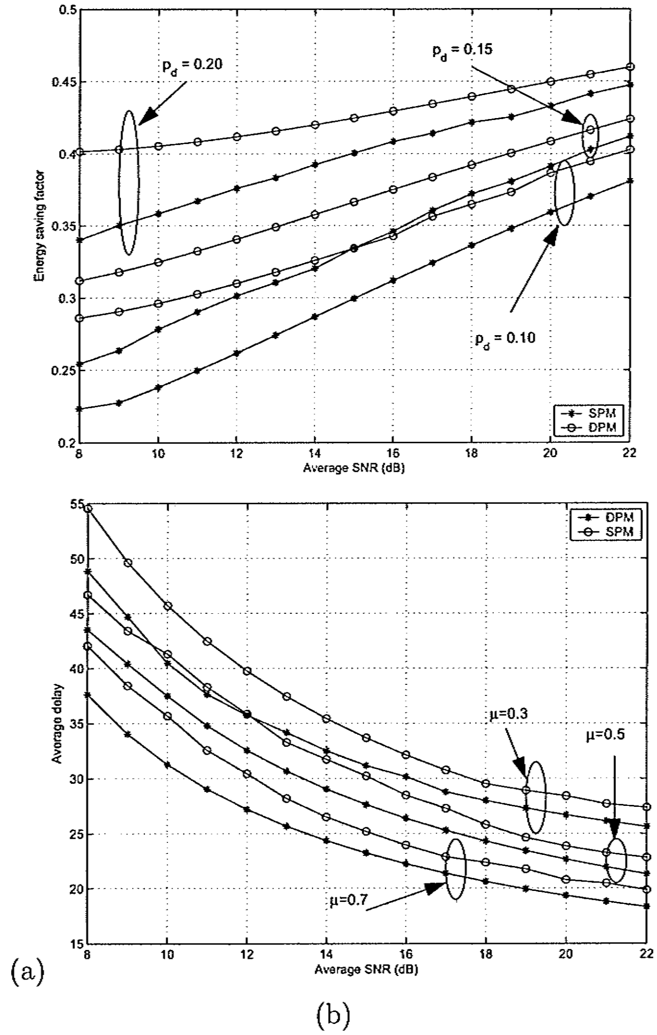
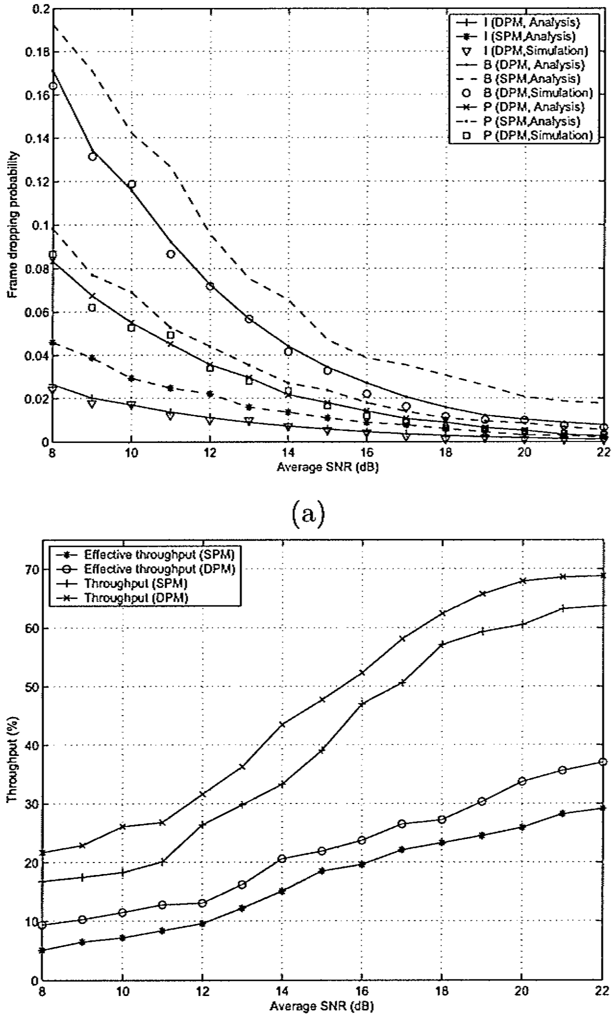


**Figure 4.5.** *(a) Energy saving factor for different dropping rates $p_d$ and for service probability $\mu$ = 0.75, and (b) average delay for different values of service probability $\mu$, and for $p_d = 0.1$.*



(a)

(b)

**Figure 4.6.** *(a) Average frame dropping probability for different frame types, and (b) effective throughput with no retransmission mechanism* ($\mu = 0.9$).

# Chapter 5

# Reinforcement Learning-Based Distributed Dynamic Power Management in Wireless Multimedia Sensor Networks

## 5.1 Introduction

Although dynamic programming [77] provides a suitable framework and algorithms to find optimal policies for distributed dynamic power management as introduced in Chapter 4, these algorithms become impractical when the underlying state-space of the Markov decision problem is large (i.e., *the curse of dimensionality*). On the other hand, reinforcement learning [60] techniques have been known to scale better than their dynamic programming counterparts [60] in solving the Markov decision problems. Reinforcement learning is a machine learning framework for autonomous agents (e.g. sensor nodes). An agent is trained to choose actions that maximize the long-run reward it receives. The objective is to find a policy that maximizes these rewards or minimizes the punishments. An MDP is mostly used as a model of the system in a model-based reinforcement learning scenario. In practice, the applicability of model-based learning may be prohibited due to the size of the underlying state space for a real-world problem. One approach to deal with this difficulty is to use model-free learning algorithm such as Q-learning. However, extensive training may be required when no system model is available.

In this chapter, we present a dynamic power management framework for transporting multimedia traffic over a distributed MAC protocol in a wireless multimedia sensor network. The distributed dynamic power management problem in a node is modeled as a Markov decision process. The MDP model is developed based on a priority queue model with vacation [53] for video/voice and data traffic. Here video/voice traffic is modeled using Batch Markovian arrival process (BMAP) and data traffic is modeled by using MAP. Video/voice traffic and data traffic arrivals correspond to high priority and low priority arrivals in a sensor node, respectively. The distribution of sleep time in a sensor node is assumed to have a phase-type distribution. Due to the complexity of model-based algorithms, and the slow convergence in the learning process for a model-free learning approach, a modified Q-learning approach called *Dyna* [60] is employed to optimize the energy consumption in the node under QoS constraints. The performance of the proposed framework is analyzed in terms of convergence, complexity, and different QoS metrics.

## 5.2 Background and Motivation

A reinforcement learning-based MAC framework, namely, the RL-MAC protocol was proposed in [89] for sensor networks with an objective to optimizing throughput and minimizing energy utilization. An underlying MDP was assumed which allowed nodes to infer the state of other nodes in the network in order to dynamically optimize their MAC policy. A near-optimal reinforcement learning framework was presented in [90] for energy-aware sensor communications which maximizes average throughput per unit of consumed energy.

A discounted reward Q-learning approach was presented in [91] for QoS provisioning in multimedia wireless networks. The problem was formulated as a Markov decision problem to find the optimal call admission control and bandwidth adaptation algorithms which can maximize network revenue under QoS constraints. Adaptive delay-aware transmission control algorithms for MIMO systems under unknown channel and traffic conditions were explored in [92]. The problem of rate and power adaptation under delay constraints was formulated as a constrained MDP and its solution was obtained in an on-line fashion employing Q-learning approach. The queueing dynamics in a wireless node was not considered in any one of the above works. The distributed dynamic power management framework presented in this paper considers the queueing dynamics at a sensor node under correlated and bursty traffic arrival (e.g., Markovian arrival, batch Markovian arrival) as well as general frame transmission (e.g., phase-type distribution for frame transmission time) scenario. Also, we model the service differentiation between multimedia sensor data (e.g., video/voice) and scalar data in a sensor node through a priority scheduling mechanism.

The MAP [56] has been extended to the batch arrival case, called the batch Markovian arrival process (BMAP) [58] to capture the bursty nature of traffic source behavior, and therefore, well suited to model multimedia data [59]. MPEG-4 video traffic was modeled as a discrete BMAP process in [93]. Phase-type distributions were shown to approximate many general distributions, particularly those used to model the service time [55]. A DBMAP/PH/1 queue with different service disciplines was studied in [94].

Priority queueing models are commonly used in telecommunications systems. A D-MAP/PH/1 priority queueing model was studied in [29], in which the arrival process was modeled as a single arrival MAP with two priority levels, and the service process was modeled to follow discrete-time phase-type distribution. No batch arrival was studied in this work. A queueing system consisting of a discrete BMAP arrival process with two priority levels and a single server with a phase-type service distribution was studied in [95] to capture the bursty and correlated nature of data traffic generated in communication networks. However, analysis of priority queues with BMAP arrival traffic was not studied for the case that server is allowed to be idle for some periods during its operation (i.e. vacation queueing). In this paper, we setup a queueing model for a sensor node where BMAP and MAP are used to model multimedia and data traffic arrival, respectively, and a vacation process at the server is used to model the sleep mode.

## 5.3 Queueing Dynamics at a Sensor Node Under Dynamic Power Management

### 5.3.1 Dynamic Power Management in a Multimedia Sensor Node

Dynamic power management is a general design methodology aiming at controlling performance and power levels of a system dynamically by exploiting the idleness of different parts of the system and performing selective shutdown of idle system resources. The ultimate goal of DPM is to achieve the maximum performance while minimizing the energy consumption. However, this objective is achievable only by an ideal power management scheme with complete knowledge of present, past, and future workloads. Due to the uncertainty in the communication/computation load in a wireless sensor node, a stochastic approach can be used to describe the system and to find the optimum policy to be used by the power manager.

The block diagram of a wireless multimedia sensor node with a dynamic power management model is shown in Fig. 5.1. The major parts of the system are high priority and low priority data queues and wireless transceiver. The wireless transmitter is responsible for major part of power consumption in the sensor node. The power manager monitors the system and observes the states of the system components and controls the power state of each component through its command for sleep or wakeup. Obviously, the sleep and wakeup dynamics at a sensor node affects the performance of data transmission by increasing the delay when the transmitter is shut down. Based on a policy, the power manager issues a command to control the system by turning on/off the controllable part(s) (i.e. the wireless transmitter in the sensor node in Fig. 5.1).

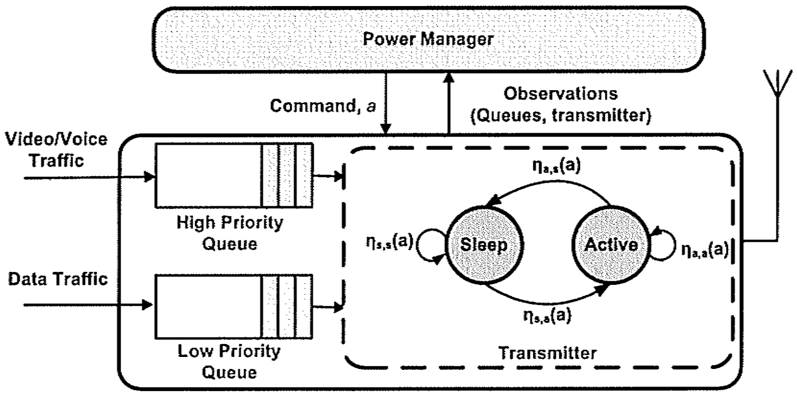


**Figure 5.1.** *Dynamic power management in a wireless multimedia sensor node.*

## 5.3.2 Queueing in a Multimedia Sensor Node: Arrival and Service Processes

### 5.3.2.1 MAP Arrival Process

The scalar sensor data are stored in the low-priority queue and their arrivals are modeled by a discrete Markovian arrival process (D-MAP). The D-MAP is described by transition matrix $H$ which is decomposed into substochastic matrices $H_0$ and $H_1$ such that $H = H_0 + H_1$. The element $(H_0)_{ij}$ represents the probability of transition from phase $i$ to phase $j$ with no arrival, and $(H_1)_{ij}$ represents a transition with an arrival at the node.

### 5.3.2.2 BMAP Arrival Process

We assume a discrete BMAP arrival (described by transition matrix $D$) process for the multimedia sensor data which are stored in the high priority queue. With maximum batch size of $N$, the transition matrix $D$ can be decomposed as $D = D_0 + D_1 + \cdots + D_N$, where $(D_k)_{ij}$ represents the probability of a transition from phase $i$ to phase $j$ with $k$ arrivals at the node.

BMAP arrival process is very general which can capture burstiness in the traffic arrival pattern. A special case of BMAP arrival process was used in [96] where the BMAP with batch size of $M$ was represented by a Markovian source with an $M+1$-state transition probability matrix. At each state $m$ ($m = 0, \ldots, M$), $m$ packets are generated in one time interval. This Markovian source is a BMAP process if we consider its states as the phases of the BMAP process. The batch Bernoulli arrival process in which $m$ packets ($m = 0, \ldots, M$) arrive in one time interval with probability $\lambda_m$ is also a special case of BMAP arrival process, where $\lambda_0 + \lambda_1 + \cdots + \lambda_M = 1$.

#### 5.3.2.3 Phase-Type Service Process

We assume that the service time[1] for both the queues follow phase-type distribution with parameters $(\delta, T)$ and $(\beta, S)$ for high-priority and low-priority queue, respectively. The service process follows standard preemptive resume discipline [97]. Low-priority packets are processed only when there are no high-priority packets in the system. The service for the low-priority queue is interrupted whenever high priority packets arrive. When the server returns to serving the preempted low-priority queue, the service process begins from the phase where the service was interrupted. We assume an exhaustive service process, that is, when the server becomes active, it continues serving all packets and returns to idle mode only if there is no packet to process.

The phase-type service process for getting the shared media and transmission of the packet is followed by a Bernoulli channel model with $\theta$ as the probability of error for the transmitted packet. A probabilistic retransmission policy is applied in the node for video/voice traffic. That is, a corrupted packet is rescheduled for transmission with probability $q_r$ (i.e., dropped with probability $1 - q_r$). Note that, this probabilistic retransmission policy falls back to infinite retransmission and no retransmission policies when $q_r$ is equal to 1 and 0, respectively. For data traffic, an infinitely persistent retransmission policy is applied.

### 5.3.3 Probability Transition Matrix

The states of the multimedia sensor node can be described by a Markov chain the state space for which, in general, can be described as $\mathcal{X} = \mathcal{K} \times \mathcal{L} \times \mathcal{D} \times \mathcal{H} \times \mathcal{T} \times \mathcal{S} \times \mathcal{V}$,

[1] It refers to the time required for a packet to be transmitted for the first time from the queue since its arrival. Note that, after transmission, the packet could be corrupted due to wireless error. The term 'packet' is used here in a generic sense noting the fact that the MAC layer transmission units are generally referred to as frames.

where $\mathcal{K}$ and $\mathcal{L}$ are the number of packets in the high priority queue and the low priority queue, $\mathcal{D}$ and $\mathcal{H}$ are the phases of arrival for high priority and low priority packets, respectively. Also, $\mathcal{T}$ and $\mathcal{S}$ are phases of service process for high priority and low priority queues, respectively. Finally, $\mathcal{V}$ is the phase of the sleep process.

$$P = \begin{array}{c} 0 \\ 1 \\ \\ \\ \\ \\ \\ \\ \\ K \end{array} \left[ \begin{array}{cccccccccc} B_0 & B_1 & B_2 & B_3 & \dots & B_N & & & & \\ C_0 & A_0 & A_1 & A_2 & \dots & A_{N-1} & A_N & & & \\ & C_1 & A_0 & A_1 & \dots & A_{N-2} & A_{N-1} & A_N & & \\ & & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & \\ & & & C_1 & A_0 & A_1 & \dots & A_{N-2} & A_{N-1} & A_N \\ & & & & C_1 & A_0 & A_1 & \dots & A_{N-2} & E_1 \\ & & & & & \ddots & \ddots & & \ddots & \vdots \\ & & & & & & C_1 & \dots & A_{N-i-1} & E_i \\ & & & & & & & \ddots & \dots & \vdots \\ & & & & & & & & C_1 & E_N \end{array} \right] \tag{5.1}$$

Transition matrix $P$ in (5.1) denotes the transition matrix for this Markov chain where $N$ is the batch size in BMAP arrival model for video/voice traffic. We also assume that the length of high priority and low priority queues is $K$ and $L$, respectively. For the service processes, we assume that $\delta, T, \beta$ and $S$ are of dimensions $1 \times n_h$, $n_h \times n_h$, $1 \times n_l$ and $n_l \times n_l$, respectively. In general, matrix $P$ is a level-dependent matrix. Note that, $E_i = \sum_{j=0}^{i} A_{N-j}$. Description of the block matrices are as follows where $\otimes$ denotes kronecker product and $I_{n_l}$ is identity matrix of size $n_l$ wherever applicable.

The matrices $B_k, k = 0, \dots, N$ represent arrival of video/voice traffic with batch size of $k$ at the node when the high priority queue is empty.

$$B_k = \left[ \begin{array}{cccccc} \tilde{B}_{k1} & \tilde{B}_{k2} & & & & \\ \tilde{B}_{k0} & B_{k1} & B_{k2} & & & \\ & B_{k0} & B_{k1} & B_{k2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & B_{k0} & B_{k1} & B_{k2} \\ & & & & B_{k0} & B_{k1} + B_{k2} \end{array} \right] \tag{5.2}$$

Submatrices $\tilde{B}_{k1}$ and $\tilde{B}_{k2}$ in $B_k$ represent no arrival and an arrival in low priority queue, respectively, for the case that a batch arrival of size $k$ occurs in the high priority queue and low priority queue is empty. They are given as follows.

$$\begin{aligned}\tilde{B}_{k1} &= D_k \otimes H_0 \otimes \begin{bmatrix} V & V^0 \otimes \delta \\ 0 & 0 \end{bmatrix} &(5.3)\\ \tilde{B}_{k2} &= D_k \otimes H_1 \otimes \begin{bmatrix} V & V^0 \otimes (\delta \otimes \beta) \\ 0 & 0 \end{bmatrix} &(5.4)\end{aligned}$$

$B_{k1}$ and $B_{k2}$ have the same description as $\tilde{B}_{k1}$ and $\tilde{B}_{k2}$ respectively, when low priority queue is not empty.

$$\begin{aligned}B_{k1} &= D_k \otimes H_0 \otimes \begin{bmatrix} V & V^0 \otimes \delta \\ 0 & \delta \otimes (S + S^0\theta) \end{bmatrix} \\ &\quad + D_k \otimes H_1 \otimes \begin{bmatrix} 0 & 0 \\ 0 & \delta \otimes S^0(1-\theta)\beta \end{bmatrix} &(5.5)\\ B_{k2} &= D_k \otimes H_1 \otimes \begin{bmatrix} V & V^0 \otimes \delta \\ 0 & \delta \otimes S + \delta \otimes S^0\theta\beta \end{bmatrix} &(5.6)\end{aligned}$$

In $\tilde{B}_{k0}$ and $B_{k0}$, a packet in low priority queue is served and in $B_{k2}$, an arrival of data packets occurs and no packet in low priority queue is served in this time interval. In $B_{k1}$, either one data packet is served and an arrival occurs or no service and arrival occur in the low priority queue in this time interval.

$$\tilde{B}_{k0} = D_k \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & \delta \otimes S^0(1-\theta) \end{bmatrix} \tag{5.7}$$

$$B_{k0} = D_k \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & \delta \otimes S^0(1-\theta)\beta \end{bmatrix} \tag{5.8}$$

When there is no arrival in high priority queue (i.e. for batch size 0), submatrices in $B_0$ are as follows.

$$\tilde{B}_{01} = D_0 \otimes H_0 \otimes \begin{bmatrix} V & V^0\nu \\ 0 & 0 \end{bmatrix} \tag{5.9}$$

$$\tilde{B}_{02} = D_0 \otimes H_1 \otimes \begin{bmatrix} V & V^0 \otimes \beta \\ 0 & 0 \end{bmatrix} \tag{5.10}$$

$$\tilde{B}_{00} = D_0 \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ S^0(1-\theta) \otimes \nu & 0 \end{bmatrix} \tag{5.11}$$

and

$$\begin{aligned} B_{01} &= D_0 \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & S + S^0\theta \end{bmatrix} \\ &\quad + D_0 \otimes H_1 \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0(1-\theta)\beta \end{bmatrix} \end{aligned} \tag{5.12}$$

$$B_{02} = D_0 \otimes H_1 \otimes \begin{bmatrix} 0 & 0 \\ 0 & S + S^0\theta \end{bmatrix} \tag{5.13}$$

$$B_{00} = D_0 \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & S^0(1-\theta)\beta \end{bmatrix}. \tag{5.14}$$

Matrices $C_0$ and $C_1$ represent serving one video/voice packet when there is no arrival of multimedia traffic. In $C_0$, the high priority queue will be empty. The service phase for the low priority packet in service, if any, is tracked and the service is resumed in that phase after the high priority packets have been served.

$$C_i = \begin{bmatrix} \tilde{C}_{i1} & C_{i2} \otimes \beta & & & \\ & C_{i1} \otimes I_{n_l} & C_{i2} \otimes I_{n_l} & & \\ & & \ddots & \ddots & \\ & & & C_{i1} \otimes I_{n_l} & C_{i2} \otimes I_{n_l} \\ & & & & (C_{i1} + C_{i2}) \otimes I_{n_l} \end{bmatrix}, \; i = 0, 1. \tag{5.15}$$

$C_{i1}$ and $C_{i2}$ represent no arrival and one arrival of data packets, respectively, when the low priority queue is not empty. $\tilde{C}_{i1}$ and $\tilde{C}_{i2}$ have the same definition for the case that the low priority queue is empty.

$$\tilde{C}_{01} = D_0 \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ (T^0(1-\theta) + T^0\theta(1-q_r)) \otimes \nu & 0 \end{bmatrix} \tag{5.16}$$

$$C_{01} = D_0 \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & T^0(1-\theta) + T^0\theta(1-q_r) \end{bmatrix} \tag{5.17}$$

$$C_{02} = D_0 \otimes H_1 \otimes \begin{bmatrix} 0 & 0 \\ 0 & T^0(1-\theta) + T^0\theta(1-q_r) \end{bmatrix} \tag{5.18}$$

$$\tilde{C}_{11} = C_{11} = D_0 \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & (T^0(1-\theta) + T^0\theta(1-q_r))\delta \end{bmatrix} \tag{5.19}$$

$$C_{12} = D_0 \otimes H_1 \otimes \begin{bmatrix} 0 & 0 \\ 0 & (T^0(1-\theta) + T^0\theta(1-q_r))\delta \end{bmatrix}. \tag{5.20}$$

$A_k$ $(k = 0, \ldots, N-1)$ represents either a batch arrival of video/voice packets with size $k$ when no multimedia packet is served in that time interval or the batch size is $k+1$ and a multimedia frame is served.

$$A_k = \begin{bmatrix} A_{k1} & A_{k2} \otimes \beta & & & \\ & A_{k1} \otimes I_{n_l} & A_{k2} \otimes I_{n_l} & & \\ & & \ddots & \ddots & \\ & & & A_{k1} \otimes I_{n_l} & A_{k2} \otimes I_{n_l} \\ & & & & (A_{k1} + A_{k2}) \otimes I_{n_l} \end{bmatrix}, k = 0, 1, \ldots, N-1 \tag{5.21}$$

In either case, $A_{k1}$ denotes no change in the number of the packets in the low priority queue and results from no arrival and no service for the data packets. In $A_{k2}$, a data packet arrival occurs and no packet from the low priority queue is served.

$$\begin{aligned} A_{k1} &= D_k \otimes H_0 \otimes \begin{bmatrix} V & V^0 \otimes \delta \\ 0 & T + T^0\theta q_r \end{bmatrix} \\ &+ D_{k+1} \otimes H_0 \otimes \begin{bmatrix} 0 & 0 \\ 0 & (T^0(1-\theta) + T^0\theta(1-q_r))\delta \end{bmatrix} \qquad (5.22) \\ A_{k2} &= D_k \otimes H_1 \otimes \begin{bmatrix} V & V^0 \otimes \delta \\ 0 & T + T^0\theta q_r \end{bmatrix} \\ &+ D_{k+1} \otimes H_1 \otimes \begin{bmatrix} 0 & 0 \\ 0 & (T^0(1-\theta) + T^0\theta(1-q_r))\delta \end{bmatrix} \qquad (5.23) \end{aligned}$$

$$\begin{aligned} A_{N1} &= D_N \otimes H_0 \otimes \begin{bmatrix} V & V^0 \otimes \delta \\ 0 & T + T^0\theta q_r \end{bmatrix} \qquad (5.24) \\ A_{N2} &= D_N \otimes H_1 \otimes \begin{bmatrix} V & V^0 \otimes \delta \\ 0 & T + T^0\theta q_r \end{bmatrix}. \qquad (5.25) \end{aligned}$$

A dynamic power management policy can be designed where the sleep and the wakeup transitions in the Markov chain are controlled by a command $a \in$ {sleep, active} issued by the power manager in the node. This results in a controllable Markov chain which can be described as a Markov decision process [88]. For this controllable Markov chain, the transition matrix $P$ in (5.1) becomes $P(a)$ which is a function of the command $a$ issued by the power manager. Specifically, for the system model shown in Fig. 5.1, based on the command $a$, the probability of transition from state $i$ to state $j$ (i.e., $\eta_{i,j}(a)$ in the wireless transmitter, where $i, j \in$ {sleep, active}), can be varied as functions of $a$, based on the system information in the power manager. This will result in a probabilistic sleep and wakeup mechanism for energy saving in the sensor node.

## 5.4 Analysis and Optimization of the Power Management Model

### 5.4.1 Q-Learning

Q-learning is a popular model-free, temporal difference reinforcement technique with a strong proof of convergence established in [62]. Let $Q(x_k, a)$ be the expected discounted reward of taking action $a$ in state $x_k$, and then continuing by choosing actions optimally. Then $Q(x_k, a)$ can be written recursively at time $n$ [63] as follows:

$$Q_{n+1}(x_n, a_n) = g_{a_n}(x_n) + \alpha \sum_{y} p_a(x_n, y) \min_{\acute{a}} Q_n(y, \acute{a}) \tag{5.26}$$

where $\alpha$ is the discount factor with values between 0 and 1. The closer this value to 1, for the learning agent, the more important the future costs/rewards are compared to the immediate costs/rewards. The setting for $\alpha$ can be fixed or gradually decaying over time.

However, when the model is unknown or it is expensive to compute the model, we implicitly estimate the expectation above based on the state transitions observed in system trajectories. Then, $Q(x, a)$ can be obtained as follows:

$$Q_{n+1}(x_n, a_n) = g_{a_n}(x_n) + \alpha \min_{\acute{a}} .Q_n(x_{n+1}, \acute{a}). \tag{5.27}$$

In fact, $Q_n(x_{n+1}, \acute{a})$ is a noisy estimate of $\sum_y p_a(x_n, y) \min_{\acute{a}} Q_n(y, \acute{a})$ and the algorithm may not converge. To make it converging, we attenuate the noise at each step as follows:

$$Q_{n+1}(x_n, a_n) = (1 - \varrho) Q_{x_n, a_n} + \varrho \left[ g_{a_n}(x_n) + \alpha \min_{\acute{a}} Q_n(x_{n+1}, \acute{a}) \right] \tag{5.28}$$

where $\varrho$ is a discounting factor or *learning rate* with values between 0 and 1. A value of 0 means that the Q-values are never updated (nothing is learned). A higher value implies that the learning process can occur more quickly.

### 5.4.2 Dyna Algorithm for Dynamic Power Management

Although Q-learning is an effective and well-known learning method, it often requires many experiences before converging to an optimal policy. *Dyna* architecture [60] exploits a middle ground, yielding strategies that are more effective than model-free learning and also computationally more efficient than the certainty-equivalence approach. It simultaneously uses experience to build a model and uses the model to adjust the policy.

The Dyna architecture for dynamic power management in a sensor node is depicted in Fig. 5.2. Here, real experience is the result of interaction between agent (i.e. sensor node) and environment (i.e. arrival traffic, service process and sleep mechanism in the node). Real experience is used for direct reinforcement learning update to improve the value function and the policy. The model-based part of the process is on the right hand side where the simulated experience arises from the model which is learned by real experience.

MDP is used here as a stochastic model for the environment. Sleep probability is the parameter in the MDP which is to be tuned in the model by learning from the environment, the actions taken, and the costs incurred. The planning is obtained by applying reinforcement learning methods to the simulated experiences just as if they had really happened. The power manager uses the policy or value function saved in the node to generate a command for the next action in the node at each state.

After each experience, Dyna updates the model, incrementing the statistics for the specified state transition. It also updates the policy at that state based on the newly updated model using the following rule:

$$Q(x,a) = g_a(x) + \alpha \sum_y p_a(x,y) \min_{\acute{a}} Q(y,\acute{a}). \tag{5.29}$$

Dyna performs $m$ additional updates. It chooses $m$ state-action pairs at random and updates them according to the same rule as before. Finally, Dyna chooses action $\acute{a}$ to perform in state $y$ based on the $Q$ values obtained from *exploration* strategy in (5.28).

The Dyna algorithm requires about $m$ times the computation of Q-learning per instance, but this is significantly less than that for a naive model-based method. A reasonable value of $m$ can be determined based on the relative speeds of computa-

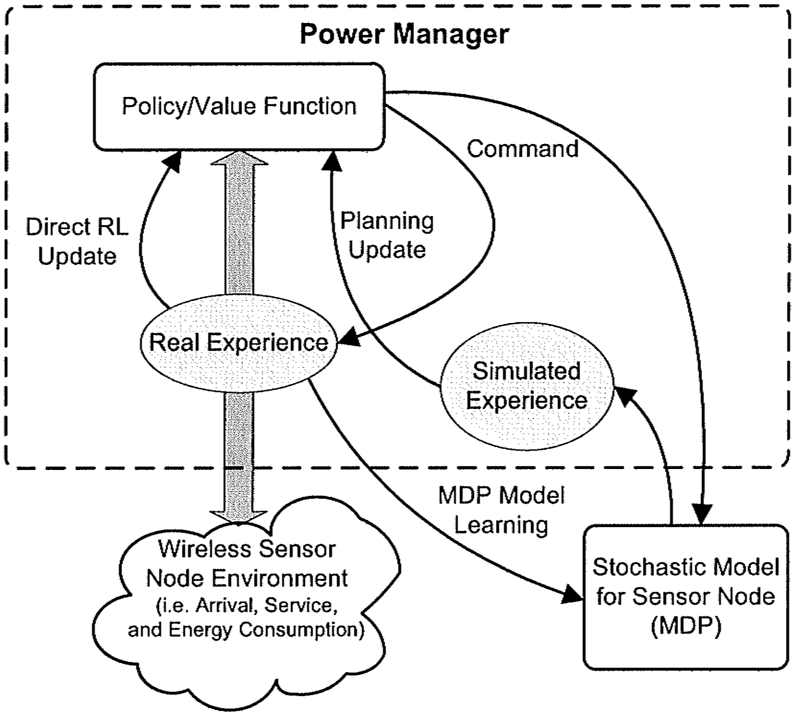


**Figure 5.2.** *Dyna Architecture for dynamic power management.*

tion and of taking action. The Dyna algorithm for dynamic power management is summarized in Fig. 5.3.

## 5.5 Performance Evaluation

### 5.5.1 Assumptions, System Parameters, and Performance Measures

The transmitter in Fig. 5.1 adapts itself with the queue status. We consider an exhaustive service discipline [53] in which the transmitter remains in active (i.e., wakeup) mode until the queue is empty, and then it goes to the sleep mode.

The power manager controls the sleep period of the transmitter. The transmitter stays in the sleep mode for a random period of time which is controlled by the power manager. The multiple vacation strategy [53] is assumed for the transmitter which

```
1. $x \leftarrow$ current state.
2. Choose an action, $a$, and take it.
3. Receive next state, $\acute{x}$, and cost, $g$.
4. Apply RL backup to $x$, $a$, $\acute{x}$, $g$ (e.g., Q-learning update).
5. Update Model(x,a) with $\acute{x}$, $g$.
6. Repeat $m$ times:
       select a previously seen state-action pair $x$,$a$.
       $\acute{x}$,$g \leftarrow$ Model($x$,$a$).
       Apply RL backup to $x$, $a$, $\acute{x}$, $g$.
7. Go to 1.
```

**Figure 5.3.** *Dyna algorithm for dynamic power management.*

saves more energy by putting it again to sleep mode immediately at the end of sleep period when the queue is empty.

We assume that the voice and video traffic can be modeled by an ON/OFF source as depicted in Fig. 5.4. In the *OFF* state, there is no voice or video traffic; in the *Voice ON* state, there is voice traffic with a fixed batch size of 1; and in the *Video ON* state, there is video traffic with a batch size ranging from 1 to 8. We assume that the voice packet size is fixed and the video packet batch size follows a log-normal distribution. The probability mass function for the video traffic batch size $v[i], i = 1, \ldots, 8$, is given by $[0.002, 0.153, 0.427, 0.286, 0.099, 0.025, 0.006, 0.002]$ as in [95].

Under the above assumptions, the BMAP arrival process for voice and video traffic has the following $D_k$ matrices

$$D_0 = \begin{bmatrix} 1 - h_1 - h_2 & 0 & 0 \\ g_2 & 0 & 0 \\ g_1 & 0 & 0 \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 & h_2 & h_1 v[1] \\ 0 & 1 - g_2 & 0 \\ 0 & 0 & (1 - g_1) v[1] \end{bmatrix}$$

$$D_i = \begin{bmatrix} 0 & 0 & h_1 v[i] \\ 0 & 0 & 0 \\ 0 & 0 & (1 - g_1) v[i] \end{bmatrix}, \quad 2 \leq i \leq 8. \tag{5.30}$$

We set the parameters of the arrival process as $h_1 = 0.1$, $g_1 = 0.9$, $h_2 = 0.2$, and $g_2 = 0.9$. For low priority queue (i.e., for data traffic), we assume a geometric

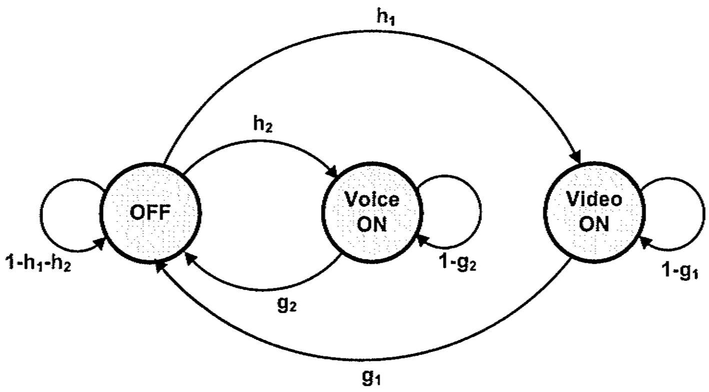


**Figure 5.4.** *Video and voice traffic model.*

inter-arrival process. The probability of arrival is assumed to be $\xi$ with default value of $\xi = 0.25$. Retransmission probability is set to 1 by default which means infinite retransmission policy is the default policy for both types of arrivals.

Let $N_{xy}^{a}(k)$ be the number of times the system state changes from $x$ to $y$ under action $a$ before the $k^{th}$ decision epoch, and $N_{x}^{a}(k)$ be the number of times of executing action $a$ in state $x$ before the $k^{th}$ epoch. Then, the controlled state transition probabilities at the $k^{th}$ decision epoch are estimated as follows:

$$p_{xy}^{a}(k) = \begin{cases} \frac{N_{xy}^{a}(k)}{N_{x}^{a}(k)}, & \text{if} \quad x = x_{k-1}, y = x_k, a = a_{k-1} \\ p_{xy}^{a}(k-1), & \text{otherwise.} \end{cases} \tag{5.31}$$

We use (5.31) above to tune the sleep process in a stochastic fashion. In other words, the action stored in Q-table is not directly applied to the node to control the sleep period. Instead, a node learns how to implement the command. The recursive procedure above corrects itself and leads to an optimum strategy for the stochastic model.

We assume a simple case of phase-type distribution for sleep process where the node stays in sleep mode at the end of each interval with probability $\gamma$ or wakes up with probability $1 - \gamma$. The sleep parameter $\gamma$ is controlled by command $a$ from the power manager where the parameter $\gamma$ is tuned as above. The default value of $\gamma$ is assumed to be 0.2.

The updated MDP is used in Dyna architecture in (5.29) and finally $Q$ values are obtained from (5.28) where we choose $\alpha = 0.9$ and $\varrho = 0.95$. The value of $m$ in the Dyna algorithm is set to values ranging from 5 to 20 for comparison. We also assume that the service process is geometric for both high priority and low priority queues with the probability of service being $\mu = 0.5$. Moreover, both queues are limited and their queue lengths are set as $K = L = 30$.

We use a single metric to quantify the QoS performance and the energy consumption performance in a sensor node. For this, a cost function ($g$) is defined as follows:

$$g = w_{qos}^{(h)} d^{(h)} + w_{qos}^{(l)} d^{(l)} + w_{energy} \eta \tag{5.32}$$

where $d^{(h)}$ and $d^{(l)}$ denote, respectively, the average delay for high-priority and low-priority packets, and $\eta$ denotes energy consumption. Here, $w_{qos}^{(h)}$, $w_{qos}^{(l)}$, $w_{energy}$ denote, respectively, the weights corresponding to QoS performance (i.e., average packet delay) of high and low priority packets and energy consumption.

For energy consumption, we assume that a unit of energy is consumed at each epoch the node is active. We also assume that the switching between the sleep and active modes requires 0.1 unit of energy. There is no energy consumption when the node is in sleep mode. The weights in (5.32) are set as follows: 0.4,0.1 and 0.5 for $w_{qos}^{(h)}$, $w_{qos}^{(l)}$, and $w_{energy}$, respectively.

## 5.5.2 Performance Results

We use a discrete-time event-driven simulation in MATLAB to evaluate the performance of the power management framework in a sensor node. For Q-learning, the proposed scheme is trained and the Q values are learnt by running the simulation for 20 million time slots.

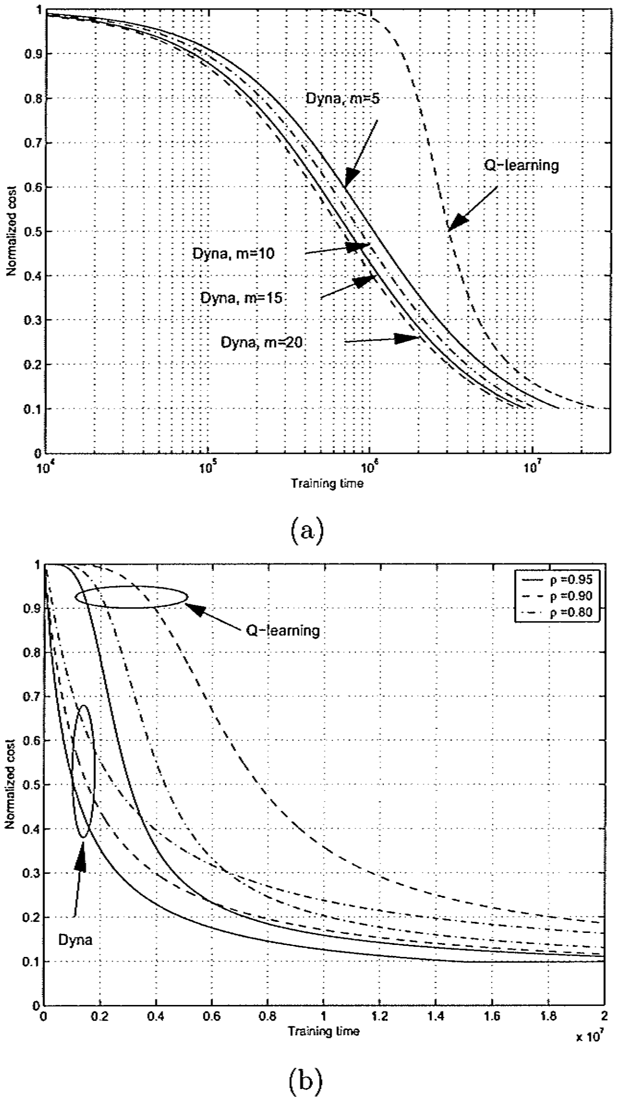
### 5.5.2.1 Learning Curve and Adaptation

The normalized cost function, which is defined as the value of cost function at each time divided by the initial cost value, is depicted in Fig. 5.5(a) which we refer to as the *learning curve* for the system and it shows the performance improvement in the node. The learning curve for Q-learning algorithm is compared to that for the Dyna architecture in Fig. 5.5(a). To obtain the results in Fig. 5.5(a), for each algorithm, an average is taken over the results for each simulation run where the initial costs are assigned randomly.

As is evident from Fig. 5.5(a), the Dyna algorithm provides superior performance than the Q-learning algorithm in terms of training time. This result implies that, with Dyna algorithm, a sensor node is able to follow the changes in the environment (e.g. channel dynamics, changes in battery state) faster compared to the Q-learning algorithm. Therefore, Dyna is a more adaptable algorithm in a dynamic environment. For Dyna, the learning curves for different values of $m$ show that a small value of $m$ can increase the learning speed without considerable increase in complexity. Larger values of $m$ may increase the complexity without enough gain in the learning speed.

All reinforcement learning algorithms use the cost (reward) function as a target to be minimized (maximized). This function may be calculated locally or broadcast periodically by the sink node. However, it may change during the time of node's operation. In a dynamic environment, *adaptation* refers to making necessary changes in the policy by the power manager so that the optimum performance can be obtained under the new environment as soon as possible. Higher learning rate in the reinforcement learning algorithm makes it faster to adapt the node to the new requirements. Fig. 5.5(b) shows how the learning curves change for both the Dyna algorithm and the Q-learning algorithm when we allow 20% changes in the initial weights of cost function $g$ in (5.32). Dyna algorithm learns faster compared to the Q-learning algorithm given that the model is perfect and stable. This figure also shows how the learning rate parameter ($\varrho$) affects learning time in both the algorithms.

On the other hand, a change in network dynamics may cause changes in the parameters of the model. This includes arrival traffic characteristics, service time parameters, and wireless channel condition. In a model-based scenario, we may assume that the parameters are broadcast periodically. Therefore, if learning is in progress (which is a must for a dynamic environment), the model-based algorithm uses the updated model to update the Q-functions and reaches to the optimum policy faster. In a model-free algorithm such as Q-learning, it takes more time to do such adaptation because of lack of reference. In the Dyna algorithm, the higher the values of $m$ and the discount factor, the faster is the learning rate and the adaptation. This is

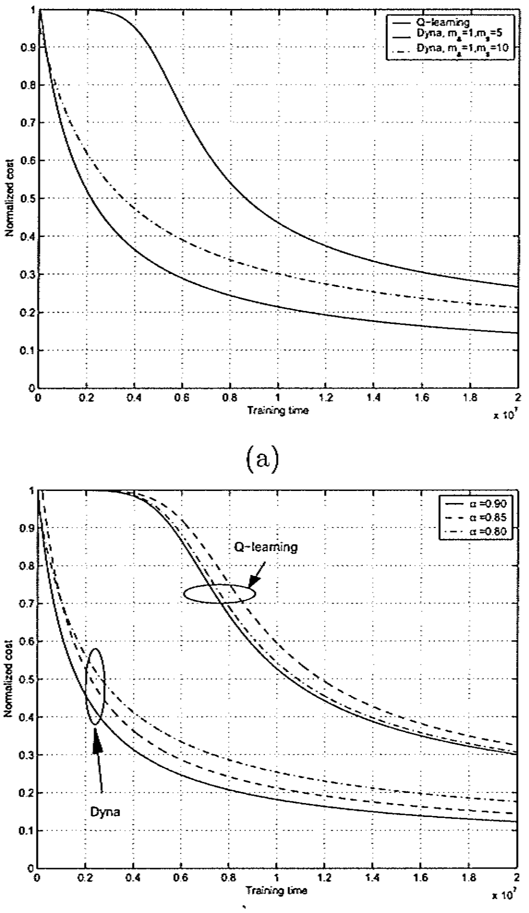


(a)

(b)

**Figure 5.5.** *The learning curve: (a) effect of simulated experiences m , and (b) effect of learning rate ρ.*

because the sensor node relies more on the model and the simulated experiences than the real experiences.



(a)

(b)

**Figure 5.6.** *The learning curve: (a) adaptation in Dyna algorithm ($m_a$ and $m_s$ denote the number of simulated experiences in active mode and sleep mode, respectively), and (b) effect of discount factor $\alpha$.*

We exploit this capability of Dyna algorithm in the sensor node in sleep periods. Since the power consumption in sleep period (i.e. in processing units) is negligible compared to active mode (i.e. in transmitter), we apply different values of $m$ for active and sleep periods. During active period, when the node is transmitting and it requires fast updates, $m$ is chosen to be a small number. On the other hand, we choose a higher value of $m$ in sleep periods when the node is idle.

In Fig. 5.6(a), we change the arrival traffic for some random time during the training period. The changes include up to 25 percent change in data arrival rate and parameters $h_1$ and $h_2$ in the multimedia traffic model. For the Dyna algorithm, we assume two different values of $m$, namely, $m_a$ and $m_s$ for the intervals when the sensor is in active and in sleep mode, respectively. Although the learning rate is slower compared to the case of static model, it shows how increasing $m_s$ improves the learning rate. On the other hand, the effect of changes in the model on learning time in the Dyna architecture is less than that on the Q-learning algorithm which is a model-free algorithm. This becomes evident from the comparison between Fig. 5.6(a) and Fig. 5.5(b).

In Fig. 5.6(b) we investigate the effect of the same changes in the model on the learning time when we apply different discount factors in both algorithms. Compared to Fig. 5.5(b), all learning curves are slower now. However, it affects the Q-learning algorithm more. We observe that, increasing the discount factor increases the learning speed.

#### 5.5.2.2 Complexity

In analyzing the reinforcement learning algorithms, there are two principle sources of complexity to be considered. The first one is the *sample complexity*, which defines the amount of real experience needed by an algorithm to achieve the result. The second one is the *computational complexity*, which specifies the amount of computational work required per experience. An algorithm with low sample and computational complexity is desired.

The Dyna architecture lies between the model-free and model-based algorithms. Figs. 5.5-5.6 show how learning the transition model in Dyna can decrease sample complexity at the price of increased computational complexity. Choosing a proper value of $m$ in the Dyna algorithm can make a compromise so that the sample complexity is decreased with only small increase in the computational complexity. Computational complexity in Dyna is much less than that of a model-based algorithm. Meanwhile, learning rate can be higher than a model-free algorithm such as a Q-learning algorithm.

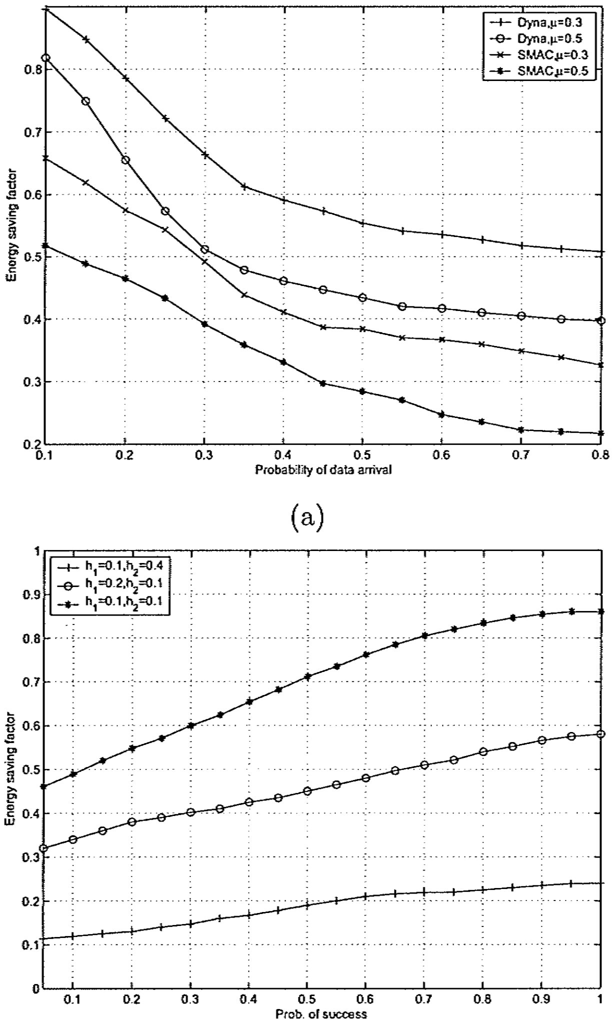
On the other hand, simulated experiences in Dyna may be done off-line for the case that training is in progress while the node is in a dynamic environment. In this case, a node regularly does the real experience. However, simulated experiences may be scheduled off-line during time when the workload is lower at the node. This is the idea behind implementing two different $m$ values for active and sleep modes in Fig. 5.6(a).

#### 5.5.2.3 Energy Efficiency Performance

Fig. 5.7(a) shows energy saving factor in a sensor node under different arrival traffic loads. Energy saving factor is defined as the probability of the transmitter to be in sleep mode so that energy is saved in the node. The graph shows the variation of energy saving factor for different service rates in the node under a perfect channel condition (i.e., $\theta = 0$). With a higher service rate, it is more likely that the transmitter will be able to successfully transmit more packets from the queue. Therefore, it will be in active mode for a relatively shorter period of time and will save more energy.

The energy efficiency for Dyna algorithm is also compared in Fig. 5.7(a) with the energy efficiency of SMAC protocol [8]. SMAC protocol uses a fixed duty cycle and we assume that for SMAC the ratio between sleep and active period is 0.20. We observe that Dyna offers more energy efficiency when compared to SMAC.

The effect of channel error on energy saving factor is depicted in Fig. 5.7(b) for different multimedia arrival traffic. Increasing the probability of successful transmission increases the energy efficiency in the node. This is because, as the probability of error in the channel increases, it increases the number of packets waiting for transmission in the queue. To serve the packets in the queue, the exhaustive discipline in the transmitter causes the node to be in active mode for a longer period of time. Moreover, this figure shows how variation in arrival traffic affects the energy saving factor. As traffic load decreases, it improves the energy saving performance especially when the channel condition improves.



(a)

(b)

**Figure 5.7.** *Energy saving factor (a) for different probability of arrival of data traffic ($\mu$ is probability of service), and (b) for different video/voice traffic parameters ($h_1$ and $h_2$) and probability of success.*

#### 5.5.2.4 Delay Performance

The average packet delay performance for multimedia traffic as well as data traffic is shown in Fig. 5.8(a) for different service probabilities. The results are also compared with those for SMAC. Fig. 5.8(b) shows the average delay for data traffic in low priority queue under different rates of service when data arrival rate $\xi = 0.4$ and 0.6. As expected, the average delay decreases when service rate increases. However, increasing the arrival rate incurs more delay for the packets in the queue. For SMAC, the average delay is higher due to non-adaptive duty cycle.

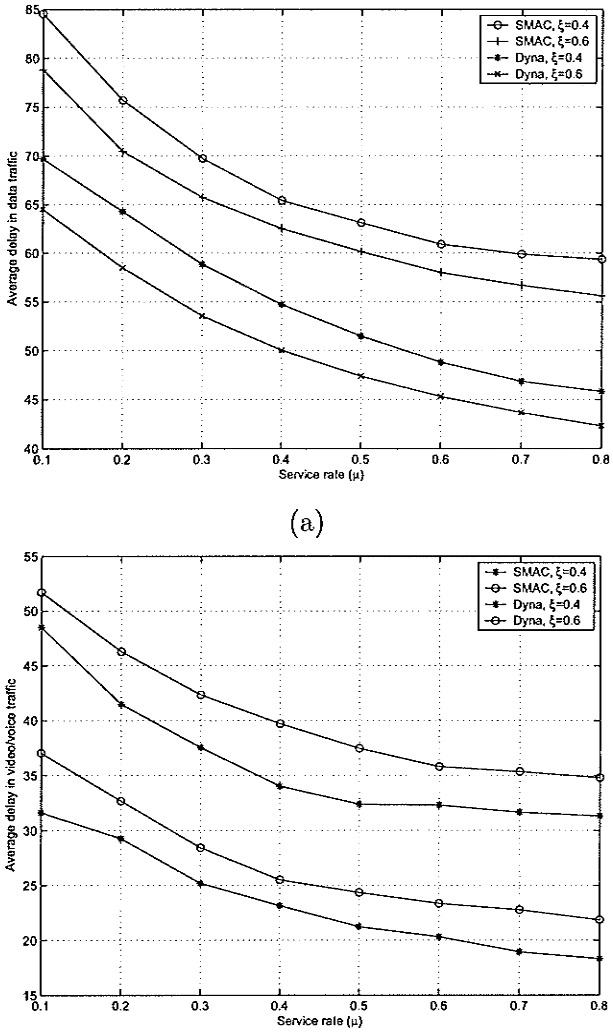
#### 5.5.2.5 Throughput Performance

Throughput in a node is defined as the ratio of the number of packets accepted (and transmitted) by the node and the number of generated packets. There are two cases when a node may fail to transmit the generated packets. First, packet dropping may occur in a node when the queue is full. In such a case, generated packet(s) are dropped fully or partly (in the batch arrival case) due to limited room in the queues. Second, due to the probabilistic dropping mechanism used at each node, a packet could be dropped if the transmission fails due to wireless channel error.

Fig. 5.9(a) shows how throughput increases as the service rate increases for both the multimedia and the data traffic. Throughput for multimedia traffic is higher than that for data traffic. This is due to the higher priority given to this type of traffic at the sensor node. The throughput performance for SMAC with fixed duty cycle is also compared with the throughput of the proposed scheme (Fig. 5.9(a)). As is evident, the fixed and non-adaptive duty cycle of SMAC results lower throughput in a sensor node.

In Fig. 5.9(b) we investigate how changes in data traffic affects the throughput when channel errors are considered. We observe that due to the high service rate in the node, throughput remains almost the same for lower traffic arrival rates. By increasing the arrival rate, it reaches a point where throughput falls. When the channel error rate is high, this threshold occurs earlier. For a lower service rate, the throughput falls at a lower arrival rate.

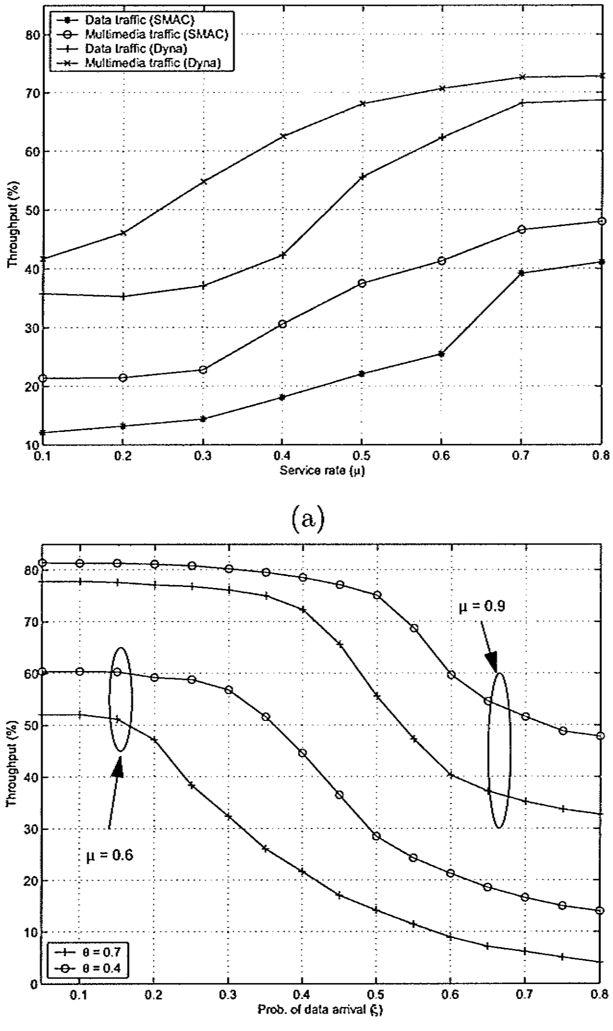
Fig. 5.10 plots throughput variations with retransmission probability $q_r$ for data arrival rate $\xi = 0.25$, service rate $\mu = 0.6$ under different values of channel error



(a)

(b)

**Figure 5.8.** *Average delay under different service probabilities (ξ is the probability of data traffic arrival): (a) data traffic, and (b) video/voice traffic.*



(a)

(b)

**Figure 5.9.** *Variations in throughput (a) for different service rates, and (b) for different data traffic arrival probability, different probability of error $\theta$, and different services probability $\mu$ (under infinite retransmission policy).*

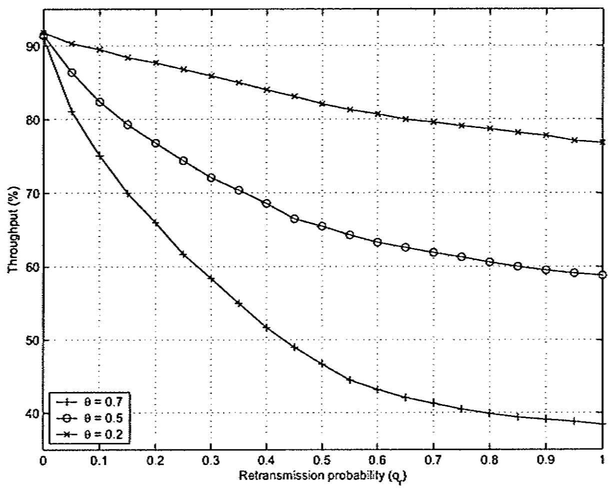


**Figure 5.10.** *Variations in throughput for different values of retransmission probability (θ is the probability of error).*

probability. Increasing the probability of retransmission results in lower throughput for all values of channel error probability in Fig. 5.10 because of higher probability of the queues to be full. However, throughput decreases faster for higher channel errors when retransmission probability increases. Therefore, choosing an optimum retransmission policy may improve the system performance including throughput.

## 5.6 Chapter Summary

We have presented a reinforcement learning-based framework for QoS-aware dynamic power management in distributed wireless multimedia sensor networks. This framework is based on the Dyna architecture which strikes a balance between a complex model-based method and a model-free algorithm such as Q-learning. The proposed framework incurs much less computational complexity than a model-based technique, while at the same time it achieves a higher performance than a model-free algorithm in terms of adaptation and learning.

The current work could be extended to the case that the perfect system model is not available or not observable. Partially observable MDP (POMDP) model might be a good choice to tackle such a case. Again, the exhaustive service discipline in the transmitter makes it limited to sensor applications with low traffic rate. The applicability of the model may be extended to broadband multimedia sensor applications if the exhaustive service discipline is replaced by a more fair service discipline.

# Chapter 6

# Summary and Future Works

## 6.1 Summary of Results

Chapters 2 and 3 present novel queueing analytical frameworks for performance evaluation of a node in a multi-hop wireless network (e.g., sensor network) with distributed and energy-aware medium access control (MAC) protocol for different types of multimedia arrival traffic in the node with different priorities. This framework models two main characteristics of multimedia transmission over sensor networks, namely, power management for energy saving and prioritization or service differentiation among network traffic. This is done by combining priority queueing with a vacation queueing model which is used to model the sleep and wakeup mechanism of the sensor node for saving energy.

Two different power saving mechanisms due to the standard exhaustive and the number-limited exhaustive vacation models (both in multiple vacation cases) are analyzed to study the tradeoff between the QoS performance of the relayed packets and the energy saving at a relay node in a sensor network. We use matrix-geometric method to obtain the stationary probability distribution for the system states from which the performance metrics are derived. Using phase-type distribution for both the service and the vacation processes and combining the priority queueing model with the vacation queueing model make the analysis very general and comprehensive.

The queueing analysis in Chapters 2 and 3 considers non-ideal wireless channel and non-saturation traffic conditions with differentiated services between two types of services. The presented analytical framework is very general and comprehensive in that it considers Markovian arrival process, phase-type distribution for service time and phase-type distribution for vacation period with two types of service disciplines, namely, the standard exhaustive and the number-limited exhaustive both in multiple vacation. The trade-off between the QoS performances for the high-priority packets (i.e., queue length distribution, distribution of access delay, packet dropping probability and throughput) and the energy saving performance at a node has been analyzed under different system parameter settings.

The queueing frameworks in Chapters 2 and 3 are exploited in Chapters 4 and 5 for dynamic power management in wireless multimedia sensor networks. In Chapter 4, a dynamic power management framework is developed based on a Markov decision process (MDP) for a wireless video sensor node to improve the energy saving performance while satisfying the video transmission quality requirements. Dynamic programming approach is used in this chapter to solve the MDP problem considering the video traffic arrival process in the sensor node, the sleep and wakeup processes in the camera and wireless transceiver electronics, the queue status, and the multi-rate wireless channel condition. Dynamic programming approach is used to find the optimum policy to achieve the desired performance measures in an energy-limited sensor node. Channel dynamics and multi-rate transmission in the node, independent sleep processes in the transmitter and camera in the node, and construction a MAP process based on MPEG video characteristics are among the contributions of the work in this chapter.

To overcome the curse of dimensionality in dynamic programming approach, a reinforcement learning-based distributed dynamic power management framework is presented in Chapter 5. The queueing model in this case captures bursty nature of the multimedia (e.g, video/voice, data) traffic arrival process and prioritization between video/voice and data traffic. A model-based reinforcement learning approach is used to solve the MDP problem in which the structure of the transition probability matrix in the MDP formulation is exploited to reduce the convergence time in the learning process. The framework is based on the Dyna architecture which strikes a balance between a complex model-based method and a model-free algorithm such as Q-learning. The proposed framework incurs much less computational complexity than a model-based technique, while at the same time it achieves a higher performance than a model-free algorithm in terms of adaptation and learning.

## 6.2 Future Works

Several possible future research directions are outlined below.

- Different sleep management and energy efficient mechanisms can be applied to our model. For instance, in our current work, there is no *return policy* for the node in the sleep state. A return policy was introduced in [98] (also known as *NT-policy*) by assigning a threshold on the time and the number of the packets in the queue which forces the node to wakeup from its sleep state. Incorporating the NT-policy in our study, along with channel state and energy state information of the node in the analysis would be interesting. Again, energy consumption in a node could be further analyzed considering other node states such as *listening, idle, standby.* An optimum return policy for a multi-state sleep mechanism is another potential extension of this work.
- The current work could be extended to the case that the perfect system model is not available or not observable. Partially observable MDP (POMDP) model might be a good choice to tackle such a case. Again, the exhaustive service discipline in the transmitter makes it limited to sensor applications with low traffic rate. The applicability of the model may be extended to broadband multimedia sensor applications if the exhaustive service discipline is replaced by a more fair service discipline.
- The effect of the battery has been represented by the discount factor in the analysis in Chapters 4 and 5. This assumption leads us to the Markov stationary policies where off-line calculations are enough for decision making. However, the optimum decision making must be taken dynamically when the remaining battery power in a sensor node changes.
- The analytical models presented here for a single sensor node can be extended for performance analysis and optimization in a multi-hop sensor network on an end-to-end basis. One important issue here is the heterogeneity among the nodes. Another issue is the idea of *cooperative reinforcement learning* in which reinforcement learning is applied to a network of nodes to optimize the network performance cooperatively.
- It would be also interesting to use the general performance analysis and opti-

mization models developed in this work to a practical wireless sensor network built from the off-the-shelf wireless sensor nodes (e.g., Bluetooth or ZigBee-based wireless multimedia sensor network).

# Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, March 2002.

[2] C. -Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247-1256, August 2003.

[3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921-960, March 2007.

[4] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *submitted to IEEE Communications Surveys and Tutorials* (available at http://www.public.asu.edu/ ssatyaja/).

[5] V. Raghunathan, S. Ganeriwal, and M. Srivastava, "Emerging techniques for long lived wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 108-114, April 2006.

[6] Z. He and D. Wu, "Resource allocation and performance analysis of wireless video sensors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 5, pp. 590-599, May 2006.

[7] A. Sinha and A. P. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers Magazine*, vol. 18, no. 2, pp. 62-74, March-April 2001.

[8] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493-506, June 2004.

[9] M. J. Miller and N. H. Vaidya, "A MAC protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228-242, May/June 2005.

[10] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: A survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115-121, April 2006.

[11] K. Kredo and P. Mohapatra,"Medium access control in wireless sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 961-994, March 2007.

[12] P. J. M. Havinga and G. J. M. Smit, "Energy-efficient wireless networking for

multimedia applications," *Wireless Communununications and Mobile Computing*, Wiley, vol. 1, no. 2, pp. 165-184, 2001.

[13] E. Gurses and O. B. Akan, "Multimedia communication in wireless sensor networks," *Annals of Telecommunications*, vol. 60, no. 7-8, pp. 799-827, July-August 2005.

[14] A. Chandra, V. Gummalla, and J. Limb, "Wireless medium access control protocols," *IEEE Communications Surveys and Tutorials*, vol. 3, no. 2, pp. 2-15, 2nd Quarter 2000.

[15] R. Jurdak,C. Lopes, and P. Baldi, "A survey, classification and comparative analysis of medium access control protocols for ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 6, no. 1, pp. 2-16, 1st Quarter 2004.

[16] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70-80, January-March 2002.

[17] R. Zheng, J. C. Hou, and L. Sha, "Performance analysis of power management policies in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1351-1361, June 2006.

[18] X. Yang and N. H. Vaidya, "A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay," *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, May 2004.

[19] F. J. Block and C. W. Baum, "Energy-efficient self-organizing communication protocols for wireless sensor networks," *Proc. IEEE Military Communications Conference 2001 (MILCOM'01)*, October 2001.

[20] C. -F. Chiasserini and M. Garetto, "Modeling the performance of wireless sensor networks," *Proc. IEEE INFOCOM'04*, March 2004.

[21] S. Singh and C. S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5-26, July 1998.

[22] F. Zhang, T. D. Todd, D. Zhao, and V. Kezys, "Power saving access points for IEEE 802.11 wireless network infrastructure," *IEEE Transactions on Mobile Computing*, vol. 5, no. 2, pp. 144-156, February 2006.

[23] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol.7 no.4, pp. 343-358, August 2001.

[24] Q. Gao, K. J. Blow, D. J. Holding, and I. Marshall, "Analysis of energy conservation in sensor networks," *Wireless Networks*, vol. 11, no. 6, pp. 787-794, November 2005.

[25] J. Schiller, "Mobile Communications," Addison-Wesley, 2000.

[26] "The Bluetooth special interest group," 1999.

[27] A. F. Mini, B. Nath, and A. A. F. Loureiro, "A probabilistic approach to predict the energy consumption in wireless sensor networks," *Proc. 4th Workshop de Comunicao sem Fio e Computao Mvel*, Sao Paulo, Brazil, October 2002.

[28] J. Misic and V. B. Misic, "Duty cycle management in sensor networks based on 802.15.4 beacon enabled MAC," *Ad Hoc and Sensor Wireless Networks Journal*, Old City Publishing, vol. 1, no. 1, pp. 207-233, March 2005.

[29] A. S. Alfa, "Matrix-geometric solution of discrete time MAP/PH/1 priority queue," *Naval Research Logistics* vol. 45, no. 1, pp. 23-50, 1998.

[30] A. S. Alfa, "Vacation models in discrete time," *Queueing Systems*, vol. 45, no. 4, pp. 5-30, December 2003.

[31] N. Akar, N. C. Oguz, and K. Sohraby, "Matrix-geometric solutions of M/G/1-type Markov chains: A unifying generalized state-space approach," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 626-639, June 1998.

[32] R. Mukhtar, S. Hanly, M. Zukherman, and F. Cameron, "A model for the performance evaluation of packet transmissions using type-II hybrid ARQ over a correlated error channel," *Wireless Networks*, vol. 10, no. 1, pp. 7-16, January 2004.

[33] U. Vornefeld, "Analytical performance evaluation of mobile Internet access via GPRS networks," *Proc. European Wireless 2002*, Italy, February 2002.

[34] F. A. Tobagi,"Carrier sense multiple access with message-based priority functions," *IEEE Transactions on Communications*, vol. 30, no. 1, pp. 185-200, January 1982.

[35] G. L. Choudhury and S. S. Rappaport,"Priority access schemes using CSMA-CD," *IEEE Trans. Comm.*, vol. 33, no. 7, pp. 620-626, July 1985.

[36] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, March 2000.

[37] V. Bhargahavan, "Performance evaluation of algorithms for wireless medium access," in *Proc. IEEE Performance and Dependability Symposium*, 1998, USA.

[38] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: Throughput and delay analysis," *Computer Communications*, vol. 25, no. 3, pp. 313-321, February 2002.

[39] C. H. Foh and M. Zukerman, "Performance analysis of the IEEE 802.11 MAC protocol," in *Proc. of European Wireless'2002.*

[40] E. M. M. Winands, T. J. J. Denteneer, J. A. C. Resing, and R. Rietman, "A finite-source feedback queueing network as a model for the IEEE 802.11 distributed coordination function," in *Proc. of European Wireless'2004*.

[41] O. Tickoo and B. Sikdar, "A queueing model for finite load IEEE 802.11 random access MAC," *Proc. IEEE International Conference on Communications (ICC)*, 2004.

[42] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123-131, March 2005.

[43] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: A survey," *Proc. International Conference on Wireless Networks (ICWN 2004)*, USA, June 2004.

[44] M. Younis, K. Akayya, M. Eltowiessy, and A. Wadaa, "On handling QoS traffic in wireless sensor networks," *Proc. 37th Hawaii International Conference on System Sciences (HICSS-37)*, Hawaii, January 2004.

[45] M. Barry, A. T. Campbell, and A. Veres, "Distributed control algorithms for service differentiation in wireless packet networks," in *Proc. of IEEE INFOCOM'2001*, pp. 582-590, April 2001.

[46] L. Rhomdani, N. Qiang, and T. Turletti, "AEDCF: Enhanced service differentiation for IEEE 802.11 wireless ad hoc networks," *INRIA Technical Report RR4544*, September 2002.

[47] C.-H. Yeh, "QoS differentiation mechanisms for heterogeneous wireless networks and the next-generation Internet," in *Proc.* $8^{th}$ *IEEE International Symposium on Computers and Communication (ISCC'2003)*.

[48] J. Yu, S. Choi, and J. Lee, "Enhancement of VoIP over IEEE 802.11 WLAN via dual queue strategy," in *Proc. of IEEE International Conference on Communications (ICC'04)*, France, June 2004.

[49] Q. Liu, S.Zhou, and G. B. Giannakis, "Queuing with Adaptive modulation and coding over wireless link: Cross-layer analysis and design," *IEEE Trans. on Wireless Communication*, vol. 4, no. 3, pp. 1142-1153, May 2005.

[50] F. Babich and G. Lombardi, "A Markov model for the mobile propagation channel," *IEEE Trans. Veh. Technol.*, vol. 49, no. 1, pp. 63-73, January 2000.

[51] J. Yun and M. Kavehrad, "Markov error structure for throughput analysis of adaptive modulation systems combined with ARQ over correlated fading channels," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 1, pp. 235-245, January 2005.

[52] L. B. Le and E. Hossain, "Delay statistics for selective repeat ARQ protocol

in multi-rate wireless networks with non-instantanous feedback," *IEEE GLOBECOM'05*, 2005.

[53] A. Fallahi and E. Hossain, "Distributed and energy-aware MAC for differentiated services wireless packet networks: A general queuing analytical framework," *IEEE Transactions on Mobile Computing*, vol. 6, no. 4, pp. 381-394, April 2007.

[54] X. Wang, Q. Liu, and G. B. Giannakis, "Analyzing and optimizing adaptive modulation coding jointly with ARQ for QoS-guaranteed traffic," *IEEE Trans. on Vehicular Technology*, vol. 56, no. 2, pp. 710-720, March 2007.

[55] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models*, John Hopkins University Press, Baltimore, MD, 1981.

[56] M. F. Neuts, "A versatile Markovian point process," *Journal of Applied Probability*, vol. 16, pp. 764-779, 1979.

[57] B. Van Houdt, R. B. Lenin, and C. Blondia, "Delay distribution of (im)patient customers in a discrete time D-MAP/PH/1 queue with age-dependent service times," *Queueing Systems*, vol. 45, no. 1, pp. 59-73, September 2003.

[58] D. M. Lucantoni, "New results on the single server queue with a batch Markovian arrival process," *Stochastic Models*, vol. 7, no. 1, pp. 1-46, 1991.

[59] C. Blondia and O. Casals, "Statistical multiplexing of VBR sources: A matrix-analytic approach," *Performance Evaluation*, vol. 16, no. 1-3, pp. 5-20, 1992.

[60] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[61] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, 1996.

[62] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Computation*, vol. 6, no. 6, pp. 1185-1201, 1994.

[63] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279-292, 1992.

[64] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *Proc. of 2001 ACM MOBICOM*, Rome, Italy, pp. 272-286, July 2001.

[65] V. Ramaswami, "A stable recursion for the steady state vector in Markov chains of M/G/1 type," *Stochastic Models*, vol. 4 pp. 183-263, 1988.

[66] G. Cantieni, Q. Ni, C. Barakat, and T. Turletti, "Performance analysis under finite load and improvements for multirate 802.11," *Computer Communications* Journal (Elsevier), vol. 28, no. 10, pp. 1095-1109, June 2005.

[67] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.

[68] *IEEE 802.11 - wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE inc. Std., 1999.

[69] A. Horvath and M. Telek, "Phfit: A general phase-type fitting tool," in *Proc. of Performance TOOLS'2002*, London, UK, April 2002.

[70] C. Xiaoming and H. Geok-Soon, "A simulation study of the predictive p-persistent CSMA protocol," in *Proc. IEEE 35th Annual Simulation Symposium*, 2002.

[71] S. Rakshit and R K. Guha,"Fair bandwidth sharing in distributed systems: A game-theoretic approach," *IEEE Trans. Computers*, vol. 54, no. 11, pp. 1384-1393, November 2005.

[72] R. Hooke and T. A. Jeeves,"Direct search solution of numerical and statistical problems," *Journal of the ACM*, vol. 8, pp. 212-229, April 1961.

[73] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*, 2nd Edition, John Wiley and Sons, 2001.

[74] D. P. De Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Operations Research*, vol. 51, no. 6, pp. 850-856, November 2003.

[75] L. B. White, "A new policy evaluation algorithm for Markov decision processes with quasi birth-death structure," *Stochastic Models*, vol. 21, no. 2-3, pp. 785-797, January 2005.

[76] F. J. Vazquez-Abad and V. Krishnamurthy, "Self learning control of constrained Markov decision processes, a gradient approach," *Les Cahiers du GERAD* G-2003-51, August 2003.

[77] P. Bellman and R. Kalaba, *Dynamic Programming and Modern Control Theory*, Academic Press, 1965.

[78] B. Ata and K. E. Zachariadis, "Dynamic power control in a fading downlink channel subject to an energy constraint," *Queueing Systems* vol. 55, no. 1, pp. 41-69, January 2007.

[79] E. Altman, *Constrained Markov Decision Processes*, Chapman Hall/CRC, 1999.

[80] R. Berry and R. Gallager, "Communication over fading channels with delay constraints," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 11351149, May 2002.

[81] X. Hong, K. Sohraby, and R. Grossman, "A dynamic programming approach for optimal scheduling policy in wireless networks," in *Proc. of International Conference on Computer Communications and Networks (ICCCN)*, 2002.

[82] T. Holliday and A. Goldsmith, and P. Glynn "Wireless link adaptation policies: QoS for deadline constrained traffic with imperfect channel estimates," in *Proc. of International Conference on Communications (ICC)*, 2002.

[83] J. Fuemmeler and V. V. Veeravalli. "Smart sleeping policies for energy efficient tracking in sensor networks," submitted to the *IEEE Transactions on Signal Processing.*

[84] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proc. of IEEE INFOCOM*, March 2004.

[85] L. Benini, A. Bogliolo, G. Paleologo, and G. De Micheli, "Policy optimization for dynamic power management," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 6, pp. 813-833, June 1999.

[86] S. J. Yoo and S. D. Kim, "Traffic modeling and QoS prediction for MPEG-coded video services over ATM networks using scene level statistical characteristics," *Journal of High Speed Networks*, vol. 8, no. 3, pp. 211-224, 1999.

[87] D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcasting video traffic," *IEEE/ACM Transactions on Networking*, vol. 4, no. 1, pp. 40-48, February 1996.

[88] E. A. Feinberg and A. Shwartz, *Handbook of Markov Decision Processes: Methods and Applications*, Kluwer, 2002.

[89] Z. Liu and I. Elhanany, "RL-MAC: A reinforcement learning based MAC protocol for wireless sensor networks," *International Journal of Sensor Networks (IJSNET)*, vol. 1, no. 3/4, pp. 117-124, April 2006

[90] C. Pandana and K. J. R. Liu, "Near-optimal reinforcement learning framework for energy-aware sensor communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 788-797, April 2005.

[91] Y. Fei, V. W. S. Wong, and V. C. M. Leung,"Efficient QoS provisioning for adaptive multimedia in mobile communication networks by reinforcement learning," *Mobile Networks and Applications*, vol. 11, no. 1, pp. 101-110, February 2006.

[92] D. V. Djonin and V. Krishnamurthy, "Q-learning algorithms for constrained Markov decision processes with randomized monotone policies: Application to MIMO transmission control," *IEEE Transactions Signal Processing*, vol. 55, no. 5, pp. 2170-2181, 2007.

[93] J. Zhao, C. W. Kok, and I. Ahmad, "MPEG-4 video transmission over wireless networks: A link level performance study," *Wireless Networks*, vol. 10, no. 2, pp. 133-146, March 2004.

[94] I. Frigui, A. S. Alfa, and X. Y. Xu, "Algorithms for computing waiting time

distributions under different queue disciplines for the DBMAP/PH/1," *Naval Research Logistics*, vol. 44, no. 6, pp. 559-576, 1997.

[95] J. A. Zhao, B. Li, X. R. Cao, and I. Ahmad,"A matrix-analytic solution for the DBMAP/PH/1 priority queue," *Queueing Systems*, vol. 53, no. 3, pp. 127-145, July 2006.

[96] J. G. Kim and M. M. Krunz, "Delay analysis of selective repeat ARQ for a Markovian source over wireless channel," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 5, pp. 1968-1981, September 2000.

[97] J. Xue and A. S. Alfa, "Tail probability of low-priority queue length in a discrete-time priority BMAP/PH/1 queue," *Stochastic Models*, vol. 21, no. 2-3, pp. 799-820, January 2005.

[98] A. S. Alfa and I. Frigui,"Discrete NT-policy single server queue with Markovian arrival process and phase type service," *European Journal of Operational Research*, vol. 88, no. 3, pp. 599-613, 1996.

# List of Acronyms

| | |
|---|---|
| ACK | acknowledgement, 69 |
| AMC | adaptive modulation and coding, 12, 15, 112, 116 |
| ARQ | automatic repeat request, 7, 12, 15, 69, 70, 112, 118, 139 |
| BMAP | batch Markovian arrival process, 19, 143, 145, 147, 149, 157 |
| CMOS | complementary metal oxide semiconductor, 1 |
| CSMA | carrier sense multiple access, 10, 95, 97, 106 |
| CSMA-CD | carrier sense multiple access with collision detection, 10 |
| CW | contention window, 48 |
| D-MAP | discrete-time Markovian arrival process, 18, 19, 26, 120, 145, 146 |
| DBMAP | discrete-time batch Markovian arrival process, 145 |
| DCF | distributed coordination function, 11, 12, 48, 49, 51–53, 64, 69, 96, 97 |
| DIFS | distributed inter-frame space, 49 |
| DPM | dynamic power management, 3, 114, 137–139, 145 |
| EDCF | enhanced distributed coordination function, 12 |
| FEC | forward error correction, 7 |

| | |
|---|---|
| FSMC | finite state Markov channel, 12, 15, 112, 116, 118, 130, 134, 135 |
| GoP | group of pictures, 116, 134 |
| GPRS | general packet radio service, 10 |
| HARQ | hybrid automatic repeat request, 10 |
| HOL | head-of-line, 15, 27, 69, 70, 79, 84 |
| IRM | Independent Replications Method, 53 |
| MAC | medium access control, 3–6, 8–15, 23–27, 48, 49, 51–53, 64, 68–70, 97, 111, 112, 118, 120, 143, 144, 148 |
| MAP | Markovian arrival process, 15, 18, 19, 24–26, 51, 68, 96, 112, 120, 143, 145 |
| MDP | Markov decision process, 13–15, 20, 21, 112, 113, 123, 143, 144, 155, 158, 169, 173 |
| MIMO | multiple-input multiple-output, 144 |
| MMAP | marked Markovian arrival process, 10 |
| MMBP | Markov modulated Bernoulli process, 18 |
| MPEG | moving picture experts group, 15, 112, 115, 116, 118, 139, 145 |
| NL | number-limited, 28 |
| NRT | non real time, 12 |
| pdf | probability distribution function, 11 |
| PH | phase-type, 24, 26, 49, 68, 69, 96, 145 |
| PHY | physical, 24–27, 49 |
| POMDP | partially observable Markov decision process, 169, 173 |

| | |
|---|---|
| QAM | quadrature amplitude modulation, 116 |
| QBD | quasi-birth-death, 17, 33, 34, 123, 125, 129, 130 |
| QoS | quality of service, 2–5, 7, 11–15, 24, 49, 50, 58, 64, 101, 106, 109, 111–113, 132, 133, 135, 139, 143, 144, 159, 169 |
| RF | radio frequency, 1 |
| RL | reinforcement learning, 144, 156 |
| RT | real time, 12 |
| S-ALOHA | Slotted ALOHA, 48, 49, 58, 64 |
| SNR | signal-to-noise ratio, 116, 135, 137 |
| SPM | static power management, 123, 134, 137, 138 |
| STA | station, 48, 49 |
| TDMA | time division multiple access, 6 |
| VoIP | voice over IP, 12 |
| WLAN | wireless local area network, 12 |

# Publications

## *Journal Publications*

1. A. Fallahi and E. Hossain, "QoS Provisioning in Wireless Video Sensor Networks: A Dynamic Power Management Framework," accepted for publication in *IEEE Wireless Communications Magazine*, special issue on Wireless Sensor Networking, December 2007.
2. A. Fallahi and E. Hossain,"Distributed and Energy-Aware MAC for Differentiated Services Wireless Packet Networks: A General Queueing Analytical Framework," *IEEE Trans. Mobile Computing*, vol. 6, no. 4, pp. 381-394, April 2007.
3. D. Niyato, E. Hossain, and A. Fallahi,"Sleep and Wakeup Strategies in Solar-Powered Wireless Sensor/Mesh Networks: Performance Analysis and Optimization," *IEEE Trans. Mobile Computing*, vol. 6, no. 2, pp. 221-236, February 2007.
4. A. Fallahi, E. Hossain, and A. S. Alfa, "QoS and Energy Tradeoff in Distributed Energy-Limited Mesh/Relay Networks: A Queueing Analysis," *IEEE Trans. Parallel and Distributed Systems*, vol. 17, no. 6, pp. 576-592, June 2006.

## *Conference Publications*

1. A. Fallahi and E. Hossain, "Queueing Analysis of Distributed MAC in Wireless Ad Hoc Networks with Differentiated Services," *IEEE International Conference on Communications 2006 (ICC'06)*, Istanbul, Turkey.
2. D. Niyato, E. Hossain, and A. Fallahi, "Analysis of Different Sleep and Wakeup Strategies in Solar Powered Wireless Sensor Networks," *IEEE International Conference on Communications 2006 (ICC'06)*, Istanbul, Turkey.
3. D. Niyato, E. Hossain, and A. Fallahi, "Solar-Powered OFDM Wireless Mesh Networks with Sleep Management and Connection Admission Control," *International Wireless Communications and Mobile Computing Conference (IWCMC'06)*, Vancouver, Canada, July 2006.

4. A. Fallahi and E. Hossain, "GeRaF-H: Geographic Random Forwarding in Wireless Ad Hoc and Sensor Networks with Heterogeneous Power Nodes," *International Conference on Next-Generation Wireless Systems 2006 (ICNEWS'06)*, Jan. 2006.
5. A. Fallahi, M. Kahrizi, and S. Valaee, "GSM Receiver Simplification Using Continuous Phase Modulations Linear Approximation", The 9th Iranian Conference on Electrical Engineering (ICEE), Proceedings of Communication Theory and Systems, pp. 2.1-2.8, Power & Water Inst. of Tech., Tehran, Iran, May 2001.
6. A. Fallahi, S. Valaee, and M. Kahrizi,"A Hybrid Algorithm for Viterbi Equalizer Complexity Reduction in GSM," The 7th ICEE, Proceedings of Communication Theory and Systems, pp.9-16, Iran Telecom. Research Center, Tehran, Iran, May 1999.

***Work Submitted for Publication***

1. A. Fallahi and E. Hossain, "A Dynamic Programming Approach for QoS-Aware Power Management in Wireless Video Sensor Networks," submitted to *IEEE Transactions on Vehicular Technology.*