ENERGY-DELAY AND ENERGY-AGE TRADE-OFF IN SLEEP-WAKE SCHEDULING OF QUEUING SYSTEMS

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Ömer Gürsoy
September 2025

Energy-Delay and Energy-Age Trade-Off in Sleep-Wake Scheduling of Queuing Systems

By Ömer Gürsoy

September 2025

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Nail Akar(Advisor)

Ezhan Karaşan

Mehmet Akif Yazıcı

Klaus Werner Schmidt

Muhammed Ömer Sayın

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

ENERGY-DELAY AND ENERGY-AGE TRADE-OFF IN SLEEP-WAKE SCHEDULING OF QUEUING SYSTEMS

Ömer Gürsoy
Ph.D. in Electrical and Electronics Engineering
Advisor: Nail Akar
September 2025

Minimizing energy consumption while satisfying Quality of Service (QoS) requirements has become a critical objective for the efficient and successful deployment of next-generation networks (NGNs). Sleep-wake scheduling stands out as a prominent approach among various mechanisms that aim to enhance energy efficiency. This mechanism allows the node to save energy by temporarily turning off the components of its transmission module, albeit at the cost of potential QoS degradation. Numerous sleep-wake scheduling strategies have been proposed in both academic and industrial contexts to manage the transition between Sleep and Active operational modes. Most of the existing methods focus on a narrow set of QoS metrics while employing conventional algorithmic approaches. As communication systems grow in complexity and diversity, there is an increasing need for more advanced mechanisms that can accommodate a broader range of QoS considerations. In parallel, the advances in computational capabilities support the integration of sophisticated control systems, such as model predictive control (MPC), which offer greater adaptability to dynamic environments.

Motivated by these developments, this thesis explores the application of innovative sleep-wake scheduling strategies in communication systems. In particular, we consider (i) delay (ii) information freshness, as the QoS metrics for the analysis and control of sleep-wake scheduling in two separate settings. In the first setting, we introduce a novel open-loop dynamic coalescing method for Energy Efficient Ethernet (EEE), which leverages queuing theory and is inspired by MPC. Our first proposed method, referred to as MPC-mean, aims to reduce the energy consumption of Ethernet links while maintaining constraints on the average queue waiting time. Additionally, we propose another MPC inspired approach, called MPC-tail, which focuses on controlling the tail distribution of queue waiting times. Beyond Ethernet interfaces, we investigate information freshness in Internet of Things (IoT) sensor networks, where nodes alternate between Sleep (or Low-power) and Active modes. We analyze the trade-off between energy consumption and the Age of Incorrect Information (AoII) using timer-based sleep-wake scheduling. To validate the performance and efficiency of our proposed techniques, we provide extensive numerical simulations.

Keywords: Energy efficiency, Sleep-wake scheduling, Information freshness, Age of Incorrect Information, Energy efficient Ethernet, Model predictive control, Queuing theory, Phase-type distribution, Matrix exponential distribution.

ÖZET

KUYRUK SİSTEMLERİNDE UYKU-UYANIKLIK ZAMANLAMASINDA ENERJİ-GECİKME VE ENERJİ-BİLGİ YAŞI DENGELEMESİ

Ömer Gürsoy
Elektrik ve Elektronik Mühendisliği, Doktora
Tez Danışmanı: Nail Akar
Eylül 2025

Enerji tüketimini en aza indirirken Hizmet Kalitesi (QoS) gereksinimlerini karşılamak, yeni nesil ağların (NGN'ler) verimli ve başarılı bir şekilde devreye alınması için kritik bir hedef haline gelmiştir. Enerji verimliliğini artırmayı amaçlayan çeşitli mekanizmalar arasında, uyku-uyanıklık zamanlaması öne çıkan bir yaklaşım olarak dikkat çekmektedir. Bu mekanizma, düğümün iletim modülünün bazı bileşenlerini geçici olarak kapatarak enerji tasarrufu sağlamasına olanak tanır; ancak bu, potansiyel QoS bozulması pahasına olabilir. Uyku ve Aktif çalışma modları arasındaki geçişi yönetmek için hem akademik hem de endüstriyel bağlamlarda çok sayıda uyku-uyanıklık zamanlama stratejisi önerilmiştir. Mevcut yöntemlerin çoğu, yalnızca dar bir QoS ölçütü kümesine odaklanmakta ve geleneksel algoritmik yaklaşımları kullanmaktadır. Ancak iletişim sistemleri karmaşıklaştıkça ve çeşitlendikçe, daha geniş bir QoS yelpazesini karşılayabilecek gelişmiş mekanizmalara olan ihtiyaç artmaktadır. Buna paralel olarak, hesaplama yeteneklerindeki ilerlemeler, model öngörülü kontrol (MPC) gibi dinamik ortamlara daha iyi uyum sağlayabilen gelişmiş kontrol sistemlerinin entegrasyonunu mümkün kılmaktadır.

Bu gelişmelerden ilhamla, bu tez iletişim sistemlerinde yenilikçi uyku-uyanıklık zamanlama stratejilerinin uygulanmasını araştırmaktadır. Özellikle, uyku-uyanıklık zamanlamasının analiz ve kontrolü için QoS ölçütleri olarak bilgi güncelliği ve gecikmeyi ele alıyoruz. İlk olarak, MPC ve kuyruk teorisinden yararlanan, Enerji Verimli Ethernet (EEE) için yeni bir açık döngü dinamik birleştirme yöntemi sunuyoruz. "MPC-mean" olarak adlandırılan ilk önerdiğimiz yöntem, ortalama kuyruk bekleme süresi üzerindeki kısıtları koruyarak Ethernet bağlantılarının enerji tüketimini azaltmayı amaçlamaktadır. Buna ek olarak, kuyruk bekleme sürelerinin kuyruk dağılımını yönetmeye odaklanan "MPC-tail" adında başka bir MPC tabanlı yaklaşım daha öneriyoruz. Ethernet arayüzlerinin ötesinde, düğümlerin Uyku (veya Düşük Güç) ve Aktif modlar arasında geçiş yaptığı Nesnelerin İnterneti (IoT) sensör ağlarında bilgi güncelliğini de inceliyoruz. Zamanlayıcı tabanlı uyku-uyanıklık zamanlaması kullanılarak, enerji tüketimi ile hatalı bilginin yaşı arasındaki denge analiz edilmektedir. Önerilen tekniklerin performansını ve verimliliğini doğrulamak amacıyla kapsamlı sayısal simülasyonlar sunulmaktadır.

Anahtar sözcükler: Enerji verimliliği, Uyku-uyanıklık zamanlaması, Bilgi güncelliği, Hatalı bilginin yaşı, Enerji verimli Ethernet, Model öngörülü kontrol, Kuyruk teorisi, Faz türü dağılım, Matris üstel dağılım.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Nail Akar for his endless support, encouragement, and patience during the past six years. I greatly appreciate his comments and ideas from which I have learned a lot. I have been extremely fortunate to have such an encouraging and caring supervisor who has become more of a mentor to me in the meantime.

I am indebted to my Ph.D. committee members, Prof. Ezhan Karaşan and Asst. Prof. Mehmet Akif Yazıcı, who provided me with their precious suggestions during the past four years. I also would like to thank Prof. Klaus Werner Schmidt and Asst. Prof. Muhammed Ömer Sayın for accepting to serve as jury members in my Ph.D. thesis defense.

Last, but not least, I would like to express my eternal gratitude to my family for their everlasting love and support.

Contents

List of Figures

List of Tables

Abbreviations

NGN	next generation network
QoS	quality of service
MPC	model predictive control
EEE	energy efficient ethernet
BS	base station
LEO	low earth orbit
IoT	internet of things
UAV	unmanned aerial vehicle
V2X	vehicle-to-everything
ICT	information and communications technology
AoI	age of information
PAoI	peak age of information
AoII	age of incorrect information
AoS	age of synchronization
Ao^{2}I	age of outdated information

EE-PAoI	energy efficiency-PAoI
AEE	age-energy efficiency
LPI	low power idle
PH	phase
ME	matrix exponential
CME	concentrated matrix exponential
PID	proportional integral derivative
CDF	cumulative distribution function
PDF	probability density function
SCV	squared coefficient of variation
LCFS	last come first served
FCFS	first come first serve
WNCS	wireless networked control system
IoE	internet of everything
DRX	discontinuous reception
LTE	long term evolution
WSN	wireless sensor network
3GPP	third generation partnership project

Chapter 1

Introduction

Energy efficiency is a topic of paramount importance for next generation networks (NGNs). The rapid growth in applications such as tactile Internet, smart cities, unmanned aerial vehicle (UAV) communication, vehicle-to-everything (V2X) systems, and Internet of Things (IoT) networks leads to a broader list of performance expectations with more stringent requirements. These include higher data throughput, ultra-low latency, real-time information updates, dense device connectivity, enhanced positioning accuracy, global coverage, and robust, secure communication links [2, 3]. These demands inevitably increase the energy requirements of communication networks. As a result, the information and communications technology (ICT) sector's share of global energy consumption has significantly grown over the past decade, along with its growing relevance to the overall economy [4]. Consequently, reducing the energy consumption while maintaining Quality of Service (QoS) standards has become a key requirement for the successful and efficient deployment of emerging communication services.

Sleep-wake scheduling is one of the prominent mechanisms for energy efficiency which allows the node to save energy by temporarily turning off the components of its transmission module, e.g., physical and medium access control layer components, when there are no packets to transmit. A sleep-wake strategy usually consists of two components: i) an activation scheme that determines when the server should wake up, and ii) an idling (or sleeping) scheme that determines when the server should be put to sleep. These strategies are widely applicable across communication systems and have been standardized in technologies like Discontinuous Reception (DRX) in Long Term Evolution (LTE) [5] and energy efficient Ethernet (EEE) [6]. Sleep-wake mechanisms are particularly vital in NGNs like IoT systems, where nodes often function in power-constrained or remote environments e.g. Wireless Sensor Network (WSN) applications in which IoT nodes operate in isolated environments with limited supply of power, and the sustainability of such networks relies on energy-saving capabilities of such nodes.

The majority of the standardization efforts for sleep-wake scheduling do not enforce the use of a certain mechanism governing the transitions between the Sleep and Active modes of operation for which there is a need to develop effective strategies while trading off energy efficiency and QoS. Various activation and sleeping schemes have been proposed using the tools of queuing theory; see for example [7]. Selection of appropriate strategies to trigger the transitions between Sleep and Active modes directly impacts the complexity of the implementation and the QoS performance of sleep-wake scheduling.

Most existing research employ delay as the primary QoS metric, studying the trade-off between energy consumption and delay [8], [1], [9]. More recently, there has been growing interest in alternative QoS metrics that capture the freshness of information, particularly through the concept of Age of Information (AoI) [7], [10], [11], [12]. Despite this progress, diversity of QoS metrics addressed in the current literature remains insufficient for the varying demands of current applications. Furthermore, most existing scheduling mechanisms are developed under restrictive assumptions about traffic patterns, such as fixed distributions or static traffic models. While these assumptions simplify control design, they can result in performance degradation under real-world network dynamics.

In the meantime, aligned with the rise of computational power, the popularity of high-complexity control mechanisms has increased in recent years with novel application areas including green communications and networks [13,14]. Complex control systems such as model predictive control (MPC) have expanded into new application areas, offering higher adaptability to dynamic environments. Moreover, current advances have paved the way for integrating more complex QoS metrics, even if they require more complex control strategies and derivations.

Investigation of complex control mechanism and novel QoS metrics for sleep-wake scheduling has the potential to offer more sustainable and energy efficient NGNs. To contribute towards our understanding of how this potential may be exploited, this thesis explores the energy-age and energy-delay trade-off in sleep-wake scheduling of queuing systems proposing novel control mechanisms to address the ever-increasing demands for power consumption and performance in next generation communication networks.

1.1 Contributions of the Thesis

Energy efficiency solutions in the design of NGN technologies is considered as one of the key enablers to meet the increasing demands [4]. With this motivation, we consider various applications of novel sleep-wake scheduling mechanisms with a focus on delay and age metrics for communication systems within the scope of this thesis. In particular, we propose a novel open-loop dynamic coalescing technique for EEE that is based on MPC and queuing theory. For this purpose, we first propose the MPC-mean method which attempts to minimize the energy consumption of the Ethernet link, under constraints on mean of the queue waiting time. Additionally, we propose another MPC based coalescer, termed as MPC-tail, which operates under constraints on the tail of the queue waiting time. Furthermore, we investigate information freshness for IoT sensor nodes which can alternate between Sleep (or Low-power) and Active modes of operation by studying the energy-age of incorrect information trade-off with timer-based sleep-wake scheduling. In the following, we describe the contributions of the thesis in more detail.

MPC-mean: MPC-based dynamic EEE coalescing technique for mean of the queue waiting time

Frame coalescing is a well-established technique which manages the low power idle (LPI) mode supported by EEE interfaces. Frame coalescing enables EEE interfaces to remain in the LPI mode for a certain amount of time upon the arrival of the first frame (timer-based coalescing), or until a predefined amount of traffic accumulates in the transmission buffer (size-based coalescing). In this work, we propose a novel open-loop dynamic coalescing technique, namely MPC-mean, that is inspired by model predictive control (MPC) and queuing theory. In contrast to conventional timer-based coalescing, the proposed method enables the update of the timer parameter repeatedly throughout the duration of the LPI mode of a single coalescing cycle by taking into account the arrival instants and sizes, of the frames waiting in the buffer. The proposed MPC-mean method attempts to minimize the energy consumption of the Ethernet link, under constraints on the mean of the queue waiting time.

Our results for MPC-mean have been presented in IEEE High Performance Switching and Routing (HPSR) 2024 [15].

MPC-tail: MPC-based dynamic EEE coalescing technique for tail of the queue waiting time

We extend our study on MPC-based coalescing in EEE to control the tail of the queuing delays. Similar to MPC-mean, the proposed MPC-tail method is an open-loop controller that takes advantage of updating the timer parameter multiple times throughout a single coalescing cycle by considering the actual arrivals that take place before the timer update. In the MPC-tail scheme, an upper bound is enforced on the tail of the waiting time distribution using a queuing model which is based on M/PH/1 or M/ME/1 queues where the frame arrival process within the coalescing cycle is modeled as a Poisson arrival process and service times are modeled with PH-type (Phase Type) or the more general ME (Matrix Exponential) distributions, respectively. Moreover, for MPC-tail, delay violation probabilities are sensitive to the shape of the distribution of the service times. Therefore, we propose to use a mixture of Concentrated Matrix Exponential (CME) distributions [16] for accurately modeling Ethernet frame sizes. The CME model enables us to numerically calculate the tail of the waiting time using the M/ME/1 queue model and the optimum timer parameter is obtained through a simple search algorithm.

Our results regarding MPC-mean and MPC-tail have been published in IEEE Transactions on Green Communications and Networking [9].

The energy-age of incorrect information trade-off with timer-based sleep-wake scheduling

Reduction of the energy consumption of wireless sensor nodes is key for the successful deployment of Internet of Things (IoT) services and applications. This work investigates the trade-off between energy and information freshness from the perspective of the transmission module of IoT sensor nodes which can alternate between Sleep (or Low-power) and Active modes of operation. In particular, timer-based sleep-wake scheduling is investigated for a single sensor node. A variation of Age of Incorrect Information (AoII), which is termed as AoII+ in the literature, is used as the performance metric for information freshness. Then, we provide closed-form expressions for both average AoII+ and average power when updates take place according to a Poisson process and service times are either deterministic or phase-type distributed. Subsequently, we propose two optimal schedulers, namely AoII-Idling Scheduler (AoII-IS) and AoII-Zero Idling Scheduler (AoII-ZIS), which are devised to control the transitions between the Sleep and Active modes. For this purpose, we obtain the optimal timer parameters in closed form for the proposed schedulers that minimize the power consumption while the average AoII+ is kept below a pre-defined threshold. Moreover, we propose an adaptive version of both schedulers which dynamically tune their timer parameter when the Poisson update rate is allowed to vary in time.

Our results regarding the energy-age tradeoff have been accepted for presentation in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2025.

1.2 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 provides the necessary background on MPC, EEE and PH-type distributions as well as a brief review of sleep-wake scheduling for communication systems. In Chapter 3, we present the MPC-based frame coalescer to control the average of queuing delays, MPC-mean. In Chapter 4, the MPC-based frame coalescer to control the tail of queuing delays, MPC-tail. Chapter 5 is dedicated to the energy-age of incorrect information trade-off with timer-based sleep-wake scheduling. Finally, Chapter 6 concludes the thesis, and provides some directions for future research.

Chapter 2

Preliminaries and Literature Review

In this chapter, we provide preliminaries on MPC, EEE and PH-type (or ME) distributions, which will be required for the material covered in this thesis. We also provide a brief overview of sleep-wake scheduling for energy efficiency in communication systems. In particular, we will review various types of sleep-wake scheduling schemes and their applications to example systems to control various QoS metrics of interest.

2.1 Model Predictive Control

In the automation of technical systems, feedback controllers, also known as closed-loop controllers, evaluate a reference value r against a measured variable y to determine an appropriate manipulated variable u based on the deviation $e = r - y$ (see

Fig. 2.1). These controllers can be categorized into three types based on their operational principles: classical controllers, predictive controllers, and repetitive controllers. Classical controllers, such as PID controllers, bang-bang controllers, and state controllers, focus solely on past and current system behavior, making them reactive to deviations. In contrast, predictive controllers utilize a system model to forecast future behavior and anticipate deviations from the reference. Repetitive controllers consider the system's behavior from previous cycles to compute an optimal trajectory for the next cycle.

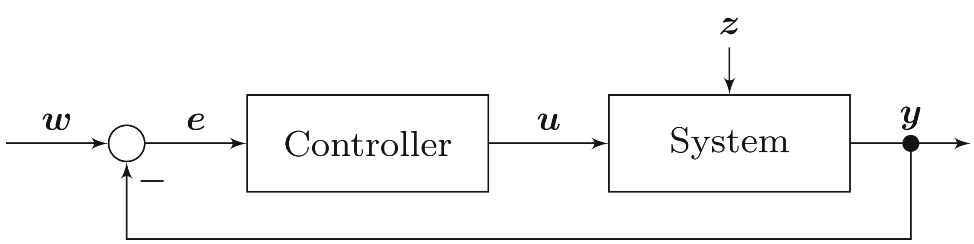

Figure 2.1: Block diagram of a classical feedback control loop (e.g. PID control).

Among these, the PID controller is the most recognized and widely used in industrial applications. While multiple setup rules exist, finding appropriate parameters can be challenging, especially for nonlinear or time-variant systems. The efficacy of any feedback design is inherently constrained by system dynamics and model precision. Therefore, achieving perfect tracking of time-varying reference trajectories through feedback control alone is theoretically infeasible, regardless of the design approach.

Specific scenarios, such as actuator limitations, necessitate tailored solutions that are often heuristic, making them difficult to comprehend and maintain. More advanced control methods, including sliding mode controllers and back-stepping controllers, tend to be similarly abstract and more complex.

Interestingly, the pioneers of Model Predictive Control (MPC) theory noted that traditional control methods adequately address 90% of control challenges. They suggested that advanced control strategies are only required for the remaining 10%. However, MPC is a viable solution for nearly all control issues, including those previously deemed unmanageable due to insufficient theoretical understanding or a lack of confidence in its feasibility. MPC operates through continuous real-time optimization of a mathematical model of the system, allowing it to predict future system behavior and optimize the manipulated variable u (illustrated in Fig. 2.2). This method offers intuitive parameterization by refining a process model, albeit at the expense of greater computational demands compared to classical controllers.

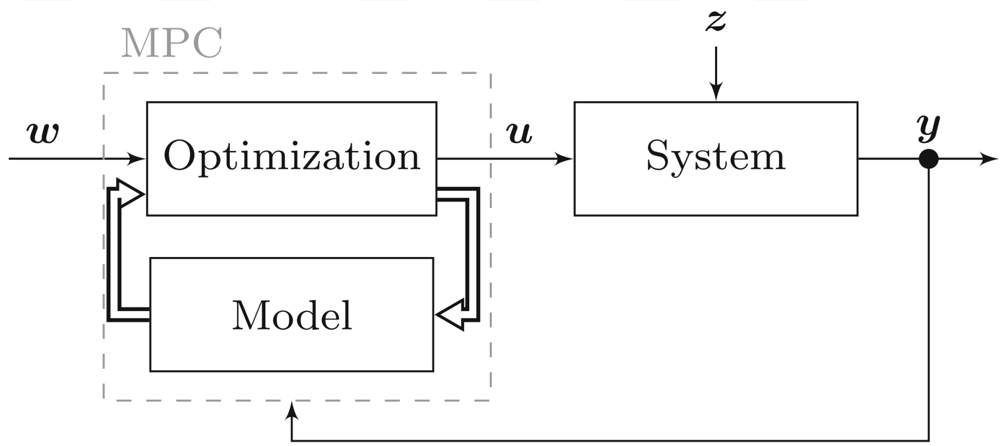

Figure 2.2: Simplified block diagram of a MPC-based control loop.

The anticipatory nature of MPC, along with its ability to account for stringent constraints, enhances its value for managing real systems. As computational power increases and models for complex processes become more accessible across various domains, MPC enables the control of previously unimaginable systems.

Since MPC is based on widely available models across numerous fields, it leverages existing knowledge and saves control specialists from explicitly formulating a control law. Instead, MPC automates the control law through model-based optimization. The implicit formulation, flexibility, and explicit incorporation of models are the primary benefits of MPC.

2.1.1 Formulation of MPC

Model Predictive Control (MPC) encompasses a class of advanced control strategies that rely on an explicit model to forecast a system's future behavior. Based on these predictions, MPC determines an optimal control input u by solving a constrained optimization problem at each time step. A key distinguishing feature of MPC is its direct handling of system constraints, which is uncommon among traditional control methods. Typically, the objective (or cost) function is structured to ensure that the system output y closely follows a desired reference signal r over a specified prediction horizon defined by N_2 (see Fig. 2.3). Although the controller calculates an optimal sequence of future control actions, only the first input in this sequence is applied to the system. This process—prediction followed by optimization—is repeated at every time step, which is why MPC is often referred to as *receding horizon control*. The underlying principle is that performing a series of short-term optimizations leads to long-term optimal behavior, as short-horizon predictions tend to be more accurate than long-term forecasts. The combination of real-time prediction and online optimization is what fundamentally differentiates MPC from traditional controllers that use precomputed control laws.

The prediction horizon N_2 must be chosen sufficiently long to capture the system's dynamic response to changes in the manipulated variable u. Time delays in the system can be accounted for either by setting a lower prediction horizon defined by N_1 or by embedding them directly into the system model. The latter approach is generally more intuitive, with N_1 often set to 1 to accommodate computation time meaning the control input u is determined in one step but applied in the next.

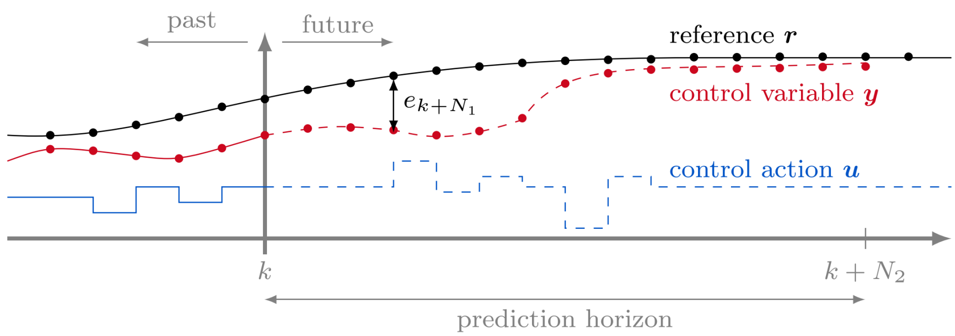

Figure 2.3: Function principle of a model-based predictive with horizons.

For a general system defined as:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= h(x(k)) \end{aligned} \tag{2.1}$$

MPC seeks to minimize a user-defined cost function J, as shown in Equation (2.2). A typical example involves minimizing the tracking error between the reference trajectory r and the predicted system output y, expressed as:

$$\min_{\mathbf{u}} \quad J(x(k), u) = \sum_{N_1}^{N_2} ||r(k+i|k) - y(k+i|k)|| \tag{2.2}$$

where $\mathbf{u}$ is the vector of control input sequence. Note that only the first input in this sequence is applied to the system at each step. Here, the norm $\|.\|$ may be chosen according to application-specific needs (e.g., 1-norm, 2-norm). The notation $y(k+i|k)$ represents the predicted output at time $k+i$, made at the current time step k.

2.1.2 Applications of MPC

Recent research on various disciplines verifies the usefulness of MPC. A comprehensive review that covers the theory, historic evolution, and practical applications of MPC is presented in [17]. MPC is especially a convenient tool for chemical, pharmaceutical, oil and gas industries due to large time delays and the non-linearity of the underlying processes [18] [19]. Building climate and energy management is another application field of MPCs which is significantly successful when the system model incorporates statistical uncertainties [20] [21]. In addition to these main fields of applications, MPC is also applied to cruise control [22] [23], and path tracking and planning for autonomous vehicles [24] [25].

Prediction-based control algorithms are also applicable to queuing system and communication network-related optimization problems where comprehensive predictive models are available. The authors in [13] combined reactive and predictive fluid model-based approaches for path selection in multi-path mmWave communications. Prediction-based model is also proposed for sleep-wake scheduling of energy efficient Ethernet [26]. Link state management for both reliable and unreliable links subject to uncertainties turns out to be a promising application field for MPCs. Optimization of server farm related concepts such as speed-up functions and dynamic server provisioning subject to waiting time or age of information related constraints are also prospective problems which MPCs are applicable.

2.2 Energy Efficient Ethernet

The IEEE 802.3az standard, also known as energy efficient Ethernet (EEE), aims to decrease the energy consumption of an Ethernet interface by putting it in a low power idle (LPI) mode when the link is not in use. The LPI mode, as defined in the

EEE amendment to the Ethernet standard [6], reduces energy consumption by 90% compared to normal operation. However, during the LPI state, EEE powers down most of its components to save energy and, the transitions to and out of the LPI mode take a certain amount of time to complete. As a result, frames that arrive at an EEE will experience larger queuing delays compared to an interface without LPI capability. Additionally, energy consumption at these transition states is equal to that of normal operation. Therefore, in order to fully exploit the energy efficiency of the LPI mode, the strategy to select the duration of the LPI mode must carefully be considered. The standard does not offer guidelines for the design of such strategies since, it would highly be associated with the optimization goal and constraints that have been imposed on QoS.

A sample path of unfinished workload for EEE with coalescing is illustrated in Fig. 2.4 in which the workload at the interface is plotted as a function of time for the duration of a single coalescing cycle. In this example, abrupt jumps represent packet arrivals and the amount of jumps are the service time needed to process the packet which is a function of packet size and interface speed. When the buffer gets empty, EEE interface makes a transition to LPI state after staying in the Sleep state for T_s duration. It remains in the LPI state for a random duration of T_{off}. When the wake-up conditions for the EEE coalescing algorithm are met, the interface immediately abandons LPI state and, after a transition of T_w duration in the wake-up state, it starts to process packets in the active state until buffer gets empty. Note that T_e represents the duration which the interface is empty at the beginning of each coalescing cycle.

2.2.1 Control Mechanisms for EEE

The IEEE 802.3az standard does not enforce the use of a certain mechanism governing the transitions from the LPI mode to the Active mode for which there is a

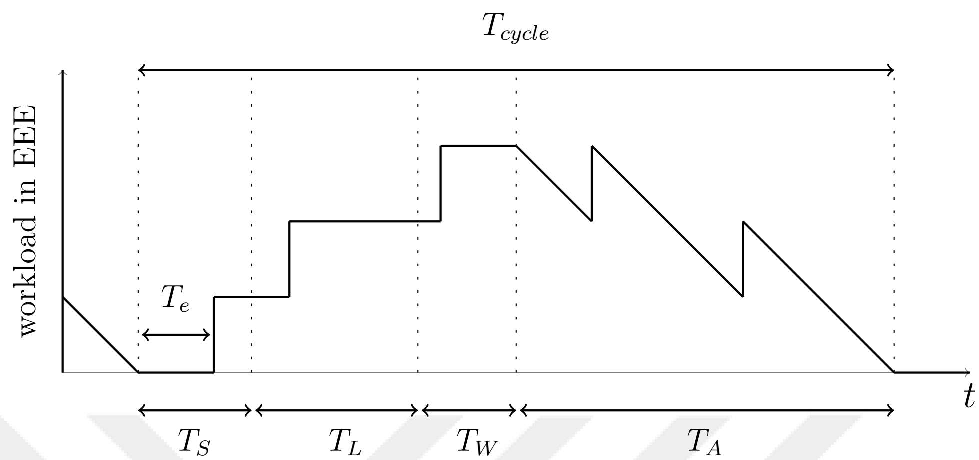

Figure 2.4: Sample path of unfinished workload for EEE with coalescing.

need to develop effective strategies while trading off energy efficiency and queuing delays. Most EEE mechanisms that have been proposed in the literature are variations of the frame coalescing (also known as burst transmission) technique which aims to keep the interface in the LPI mode until a certain amount of traffic is ready for transmission [27], [28]. Consequently, power consumption can be reduced but at the expense of increased frame delays [29].

Frame coalescing algorithms can be divided into two main types based on the type of coalescing parameters that are used to control the duration of the LPI mode [27]. In timer-based coalescing, a timer is triggered when the first frame arrives at an empty queue and a transition to the Active mode via a Wake-up (W) state is initiated when the timer expires. On the other hand, in size-based coalescing, the interface is kept in the LPI mode until a certain number of frames (or bytes) accumulate in the queue. In both types of algorithms, the LPI mode is entered (via the Sleep (S) state) as soon as the transmission queue empties. Hybrid coalescers that operate with both timer and size parameters have also been proposed in the literature [27], [29]. Static selection of the associated parameters (timer or size parameters) may lead to a significant performance degradation when the statistical parameters of the traffic, e.g., frame arrival rate, mean frame size, etc., vary in time. Therefore, several EEE mechanisms have been proposed to dynamically tune the coalescing parameters based on the instantaneous conditions of the traffic. Closed-loop mechanisms employ a feedback controller to dynamically select the coalescing parameters based on the measured power consumption figures and/or frame waiting times, whereas open-loop mechanisms do not rely on a feedback loop while considering solely the instantaneous estimates of the underlying traffic parameters in order to select a proper coalescing parameter. Whether open- or closed-loop mechanisms are in place, the proposed algorithms for size- and timer-based coalescing often rely on expressions for the relevant performance metrics such as average queuing delay, average power consumption, average duration of the LPI mode, etc., which are derived using queuing models under certain assumptions on the distributions of frame inter-arrival times and service times. However, such expressions have a certain margin of error due to the underlying assumptions.

2.2.2 Power Consumption Rate of EEE

The ratio of energy consumption of an EEE interface to a power-unaware interface that does not support LPI mode is given in [29] with the following formula,

$$\varphi = 1 - (1 - \varphi_{off})(1 - \rho)\frac{E[T_{off}]}{E[T_{off}] + T_w + T_s} \tag{2.3}$$

where φ_{off} stands for the energy consumption ratio of the LPI mode to the active mode. Since parameters φ_{off}, T_w and T_s are known characteristics of a given EEE-capable interface, the power efficiency depends only on $E[T_{off}]$ for constant traffic conditions. For a standard 10GBASE-T ethernet link it is assumed that $\varphi_{off} = 0.1$, $T_w = 4.48\,\mu s$ and $T_s = 2.88\,\mu s$.

2.3 PH-Type and ME Distributions

For the purpose of describing PH-type distributions, a Markov process with state-space $\mathcal{S} = \{1, 2, \ldots, m, m+1\}$ is defined with the last state being absorbing, initial probability vector α of size $1 \times m$, and the following generator:

$$\begin{bmatrix} A & h \\ \mathbf{0} & 0 \end{bmatrix},$$

where the sub-generator A is $m \times m$, h is a vector of size $m \times 1$ such that $h = -Ae$ and e denotes a column vector of ones of appropriate size. The time required for absorption for this Markov process is represented by the random variable Θ which is called PH-type characterized with the pair (α, A), i.e., $\Theta \sim PH(\alpha, A)$, where the notation $\sim$ is synonymous with "distributed according to" [30]. The cumulative distribution function (CDF) and probability density function (PDF) of $\Theta \sim PH(\alpha, A)$, denoted by $F_\Theta(x)$ and $f_\Theta(x)$, respectively, are written as

$$F_\Theta(x) = 1 - \alpha e^{Ax} e, \quad f_\Theta(x) = -\alpha e^{Ax} A e, \quad x \geq 0. \tag{2.4}$$

Phase-type (PH-type) distributions have been successfully used for modeling independent and identically distributed non-exponential inter-arrival and/or service times in various disciplines [31]. PH-type distributions are subcases of the more general Matrix Exponential (ME) distribution. A random variable $\Theta \sim ME(\alpha, A)$ when the PDF (or the CDF) of the random variable Θ is written as in (2.4) with $f_\Theta(x) \geq 0, \quad \int_0^\infty f_\Theta(x)\,\mathrm{d}x = 1$ [32], [33]. However, for ME distributions, the parameters α and A need not be a probability vector and a sub-generator, respectively, making ME distributions relatively more general than their PH-type counterparts. Moreover, results known for queuing systems using PH-type distributions can also be applied to those using ME distributions while employing the same existing methods [34–36].

In [16], a family of ME distributions, called concentrated ME (CME) distributions, are proposed to approximate deterministic random variables. The main result of [16] is that the asymptotic behavior of the Squared Coefficient of Variation (SCV) of the proposed CME distributions of order l is $2/l^2$. For the purpose of approximating deterministic distributions, CME distributions have a relatively faster convergence rate compared to Erlang distributions which have the lowest SCV value $1/l$ among PH-type distributions with order l [37].

2.4 Sleep-wake Scheduling in Communication Systems

Queuing systems with sleep-wake scheduling has been widely studied to stochastically model communication systems, production, and manufacturing applications. Various activation schemes are proposed to control the transitions from the Low-power state to the Active state, such as N-policy [38], D-policy [39], single-sleep scheme [8], multiple-sleep scheme [40]. For single server queues, the conventional approach is to enter sleep mode as soon as there are no customers in the system and control the sleep mode length either with activation schemes that is tuned by queue length N (N-policy [41]) or with a timer T (T-policy [42]). As a means to achieve tighter control over the sleep period, various combinations of N and T policies are also feasible [43] [44]. The authors in [45] first introduced the concept of min(N, T) policy which observes the system periodically with T discrete time units and terminates the sleep period when there is at least one customer at an observation instant, or terminates immediately if there are N customers in the queue. [46] proposed an NT policy on M/G/1 queues for which the sleep period expires either when the waiting time of the customer at the head of the queue reaches T, or there are N customers in the system. Some studies also study idling schemes for sleep-wake servers instead of employing strategies that immediately make a transition to the Sleep mode when the queue empties. A hysteretic timer scheme under which the server stays idle until either a random threshold is reached or a new packet arrives is studied in [47] and [48] for base stations. Other sleeping schemes such as Bernoulli sleep and conditional sleep have been discussed in [7] and [49], respectively.

Objective function for sleep-wake scheduling optimization is usually related to the energy consumption of the server where energy consumption rates of Setup, Sleep, Active states of the server vary depending on the system [46]. Most existing works employ delay as the QoS constraint by studying the energy-delay trade-off for sleep-wake scheduling [8], [1], [9], [50]. The energy-delay trade-off for EEE has also been studied considering queue waiting times. The energy-delay trade-off of EEE is investigated for timer-based coalescing in [51] and for size-based coalescing in [52]. Furthermore, the authors of [53] propose an analytical model of queuing delay for byte-based coalescing which terminate the LPI state when the accumulated workload is above a threshold.

There have been few attempts to consider the relation between information freshness and energy from the perspective of the sleep-wake server. The authors of [7] analyzed the relation between energy consumption and AoI for a single-server multi-source bufferless system under different sleep-wake strategies. Energy and PAoI trade-off was studied for an LCFS single-server single-source system in [10]. Also, the authors in [11] studied the relation between PAoI and energy consumption for a transmitter which utilizes the DRX mechanism and proposed an optimal policy. Recent work of [54] consider Age of Synchronization (AoS) energy trade-off for multi source setting.

Most of the existing efforts focus on the evaluation and control of sleep-wake strategies under static traffic conditions [7], [10], [11]. Although the assumption of static traffic parameters facilitates analysis, static strategies often lead to a significant performance degradation when the statistical parameters of the traffic, e.g., arrival rate, mean processing time, etc., vary in time. Therefore, adaptive sleep-wake control for delay has been proposed in the literature for real-life applications such as DRX [55], EEE [1], [9]. Additionally, recent work in [12] proposes an adaptive on/off controller for an energy harvesting receiver to minimize the average AoI through a Markov decision process formulation. The complexity of the underlying mechanism to control sleep-wake scheduling adaptively is of great importance for the feasibility of its implementation. Even if the system input parameters such as arrival process, service time distribution assumed to be time invariant, the optimization problem may consist of complex and non-linear functions of the control parameters [56].

2.5 Chapter Summary

In this chapter, we have provided a brief review of MPC fundamentals and applications for MPC inspired EEE mechanisms that are to be proposed in the following chapter. We have also briefed the EEE principles and control mechanisms. Then, we have provided preliminaries for PH (ME and CME) type distributions which will be employed for modeling the service times of queuing systems throughout different chapters. Finally, we have provided a brief review for sleep-wake scheduling for communication systems focusing on various strategies as well as QoS metrics that have been considered in the literature.

Chapter 3

MPC-mean: MPC-based dynamic EEE coalescing technique for the mean of the queue waiting time

In this chapter, we propose a novel open-loop dynamic coalescing technique, namely MPC-mean, that is inspired by model predictive control (MPC). In contrast to conventional timer-based coalescing, the proposed method enables the update of the timer parameter repeatedly throughout the duration of the LPI mode of a single coalescing cycle by taking into account the arrival instants and sizes of the frames waiting in the buffer. The proposed MPC-mean method attempts to minimize the energy consumption of the Ethernet link, under constraints on the mean of the queue waiting time. The effectiveness of the algorithm is validated using simulations with synthetic and actual traffic traces. In exhaustive simulation studies, we have observed consistent energy consumption gains with MPC-mean with better control over average queuing delays, when compared to the open-loop dynamic timer-based coalescer from the existing literature.

Our results for MPC-mean have been presented in IEEE High Performance Switching and Routing (HPSR) 2024 [15].

The chapter is organized as follows. We give a brief review of analytical models and frame coalescing algorithms for EEE in Section 3.1. We describe the problem definition and contributions in Section 3.2. We provide the system model in Section 3.3. We introduce the details of the proposed MPC-mean in Section 3.4. Section 3.5 presents extensive numerical results. Finally, we conclude the chapter in Section 3.6.

3.1 A Brief Review of Analytical Models and Frame Coalescing Algorithms for EEE

Analysis of frame coalescing for EEE has been studied from different perspectives. Early efforts solely focused on the analysis of power consumption under frame coalescing [57], [58]. Later, research studies took a more comprehensive approach while considering the effects of frame coalescing on queuing delays. The energy-delay trade-off is investigated for timer-based coalescing in [51] and for size-based coalescing in [52]. Furthermore, the authors of [53] propose an analytical model for byte-based coalescing which terminates the LPI state when the accumulated workload is above a threshold. The work of [59] investigated the EEE interfaces as a part of a network rather than evaluating the interface in isolation and proposed a theoretical model for the network delay. Most analytical models developed for 10GBASE-T consider the joint use of time and size based coalescing [60], [29], [61], [62].

In frame coalescing, a number of frames are deliberately allowed to accumulate (or coalesce) at a coalescing buffer in the LPI mode rather than exiting the LPI mode immediately upon a single new frame arrival [27], [28]. Consequently, power consumption can be reduced but at the expense of increased frame delays [29]. If the exit from the LPI mode occurs when a timer expires (a certain number of frames have been accumulated) after the first frame arrival, then we have timer-based (size-based) coalescing [27]. In the references [27] and [29], a hybrid scheme is used and the exit from the LPI mode occurs at the earlier of the following two events: timer expiration or accumulation of a certain number of frames in the coalescing buffer. Algorithms for determining the timer and size parameters are proposed in [60]. A new analytical model for EEE interfaces with hysteresis is studied in [63] which proposes to implement frame coalescing before traffic reaches the EEE interface. The work of [64] considers energy savings for bundled EEE interfaces and proposes a load sharing mechanism which activates additional links only when active links are fully utilized.

Adaptive tuning algorithms employ a closed-loop controller to have the system converge to a desired state. Dynamic tuning of the size parameter to bound the energy consumption is considered in [65] whereas the references [66] and [67] aim to control the average queuing delay by selecting appropriate timer and size parameters, respectively. On the other hand, the references [26] and [68] opt to minimize an objective function that is a linear combination of the power consumption and average queuing delay by adjusting the length of the sleeping periods. In order to control an EEE interface under varying traffic conditions, closed loop systems rely on the appropriate selection of the loop gain parameter for the speed of convergence and stability. Open loop control methods do not rely on the use of a gain parameter. The authors of [1] propose a dynamic tuning algorithm for both size and timer parameters to control the average queuing delay. The tuning is based on the average queuing delay expressions obtained in [29] with the assumption of Poisson arrivals. In the context of adaptive algorithms, the MPC coalescer belongs to the class of open-loop controllers that takes advantage of updating the timer parameter multiple times throughout a single coalescing cycle by considering the actual arrivals that take place before the timer update.

Finally, there have been few attempts to control the tail of the queuing delays which may be required for certain applications with stringent requirements for the tail of the delay distribution [69]. The work of [51] emphasizes the importance of the tail of the delay and consequently derives the delay distribution for an EEE link using timer-based coalescing that can be used to select the timer parameter to control the tail in a static manner. Furthermore, a tail control algorithm is presented in [70] for the dual-mode control of 40-100 Gbps links.

3.2 Problem Statement and Contributions

Encouraged by successful industrial MPC applications, this chapter proposes a novel open-loop EEE control mechanism, MPC-mean, which updates the timer parameter for a single coalescing cycle repeatedly at multiple instants throughout the duration of the LPI state using MPC and queuing theory. This approach is in contrast to determining the timer parameter once and upon the arrival of the first frame when in the LPI state. In the proposed MPC-mean scheme, for a given MPC instant at which a certain number of frames with a certain workload held in the coalescing buffer, we analytically model the residual cycle (remaining portion of the coalescing cycle) to derive expressions related to the coalescing cycle of interest, including the expected waiting time, the expected duration of the coalescing cycle, etc., all obtained as a function of the timer parameter. The model is based on M/G/1 queuing system where the frame arrival process within the coalescing cycle is modeled as a Poisson arrival process and service times are generally distributed. Subsequently, we derive the timer parameter in closed-form in order to minimize the energy consumption in the current coalescing cycle while the mean waiting time is kept below a given desired threshold.

3.3 System Model

For the analysis of the residual cycle, the frame arrival process is assumed to be Poisson with rate λ. The frame service times are assumed to have a general distribution with the mean service rate μ and variance σ^2. Obviously, the utilization factor (or load) $\rho = \lambda/\mu < 1$ must be satisfied for stability. A sample path of the workload (unfinished work in time units) for EEE with coalescing is illustrated in Fig. 3.1 in which the workload at the interface is plotted as a function of time for the duration of a single coalescing cycle. In this example, abrupt jumps represent frame arrivals and the amount of jumps is the service times needed to process and transmit the frame which is a function of the frame size and the interface speed. When the buffer empties, the EEE interface makes a transition to the LPI mode after staying in the Sleep state for a fixed duration denoted by T_S. The interface remains in the LPI mode for a random duration of T_L. When the wake-up condition for the EEE coalescing algorithm is met, the interface abandons the LPI mode and, after a fixed duration of T_W in the Wake-up state, it starts to transmit frames in the Active mode using FCFS (First Come First Serve) queuing until the buffer empties again, for a random duration of T_A. This process repeats itself. Note that T_e represents the random duration for which the coalescing buffer is empty at the beginning of coalescing cycle.

3.4 MPC-Based Coalescing for Average Queuing Delay

An MPC instant is one where the buffer is non-empty and the interface is in either the LPI or Sleep states. At such an instant, the number of arrivals that have accumulated in the buffer is denoted by $N > 0$. The service time introduced by the n^{th} arrival

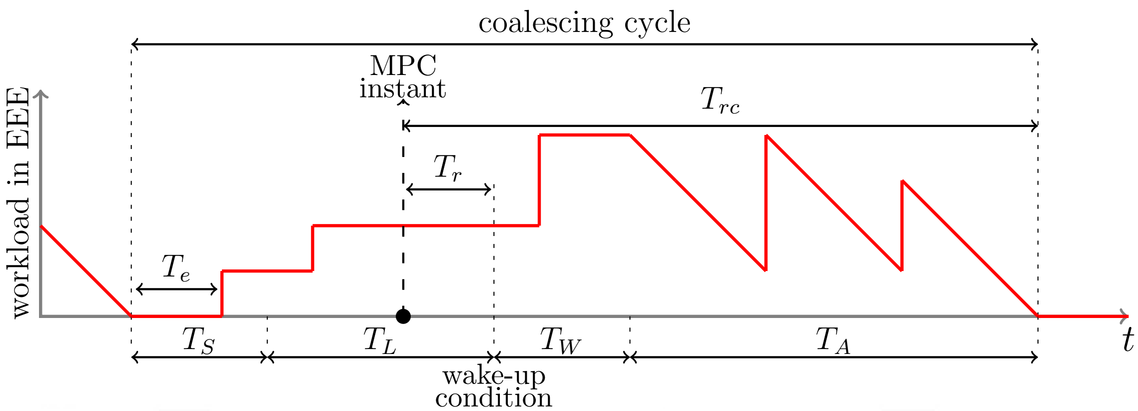

Figure 3.1: Sample path of the workload (unfinished work in time units) for EEE with MPC-mean.

and its queue wait until the MPC instant are denoted by s_n and $q_n, n = 1, 2, \ldots, N$, respectively. Then, the total accumulated workload is given by

$$w = \sum_{n=1}^{N} s_n. \tag{3.1}$$

Let T_r denote the time to transition to the W state at the current MPC instant. A sample MPC instant and the corresponding T_r duration is depicted in Fig. 3.1.

For a given value of T_r, the total queue wait that will be experienced by the n^{th} arrival is a deterministic value that is given by

$$Q_n = T_r + a_n, \tag{3.2}$$

where

$$a_n = \begin{cases} q_n + T_W, & \text{if } n = 1, \\ q_n + T_W + \sum_{l=1}^{n-1} s_l, & \text{if } n > 1. \end{cases} \tag{3.3}$$

Next, we focus on the MPC coalescer MPC-mean which ensures that the expected queuing delay within the current cycle is below a threshold denoted by τ by adjusting the timer parameter T_r. The mean queuing delay of the current cycle at the MPC instant can be written by combining information about prior arrivals and the analysis results to be obtained for the residual cycle, as follows,

$$\mathbb{E}[W_c] = \frac{\sum_{n=1}^{N} Q_n + \mathbb{E}[\Sigma]}{N + \mathbb{E}[N_r]}, \tag{3.4}$$

where $\mathbb{E}[\Sigma]$ and $\mathbb{E}[N_r]$ are the expected sum of queuing delays and expected number of arrivals that will occur within the residual cycle, respectively. Under Poisson arrivals, the expected number of arrivals within the residual cycle is given by

$$\mathbb{E}[N_r] = \lambda \mathbb{E}[T_{rc}], \tag{3.5}$$

where T_{rc} is the random variable that stands for the residual cycle length as illustrated in Fig. 3.1. For any given coalescing cycle, all the arrivals that occur within the residual cycle must be served in the interval of length $T_{rc} - T_r - T_W - w$. Therefore, the identity $\lambda \mathbb{E}[T_{rc}] = \mu(\mathbb{E}[T_{rc}] - T_r - T_W - w)$ holds. Consequently, the expected length of the residual cycle is given by

$$\mathbb{E}[T_{rc}] = \frac{T_r + T_W + w}{1 - \rho}, \tag{3.6}$$

and

$$\mathbb{E}[N_r] = \frac{\lambda}{1 - \rho}(T_r + T_W + w). \tag{3.7}$$

The following theorem gives a closed-form expression for $\mathbb{E}[\Sigma]$ for generally distributed service times.

Theorem 1. *Consider an M/G/1 queue model of the EEE link with Poisson arrivals with rate λ, generally distributed service times with mean $1/\mu$ and variance σ^2, and load $\rho = \lambda/\mu < 1$. Also, let us be given an MPC instant during LPI or Sleep states at which we have a workload of w in the buffer and the transition to the W state will take place in T_r seconds after the MPC instant. Then, the expected sum of the* *queuing delays within the residual cycle, denoted by* $\mathbb{E}[\Sigma]$*, is given by the following closed-form expression:*

$$\mathbb{E}[\Sigma] = \mathbb{E}[T_{rc}] \left[\frac{\lambda^2\sigma^2 + \lambda}{2(1-\rho)} - \frac{1+\rho}{2} + \frac{\lambda(T_r + T_W + w)}{2} \right]. \tag{3.8}$$

Proof. The following residual cycle analysis is valid for any instant throughout the coalescing cycle where the buffer is non-empty with $w > 0$ and $T_r > 0$. To obtain the performance metrics of interest, an auxiliary regenerative process, $W_r(t)$ is introduced and, illustrated in Fig. 3.2 for three consecutive cycles. Each cycle of $W_r(t)$ is a realization of the residual coalescing cycle and, length of the k^{th} cycle is indicated by T_{rc}^k. This process keeps track of the queuing delay of a virtual arrival that would occur at time t. Each residual cycle starts from level T_Σ which is the total delay that an arrival at the MPC instant would experience with

$$T_\Sigma = T_r + T_W + w. \tag{3.9}$$

The process decreases with a drift of -1 unless an arrival occurs. As soon as $W_r(t)$ hits to level zero, the process resets itself to level T_Σ as indicated by dashed patterns and initiates a new realization of the residual coalescing cycle.

Under Poisson arrivals, the trajectory depicted in Fig. 3.2 is statistically equivalent to the queuing delay experienced by a virtual arrival to a single-server queuing system in which the server takes fixed-length vacations, lasting T_Σ, and returns to work whenever there are one or more frames waiting at the end of a vacation period. Therefore, the distribution of $W_r(t)$ is the same as the queuing delay distribution of a sleep-wake server system of this kind. The queuing delay experienced by the first arrival of vacation period is denoted by the random variable W_f^r and, the random duration for which the queue is empty denoted by T_e^r. For the sleep-wake server system described above, the findings of [71] can be employed to derive the metrics

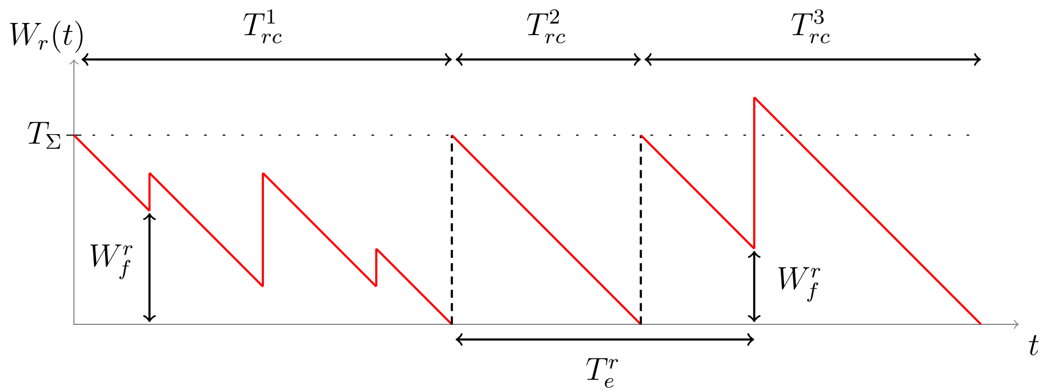

Figure 3.2: Sample path for the regenerative process $W_r(t)$ for three consecutive cycles.

related to queuing delay. The expected value of $W_r(t)$ can be written as,

$$\mathbb{E}[W_r] = \frac{\lambda\sigma^2 + 1}{2(1-\rho)} + \frac{1-\rho}{2\lambda} + \frac{\mathbb{E}[T_v^{r2}] - 2\mathbb{E}[T_v^r T_e^r]}{2\mathbb{E}[T_v^r]}, \tag{3.10}$$

where $T_v^r = W_f^r + T_e^r$ is the total vacation period elapsed between two consecutive active periods. The first two moments of T_v^r are obtained as follows,

$$\mathbb{E}[T_v^r] = T_\Sigma \sum_{n=0}^{\infty} e^{-\lambda T_\Sigma n}, \tag{3.11}$$

$$\mathbb{E}[T_v^{r2}] = {T_\Sigma}^2 \sum_{n=0}^{\infty} (2n+1) e^{-\lambda T_\Sigma n}, \tag{3.12}$$

and $\mathbb{E}[T_v^r T_e^r]$ is given by

$$\mathbb{E}[T_v^r T_e^r] = {T_\Sigma}^2 \sum_{n=0}^{\infty} n e^{-\lambda T_\Sigma n} + \frac{T_\Sigma}{\lambda} \sum_{n=0}^{\infty} e^{-\lambda T_\Sigma n}. \tag{3.13}$$

Finally, substituting (3.11)-(3.13) into (3.10), we obtain

$$\mathbb{E}[W_r] = \frac{\lambda\sigma^2 + 1}{2(1-\rho)} - \frac{1+\rho}{2\lambda} + \frac{T_\Sigma}{2}. \tag{3.14}$$

Together with the use of renewal theorem and identities provided in (3.6) and (3.14), the expected sum of the queuing delays of arrivals that occur within the residual cycle is given by

$$\mathbb{E}[\Sigma] = \lambda \mathbb{E}[T_{rc}] \mathbb{E}[W_r], \tag{3.15}$$

which is equivalent to (3.8), completing the proof of Theorem 1. □

Equating the expression (3.4) to the desired average queuing delay τ, we obtain ${T_r}^*$ which denotes the residual timer duration to achieve a mean queuing delay of τ for the entire cycle of interest,

$${T_r}^* = \sqrt{k_1^2 + 2k_2} - k_1 - w - T_W, \tag{3.16}$$

where

$$k_1 = \frac{\lambda \sigma^2 + 1}{2(1-\rho)} - \frac{1+\rho}{2\lambda} + \frac{N(1-\rho)}{\lambda} - \tau, \tag{3.17}$$

and

$$k_2 = (1-\rho) \frac{N\tau + \sum_{n=1}^{N} n s_n - q_n}{\lambda}. \tag{3.18}$$

It is important to note that the timer parameter ${T_r}^*$ can be obtained in closed-form using (3.16) for MPC-mean at an MPC instant for generally distributed service times. The algorithm only needs the mean and variance approximations of the service and interarrival times. We propose to update these approximations at an update instant with the Robbins–Monro algorithm with update coefficient β [72].

3.4.1 MPC Instantiation and Implementation Remarks

MPC-mean coalescing allows one to update the residual length of the LPI mode at multiple instants throughout a single coalescing cycle. The expression (3.16) can be employed for the selection of the residual timer parameter for any choice of an

MPC instant throughout the LPI and Sleep modes. On one end, one might need to compute the residual timer value almost continuously. However, this approach is not practically feasible due to the computational burden it gives rise to. Instead, we propose a simple MPC instantiation algorithm with the number of instants within a cycle bounded by the parameter M. In the proposed algorithm, the first MPC instant, T^1_{MPC}, is immediately after the first arrival of the coalescing cycle by considering the recently introduced workload and naturally, $N = 1$, $q_1 = 0$, and w being the service time of this first frame of the cycle. Also let T_r^{*m} denote the timer parameter found at instant $m, m = 1, \ldots, M$. We propose to set the subsequent MPC instants T^m_{MPC} through the following iterative procedure:

$$T^m_{MPC} = T^{m-1}_{MPC} + 0.5T_r^{*(m-1)}, \quad 1 < m \leq M. \tag{3.19}$$

We propose to transition to the W state at time $T^M_{MPC} + T_r^{*M}$. However, when a value $T_r^{*m} > 0$ cannot be found at instant $m \leq M$, then we immediately transition to the W state by terminating the iterations. The MPC-mean algorithm using the MPC instantiation parameter M is denoted by MPC-mean(M).

The computational load for MPC-mean at an MPC instant is similar to that encountered at the first frame arrival of a coalescing cycle in [1], and the desired timer can be computed at the interface rate since the computational complexity of the operation is $O(1)$. The energy cost of updating the timer at multiple instants for MPC-mean is negligible since timer adaptation mechanisms for EEE only require software modifications on the NIC driver side, as also indicated in [66].

3.5 Numerical Examples

We first present performance results for MPC-mean(M) using simulations with static Poisson traffic and then Pareto interarrival times with shape parameter $\alpha = 2.5$,

Table 3.1: Average traffic parameters for actual traffic traces

ID	Date, Time	ρ	λ (kilo packets per second)
1	2020/04/08, 14:00	0.2631	370
2	2020/03/11, 14:00	0.2977	530
3	2020/01/01, 14:00	0.1775	628

where the latter model was used to capture the self-similar nature of Ethernet traffic [1]. In these simulations, the frame sizes are assumed to have a PH-type bi-modal distribution which is obtained by mixing the Erlang(10,100) and Erlang(10,1500) distributions with probabilities 0.3 and 0.7, respectively, where Erlang(k,v) stands for an Erlang distribution with order k and mean size v in units of bytes. Then, we evaluate the proposed techniques with MAWI frame traces captured at the WIDE backbone Ethernet link samplepoint-G [73]. Throughout the simulations with actual traffic traces, we utilized the Robbins-Monro algorithm described before with $\beta = 0.001$ to approximate the parameters dynamically. The information about the three traffic traces indexed from 1 to 3 (when they are taken, average utilization factors, etc.) that are used in the simulations are provided in Table 3.1.

We compared the effectiveness of MPC-mean with the open-loop dynamic timer-based coalescer which is proposed and shown to achieve a performance close to the lower bound in [1]. We only focused on the performance of MPC-mean(M) with $M = 2, 3$ due to the fact that MPC-mean(1) is similar to the benchmark dynamic coalescer in the way that they both select the timer parameter at the instant of first arrival within the coalescing cycle and, energy consumption gains for $M > 3$ are relatively limited despite the increased computational burden.

First, the effectiveness of MPC-mean is shown in Fig. 3.3 for Poisson traffic with the percentage reduction (or gain) $g = \frac{\varphi_{DC} - \varphi_{MPC}}{\varphi_{DC}}$ where φ_{DC} and φ_{MPC} denote the energy consumption ratios of the benchmark dynamic coalescer and MPC-mean, respectively, to a non-EEE interface provided by (2.3). Results are obtained for varying

Table 3.2: Average queuing delays (μs) achieved by dynamic timer-based coalescing [1] (DTC) and MPC-mean coalescing algorithms under Poisson and Pareto traffic

		Poisson				Pareto			
ρ	Coalescing	$\tau = 8$	$\tau = 16$	$\tau = 32$	$\tau = 64$	$\tau = 8$	$\tau = 16$	$\tau = 32$	$\tau = 64$
0.2	DTC [1]	7.998	15.997	32.004	63.989	7.261	15.180	31.201	63.274
	MPC-mean(2)	8.001	16.000	31.996	64.000	7.765	15.810	31.818	63.832
	MPC-mean(3)	8.001	16.002	32.001	64.003	7.847	15.859	31.883	63.894
0.4	DTC [1]	8.004	16.002	32.012	64.000	7.288	15.315	31.369	63.421
	MPC-mean(2)	8.002	15.999	32.002	64.001	7.657	15.669	31.682	63.713
	MPC-mean(3)	8.010	15.996	32.004	64.002	7.680	15.711	31.735	63.745
0.6	DTC [1]	8.004	15.998	31.999	63.998	7.138	15.192	31.251	63.289
	MPC-mean(2)	8.010	16.004	31.998	64.014	7.413	15.446	31.477	63.497
	MPC-mean(3)	8.010	15.998	32.002	63.993	7.442	15.483	31.514	63.526
0.8	DTC [1]	8.000	15.994	32.010	64.005	6.610	14.706	30.776	62.843
	MPC-mean(2)	8.000	15.993	31.990	64.008	6.818	14.900	30.954	63.000
	MPC-mean(3)	7.995	16.006	31.999	64.002	6.868	14.936	30.993	63.047

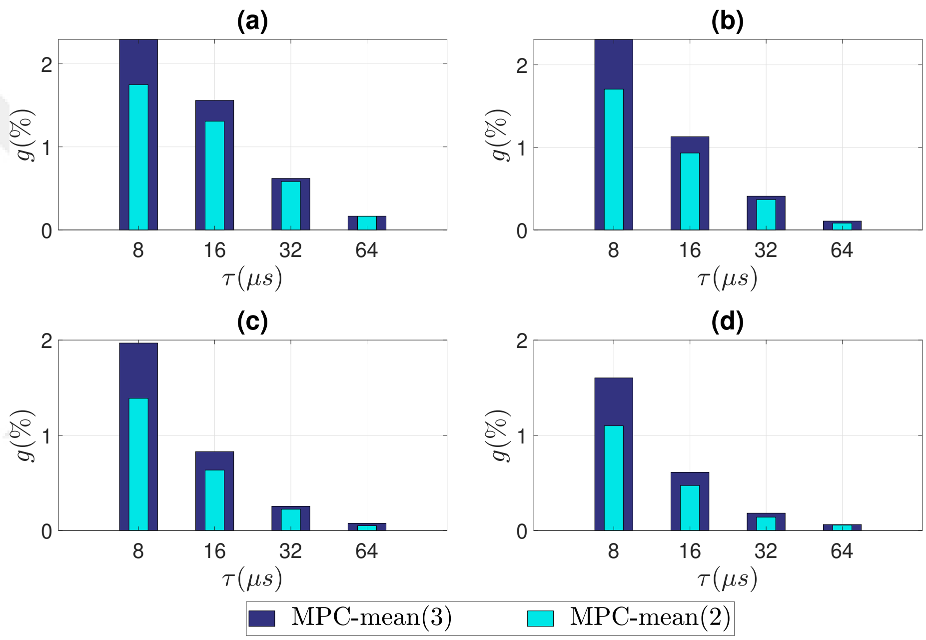

Figure 3.3: The percentage gain, g, in using MPC-mean against dynamic timer-based coalescing [1] for Poisson traffic with utilization factors (a) $\rho = 0.2$, (b) $\rho = 0.4$, (c) $\rho = 0.6$ and (d) $\rho = 0.8$.

utilization factors (from 0.2 to 0.8) and desired average queuing delays of τ ranging from 8 to 64 μs. The simulation results of [1] show that under heavy traffic scenarios with $\rho > 0.9$, transmission buffers fill up even under zero LPI durations and it was recommended to disable EEE to prevent the queuing delay from increasing even more. In order to evaluate and compare the performance of MPC-based coalescing in the region where the target delay parameters are achievable, we avoided heavy traffic scenarios and kept the load ρ in our simulations below 0.8.

In terms of energy consumption, we observe that MPC-mean consistently outperforms the dynamic timer-based coalescer with gains up to 2.37% attained under stringent target queuing delays. Moreover, MPC-mean(3) slightly outperforms MPC-mean(2) for all the scenarios we studied justifying the benefit of MPC deployment and its repeated optimization feature. Additionally, the resulting average queuing delays are presented in Table 3.2 which indicates that the target values are achieved with accuracy better than $0.015\mu s$ with both MPC-mean(2) and MPC-mean(3). Fig. 3.4 demonstrates that the gain in using MPC-mean increases for Pareto distributed interarrival times, i.e., up to 4.39% when compared to Poisson traffic case. This can be explained with the observation that MPC-mean uses the information about the actual frame arrivals that have occurred prior to MPC instants and hence it is more accurate than the benchmark policy.

For actual traffic traces, Fig. 3.5 shows that energy consumption gains up to 1.39% are achievable with MPC-mean. Furthermore, Table 3.3 demonstrates for traces 1 and 2 that the dynamic timer-based coalescer results in larger average queuing delay values than the target value whereas MPC-mean(2) and MPC-mean(3) stay below the target value with better energy consumption figures. The average queuing delay gap between MPC-mean(3) and dynamic timer-based coalescer rises up to 3.4 μs for larger target values. In all simulations, MPC-mean(3) resulted in larger energy consumption gains than MPC-mean(2) except for one single instance corresponding to trace 2 when $\tau = 8\mu s$ for which MPC-mean(2) slightly outperforms MPC-mean(3)

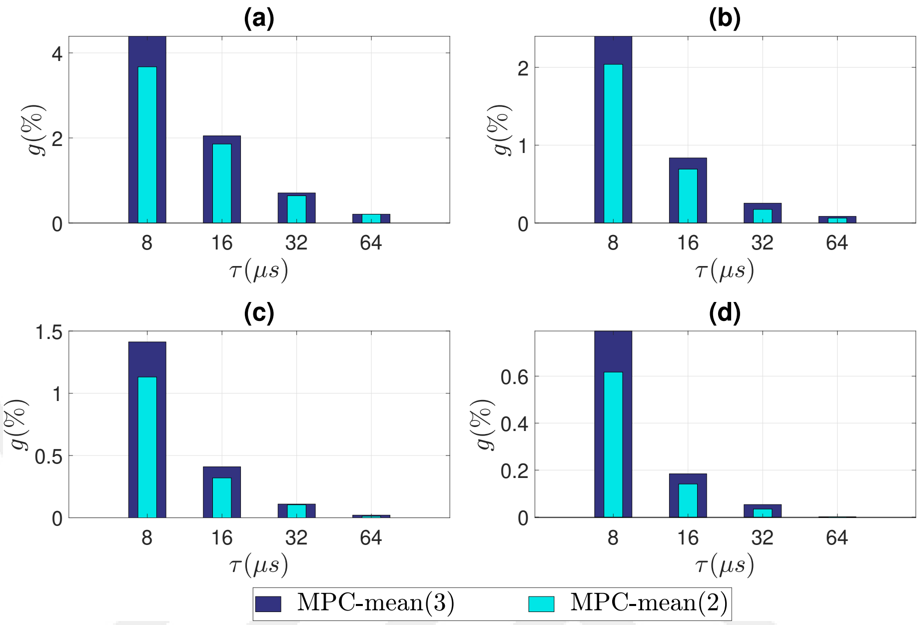

Figure 3.4: The percentage gain, g, in using MPC-mean against dynamic timer-based coalescing [1] for Pareto traffic with utilization factors (a) $\rho = 0.2$, (b) $\rho = 0.4$, (c) $\rho = 0.6$ and (d) $\rho = 0.8$.

Table 3.3: Average queuing delays (μs) achieved by dynamic timer-based coalescing [1] (DTC) and MPC-mean coalescing algorithms for actual traffic traces

Trace	Coalescing	$\tau = 8$	$\tau = 16$	$\tau = 32$	$\tau = 64$
1	DTC [1]	8.093	17.012	33.291	66.010
	MPC-mean(2)	7.902	15.777	31.089	62.844
	MPC-mean(3)	7.666	15.612	31.224	62.596
2	DTC [1]	8.060	16.432	32.437	64.730
	MPC-mean(2)	7.791	15.803	31.536	63.600
	MPC-mean(3)	7.765	15.742	31.635	63.347
3	DTC [1]	7.437	15.543	31.481	63.689
	MPC-mean(2)	7.389	15.390	31.492	63.791
	MPC-mean(3)	7.374	15.432	31.476	63.738

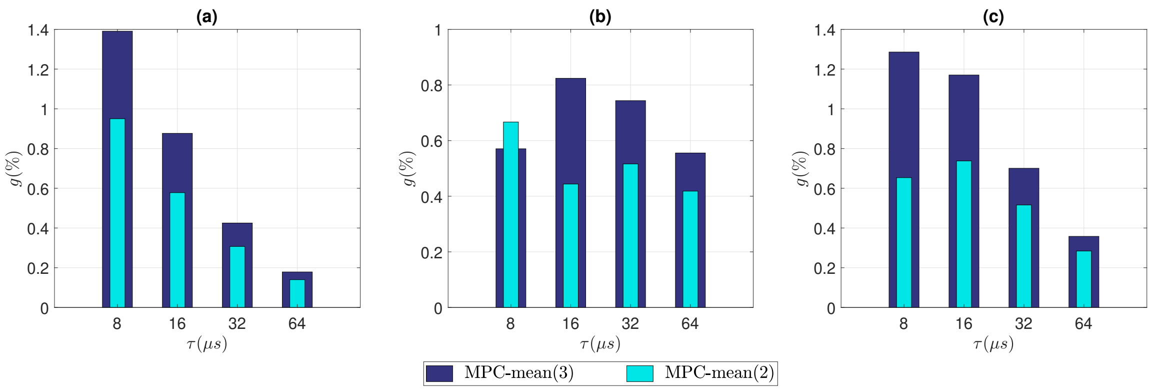

Figure 3.5: The percentage gain, g, in using MPC-mean against dynamic timer-based coalescing [1] using actual traffic traces (a) trace 1, (b) trace 2 and (c) trace 3.

with less than 0.1% gain.

3.6 Chapter Summary

We introduce a novel open-loop adaptive frame coalescing mechanism grounded in Model Predictive Control (MPC) and queuing theory. Unlike conventional coalescers which determine the timer parameter only once per coalescing cycle, the proposed, MPC-mean(M) updates the residual timer at $M \geq 1$ distinct time points within a single cycle. This adaptive strategy is designed to maintain the average queuing delay below a predefined threshold. Through extensive simulations using both synthetic and real-world traffic traces, MPC-mean demonstrated consistent gains in energy efficiency, achieving energy savings of up to 4.39%—notably under Pareto traffic with MPC-mean(3)—while also offering enhanced control over average queuing delays compared to existing open-loop dynamic timer-based coalescer in the literature.

Chapter 4

MPC-tail

We extend our study on MPC-based coalescing in EEE to control the tail of the queuing delays. Similar to MPC-mean, the proposed MPC-tail scheme functions as an open-loop controller that updates the timer parameter multiple times within a single coalescing cycle, considering actual arrivals observed prior to each update. In MPC-tail, an upper bound is imposed on the tail of the queuing delay distribution using queuing models based on M/PH/1 or M/ME/1 systems, where frame arrivals follow a Poisson process and service times are modeled using Phase-Type (PH) or the more general Matrix-Exponential (ME) distributions. For MPC-tail, the delay violation probabilities are influenced by the shape of the service time distribution. Therefore, we employ a mixture of Concentrated Matrix Exponential (CME) distributions [16] to more accurately represent Ethernet frame size distributions. The CME-based model facilitates numerical evaluation of the delay tail using the M/ME/1 framework, and the optimal timer parameter is obtained through a simple search algorithm.

Our results regarding MPC-mean and MPC-tail have been published in IEEE Transactions on Green Communications and Networking [9].

The chapter is organized as follows. We describe the problem definition and contributions in Section 4.1. We provide the system model in Section 4.2. We introduce the details of the proposed MPC-mean in Section 4.3. Section 4.4 presents extensive numerical results. Finally, we conclude the chapter in Section 4.5.

4.1 Problem Statement and Contributions

In the proposed MPC-tail scheme, an upper bound is enforced on the tail of the waiting time distribution using a queuing model which is based on M/PH/1 or M/ME/1 queues where the frame arrival process within the coalescing cycle is modeled as a Poisson arrival process and service times are modeled with PH-type (Phase Type) or the more general ME (Matrix Exponential) distributions, respectively. The reason for the use of PH-type or ME distributed service times for residual cycle analysis is three-fold. First, PH-type and ME distributions are very general and they cover the well-known exponential, hyper-exponential, Erlang, Coxian, etc. distributions as sub-cases. Secondly, closed-form expressions for the delay violation probability can be obtained as a function of the characterizing matrices of the underlying PH-type distribution. Moreover, queuing models developed for PH-type models for service times can also be used without a modification, for ME-distributed service times [74]. Finally, with a novel algorithm we propose, we fit an ME distribution to empirical service times, which makes it possible to estimate the delay violation probabilities very accurately. To the best of our knowledge, MPC-tail is the first attempt to adaptively control the delay violation probability for 10 Gb/s (and faster) EEE interfaces. One can refer to Section 3.1 for a brief review of analytical models and frame coalescing algorithms for EEE.

4.2 System Model

For the analysis of the residual cycle, the frame arrival process is assumed to be Poisson with rate λ. The frame service times are assumed to have PH-type distributions with the characterization pair (ς, S), mean service rate μ and variance σ^2. Obviously, the utilization factor (or load) $\rho = \lambda/\mu < 1$ must be satisfied for stability. A sample path of workload (unfinished work in time units) for EEE with coalescing is illustrated in Fig. 4.1 in which the workload at the interface is plotted as a function of time for the duration of a single coalescing cycle. In this example, abrupt jumps represent frame arrivals and the amount of jumps are the service times needed to process and transmit the frame which is a function of the frame size and the interface speed. When the buffer empties, the EEE interface makes a transition to the LPI mode after staying in the Sleep state for a fixed duration denoted by T_S. The interface remains in the LPI mode for a random duration of T_L. When the wake-up condition for the EEE coalescing algorithm is met, the interface abandons the LPI mode and, after a fixed duration of T_W in the Wake-up state, it starts to transmit frames in the Active mode using FCFS (First Come First Serve) queuing until the buffer empties again, for a random duration of T_A. This process repeats itself. Note that T_e represents the random duration for which the coalescing buffer is empty at the beginning of coalescing cycle.

4.3 MPC-Based Coalescing for Tail of the Queuing Delay

An MPC instant is one where the buffer is non-empty and the interface is in either the LPI or Sleep states. At such an instant, the number of arrivals that have accumulated in the buffer is denoted by $N > 0$. The service time introduced by the n^{th} arrival

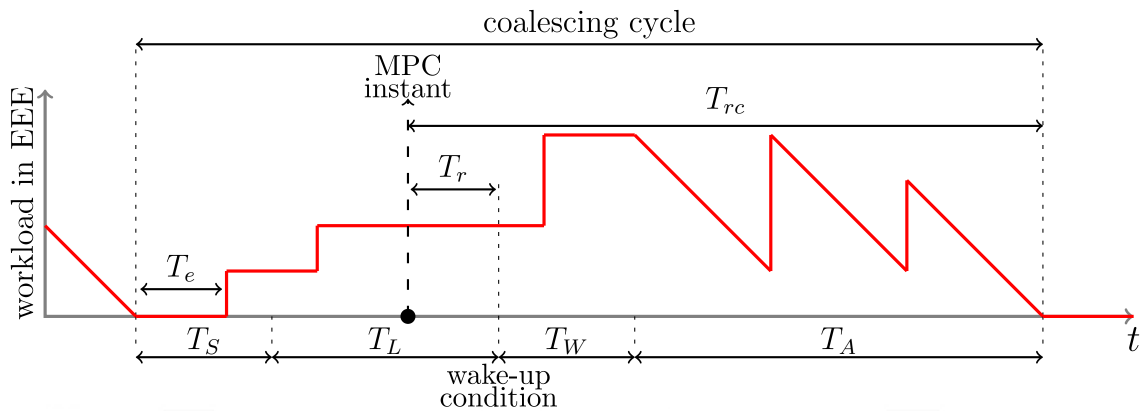

Figure 4.1: Sample path of the workload (unfinished work in time units) for EEE with MPC-tail.

and its queue wait until the MPC instant are denoted by s_n and $q_n, n = 1, 2, \ldots, N$, respectively. Then, the total accumulated workload is given by

$$w = \sum_{n=1}^{N} s_n. \tag{4.1}$$

Let T_r denote the time to transition to the W state at the current MPC instant. A sample MPC instant and the corresponding T_r duration is depicted in Fig. 4.1.

For a given value of T_r, the total queue wait that will be experienced by the n^{th} arrival is a deterministic value that is given by

$$Q_n = T_r + a_n, \tag{4.2}$$

where

$$a_n = \begin{cases} q_n + T_W, & \text{if } n = 1, \\ q_n + T_W + \sum_{l=1}^{n-1} s_l, & \text{if } n > 1. \end{cases} \tag{4.3}$$

Next, we propose MPC-based frame coalescing scheme MPC-tail by utilizing the results of the residual cycle analysis and taking into account the arrivals that have occurred prior to the MPC instant of consideration. The expressions obtained for residual cycle analysis can then be used to attain a desired delay violation probability, by adjusting the parameter T_r.

4.3.1 MPC-tail

In this subsection, we obtain the timer parameter at an MPC instant by which the probability that the queue waiting time of an arrival that occurs within the entire cycle exceeding a threshold d (violation parameter) is bounded by a given value p (target violation probability). For this purpose, we first write

$$\mathbb{P}(W_c > d) = \frac{\sum_{n=1}^{N} \mathbb{1}_{Q_n > d} + \mathbb{E}[\Lambda]}{N + \mathbb{E}[N_r]}, \tag{4.4}$$

where $\mathbb{E}[\Lambda]$ is the expected number of frames with queuing delays greater than d within the residual cycle and

$$\mathbb{1}_{x>d} = \begin{cases} 1, & \text{if } x > d, \\ 0, & \text{if } x \leq d, \end{cases} \tag{4.5}$$

is the indicator function which tests the input argument against the threshold d. In order to obtain $\mathbb{E}[\Lambda]$, we assume that the service time $S \sim PH(\varsigma, S)$.

Theorem 2. *Consider an M/PH/1 queue model of the EEE link with Poisson arrivals with rate λ, PH-type service times characterized with the matrix pair (ς, S) with mean $1/\mu$ and load $\rho = \lambda/\mu < 1$. Also, let us be given an MPC instant during the LPI state at which we have a workload of w in the buffer and the transition to W state will take place in T_r seconds after the MPC instant. The expected number of frames within the residual cycle with queuing delays greater than d, denoted by $\mathbb{E}[\Lambda]$,* *equals:*

$$\begin{cases} \lambda \mathbb{E}[T_{rc}] \frac{T_r + T_W + w - d + \gamma(1 - e^{Lx})L^{-1}e}{T_r + T_W + w}, & \text{if } T_r \geq d - w - T_W, \\ \\ \lambda \mathbb{E}[T_{rc}] \frac{\gamma(1 - e^{L(T_r + T_W + w)})L^{-1}e}{T_r + T_W + w}, & \text{if } T_r < d - w - T_W, \end{cases} \tag{4.6}$$

where

$$\gamma = \lambda \varsigma S^{-1}, \ L = S + \lambda S e \varsigma S^{-1}. \tag{4.7}$$

and $\mathbb{E}[T_{rc}]$ *is given in (3.6).*

Proof. In order to derive $\mathbb{E}[\Lambda]$, we also need an expression for the distribution of $W_r(t)$ illustrated in Fig. 3.2, which is same as the queuing delay distribution of a single server system which takes fixed length vacations of T_Σ duration (T_Σ is provided in (3.9)). For this purpose, we assert the assumption of PH-type distributed service times characterized with the pair (ς, S). To obtain the distribution of $W_r(t)$, we make use of the decomposition property of M/PH/1 queuing system with randomized vacations which states that the Laplace-Stieltjes Transform (LST) of the queuing delay is in the following form

$$W_r(s) = F(s) W_{M/PH/1}(s), \tag{4.8}$$

where $W_{M/PH/1}(s)$ is LST of the queuing delay distribution of the standard M/PH/1 queuing system in which the server is always available and $F(s)$ arises due to the vacation model the system employs [71]. For a server which employs fixed-length vacations of T_Σ duration, $F(s)$ is given by

$$F(s) = \frac{1 - e^{-T_\Sigma s}}{T_\Sigma s}. \tag{4.9}$$

Identities provided in (4.8) and (4.9) can be interpreted as follows: W_r is the sum of two independent random variables, namely $W_{M/PH/1}$ and a uniform random variable in the interval $(0, T_\Sigma)$. In [75], the queuing delay distribution of the standard M/PH/1 queuing system is shown to be also PH-type distributed characterized with the pair (γ, L) that can be computed with (4.7) using ς and S. Then, for given $d > 0$, one can write the following identity for the complementary cumulative distribution function (CCDF) of the queuing delay of the auxiliary process

$$\mathbb{P}(W_r > d) = \begin{cases} \frac{A(T_\Sigma)}{T_\Sigma}, & \text{if } d \geq T_\Sigma, \\ \frac{T_\Sigma - d + A(d)}{T_\Sigma}, & \text{if } d < T_\Sigma, \end{cases} \tag{4.10}$$

where

$$A(x) = \int_0^x \gamma e^{Lt} e \, \mathrm{d}t = \gamma(1 - e^{Lx})L^{-1}e, \tag{4.11}$$

is the definite integral of the CCDF of the standard M/PH/1 queuing system, $\mathbb{P}(W_{M/PH/1} > t)$, over the interval $(0, x)$. From (3.6) and (4.10), the expected number of the frames that occur within the residual cycle with queuing delays greater than d is given by

$$\mathbb{E}[\Lambda] = \lambda \mathbb{E}[T_{rc}] \mathbb{P}(W_r > d), \tag{4.12}$$

which is equivalent to (4.6) and completes the proof of Theorem 2.

$\square$

To achieve $\mathbb{P}(W_c > d) = p$, the desired residual timer parameter, ${T_r}^*$, must satisfy the following equation

$$\begin{aligned} \sum_{n=1}^{N} \mathbb{1}_{{T_r}^* > d - a_n} - \gamma e^{-L({T_r}^* + w + T_W - d)} L^{-1} e - \frac{\lambda p}{1-\rho} {T_r}^* =& Np + \frac{\lambda}{1-\rho}(p(w + T_W) \\ & - \gamma e^{Ld} L^{-1} e), \end{aligned} \tag{4.13}$$

for the case when ${T_r}^* < d - w - T_W$. On the other hand, for the case ${T_r}^* \geq d - w - T_W$, the following equation needs to be satisfied,

$${T_r}^* + \frac{1}{\lambda} \sum_{n=1}^{N} \mathbb{1}_{{T_r}^* > d - a_n} = \frac{Np}{\lambda}(1-\rho) + \gamma L^{-1} e + d - w - T_W - \gamma e^{Ld} L^{-1} e. \tag{4.14}$$

Note that the right hand sides of both (4.13) and (4.14) consist of parameters that do not depend on ${T_r}^*$. We note that the left hand side of (4.14) is a monotonously increasing non-linear function of ${T_r}^*$. For (4.13), the left hand side is again a monotonously increasing non-linear function of ${T_r}^*$ when ${T_r}^*$ is larger than the critical point T_c of the following function,

$$c(x) = \gamma e^{-L(x+w+T_W-d)} L^{-1} e + \frac{\lambda p}{1-\rho} x, \tag{4.15}$$

where T_c satisfies the following equation,

$$\gamma e^{-L(T_c+w+T_W-d)} e = \frac{\lambda p}{1-\rho}. \tag{4.16}$$

We propose to find this critical point T_c using bisection search. For the region ${T_r}^* \leq T_c$, the left hand side of (4.13) is a decreasing function in each interval that $\sum_{n=1}^{N} \mathbb{1}_{{T_r}^* > d - a_n}$ is constant and, makes an abrupt jump at the points where $\sum_{n=1}^{N} \mathbb{1}_{{T_r}^* > d - a_n}$ increases.

In light of these observations, a simple search algorithm is proposed to obtain ${T_r}^*$ for MPC-tail. First, we start by computing the left hand side of (4.13) with ${T_r}^* = T_c$. If the result is smaller than the right hand side, then ${T_r}^*$ can be found by employing the conventional bisection search algorithm for the region ${T_r}^* > T_c$. Otherwise, the left limits of the points at which the function $\sum_{n=1}^{N} \mathbb{1}_{{T_r}^* > d - a_n}$ makes the abrupt jumps are sorted in a descending manner and, one by one, the left hand sides of (4.13) are computed with those left limits. The greatest left limit that results in a left hand side that is smaller than the right hand side of (4.13), is to be selected as ${T_r}^*$. Note that attaining the target value d, i.e., $\mathbb{P}(W_c > d) = p$, is not always guaranteed due to the abrupt jumps occurring at the left hand sides of the expressions (4.13) and (4.14), but performance below the target value can be achieved.

4.3.2 Modeling Service Times by Mixture of CMEs

Similar to any other open-loop coalescing technique, approximated traffic parameters are also an input for the MPC-based coalescer. The approximations related to interarrival and service time distributions can be made at the update instants as we already have acquired the information of the arrivals waiting in the buffer. We only need the mean and variance approximations of the service and interarrival times and we propose to update these approximations at an update instant with the Robbins–Monro algorithm with update coefficient β [72]. For MPC-tail, an algorithm that fits a PH-type or an ME distribution to the empirical service times is also needed. It is known that the queuing results obtained with PH-type distributions are also applicable when PH-type distributions are replaced with ME distributions as shown in [34–36]. Therefore, residual timer computations given in (4.13) and (4.14) are valid both for PH-type and ME distributions. In order to achieve lower orders for the resulting random variable, we propose an offline heuristic algorithm that mixes CME random variables to obtain an ME distribution characterized with the pair (ς, S).

The proposed algorithm runs on J frames to model the service time distribution. The maximum service time of a frame is denoted by s_{max}. The interval $(0, s_{max})$ is divided into I sub-intervals with equal lengths of $\Delta = \frac{s_{max}}{I}$. These sub-intervals are then sorted according to the number of frames that belong to them and each of the first $K-1$ sub-intervals with largest number of frames are identified as main bins, and the rest of the intervals are combined into a single bin that keeps all residual frames which is indexed as the K^{th} bin. The number of frames that belong to the k^{th} bin is denoted by J_k and the ratio of the frames that belongs to a bin to the overall number of frames is given by $r_k = \frac{J_k}{J}$. The sample mean and variance for the k^{th} bin are denoted by $\bar{m}_k$ and $\bar{\sigma}_k^2$, respectively. In order to fit an ME distribution to the empirical service times, each bin is modeled as a CME distribution characterized with the matrix pair (ς_k, S_k). It is essential that the order of the CME distribution for bin k needs to be chosen such that its mean and variance match the sample mean and variance, respectively, for the same bin. It is also important to take into account the increased computational complexity brought by choosing very large CME orders. As previously stated, the SCV of the CME distribution with order l can be approximated by $2/l^2$. Therefore, we propose to set the mean and order of the k^{th} CME distribution, to $\bar{m}_k$ and l_k, respectively, where the latter is given by

$$l_k = \min\left(l_{max}, \lfloor \frac{2\bar{m}_k}{\bar{\sigma}_k} \rfloor\right), \tag{4.17}$$

where l_{max} is an upper limit that is selected to restrict the computational load and the operator $\lfloor \cdot \rfloor$ stands for the flooring operation. We propose to use the algorithm proposed in [16] that obtains the matrix pair (ς_k, S_k) of the k^{th} CME distribution with order l_k for $l_k \leq 47$ and the case of $l_k \in \{0, 1\}$ amounts to an exponential distribution. Then, these CME distributions are mixed to form a multi-modal ME distribution characterized with the matrix pair (ς, S) where

$$\varsigma = \begin{bmatrix} r_1\varsigma_1 & \cdots & r_K\varsigma_K \end{bmatrix}, \quad S = \begin{bmatrix} S_1 & & \\ & \ddots & \\ & & S_K \end{bmatrix}. \tag{4.18}$$

The algorithm presented above is to be run offline to obtain the pair (ς, S) since the frame size distributions on the Internet do not change rapidly in time. However, in case of need, it can occasionally be re-run to capture the daily or seasonal trends of varying frame size distributions. The computational complexity of the off-line frame classification and per-bin sample mean and variance calculations is $O(J)$, and interval sorting is $O(I \log I)$. On the other hand, the choice of the parameters K and l_{max} directly affect the order of resulting ME distribution, which is then input to each of the bisection search iterations employed by MPC-tail.

4.3.3 MPC Instantiation

MPC instantiation of MPC-tail is similar to that of MPC-mean described in 3.4.1.

We propose a simple MPC instantiation algorithm with the number of instants within a cycle bounded by the parameter M. In the proposed algorithm, the first MPC instant, T^1_{MPC}, is immediately after the first arrival of the coalescing cycle by considering the recently introduced workload and naturally, $N = 1$, $q_1 = 0$ and w being the service time of this first frame of the cycle. Also let T_r^{*m} denote the timer parameter found at instant $m, m = 1, \ldots, M$. We propose to set the subsequent MPC instants T^m_{MPC} through the following iterative procedure:

$$T^m_{MPC} = T^{m-1}_{MPC} + 0.5T_r^{*(m-1)}, \quad 1 < m \leq M. \tag{4.19}$$

We propose to transition to the W state at time $T^M_{MPC} + T_r^{*M}$. However, when a value $T_r^{*m} > 0$ cannot be found at instant $m \leq M$, then we immediately transition to the W state by terminating the iterations. The MPC-tail algorithm using the MPC instantiation parameter M are denoted by MPC-tail(M). In the numerical results, we obtain results for $1 \leq M \leq 2$.

4.3.4 Remarks on Implementation

For MPC-tail, control of the tail for queuing delays comes at a cost of bisection search at each MPC instant to obtain both the critical value, T_c, and the desired timer parameter, T_r^* through the calculation of (4.13), (4.14) and (4.16) repetitively depending on the search range and the desired accuracy of the outcome. The computational cost of each iteration is directly affected by the order of the ME distribution used for frame sizes which is at most Kl_{max}. The advantage of using a mixture of CMEs (as opposed to using conventional, such as Erlang, distributions) reduce the computational cost by achieving the same SCV values but with far lower orders.

The computational burden of the offline ME fitting algorithm is not included in the per-cycle load introduced by the online MPC-tail algorithm. Moreover, the matrices e^S, e^{-S} and S^{-1} can be computed and stored offline. By doing so, the matrix exponentiation and inversion operations for (4.13), (4.14) and (4.16) at each iteration can be avoided, and the online operations can hence be reduced to basic matrix operations (matrix-matrix and matrix-vector multiplications) with known computational complexity of at most $O(K^3\ l_{max}^3)$ since the size of the matrices are at most Kl_{max}. However, further investigation of how the parameters K and l_{max} are to be chosen so that MPC-tail can be implemented in real systems at the interface rate, is left for future research.

The energy cost of updating the timer at multiple instants for MPC-based coalescing is negligible since timer adaptation mechanisms for EEE only require software modifications on the NIC driver side, as also indicated in 3.4.1.

4.4 Numerical Examples

We first present performance results for MPC-tail(M) using simulations with static Poisson traffic and then Pareto interarrival times with shape parameter $\alpha = 2.5$, where the latter model was used to capture the self-similar nature of Ethernet traffic [1]. In these simulations, the frame sizes are assumed to have a PH-type bi-modal distribution which is obtained by mixing the Erlang(10,100) and Erlang(10,1500) distributions with probabilities 0.3 and 0.7, respectively, where Erlang(k,v) stands for an Erlang distribution with order k and mean size v in units of bytes. Then, we evaluate the proposed techniques with MAWI frame traces captured at the WIDE backbone Ethernet link samplepoint-G [73]. Throughout the simulations with actual traffic traces, we utilized the approximation and fitting algorithms described before

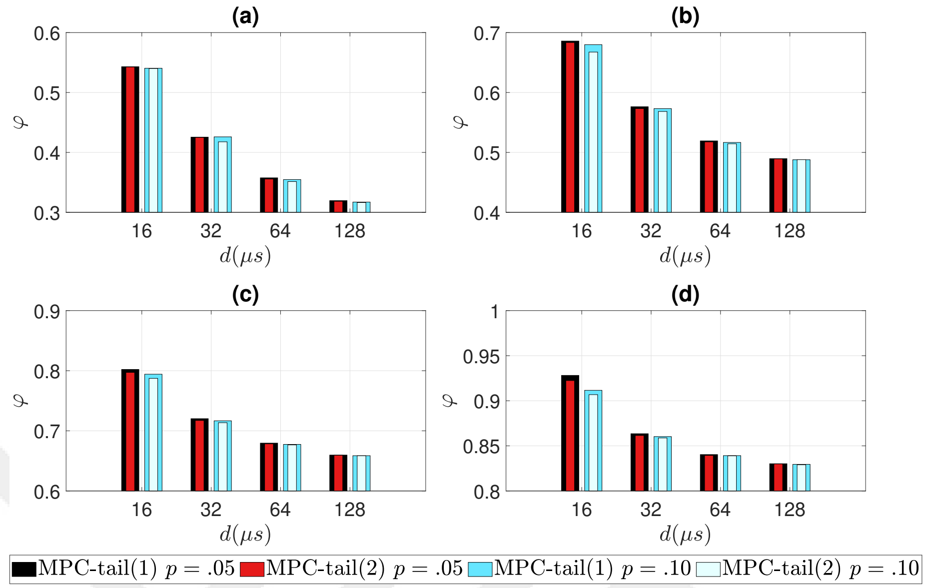

Figure 4.2: The ratio φ of the overall energy consumption of an EEE interface with MPC-tail(M), $M = 1, 2$ to that of the non-EEE interface for Poisson traffic with utilization factors (a) $\rho = 0.2$, (b) $\rho = 0.4$, (c) $\rho = 0.6$ and (d) $\rho = 0.8$.

to approximate the parameters dynamically. The update coefficient for the Robbins-Monro algorithm is chosen to be $\beta = 0.001$. The information about the three traffic traces indexed from 1 to 3 (when they are taken, average utilization factors, etc.) that are used in the simulations are provided in Table 3.1.

We only study the performance of MPC-tail(1) and MPC-tail(2) since increasing M further would introduce additional computational load due to the search algorithm introduced in Section 4.3.1. To the best of our knowledge, MPC-tail is the only dynamic open-loop coalescer that is proposed in the literature to control the tail of queuing delays. Therefore, the energy consumption of MPC-tail is compared against the non-EEE scenario using the ratio φ provided in (2.3). Throughout the simulations, we focused on the 90^{th} and 95^{th} percentile tail targets, which means p is chosen to be 0.10 and 0.05, respectively.

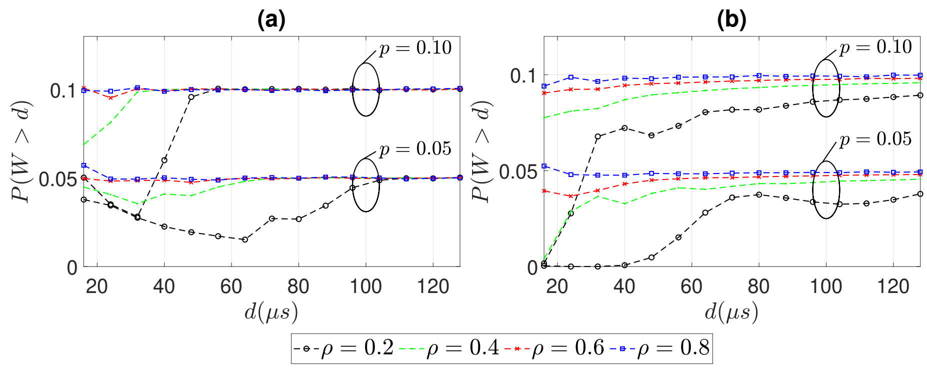

Figure 4.3: $\mathbb{P}(W > d)$ as a function of d when (a) MPC-tail(1) (b) MPC-tail(2) is employed for simulations under Poisson traffic.

Probability of the arrivals that wait longer than d is plotted in Fig. 4.3 when MPC-tail is employed for Poisson traffic for different values of d, p and λ. As previously stated, due to the abrupt jumps of the left hand sides of (4.13) and (4.14), the target violation probabilities are not always attainable at each coalescing cycle. Therefore, $\mathbb{P}(W > d)$ occasionally deviates from p. Nevertheless, we have observed that MPC-tail succeeded in keeping the delay violation probability below the target. Moreover, the delay violation probability gets closer to p as d increases. The energy consumption ratios depicted in Fig. 4.2 indicate that employing MPC-tail(2) may result in energy consumption gains up to 1.81% (under $\rho = 0.4$ with $d = 16\mu s$ and $p = 0.10$) compared to MPC-tail(1).

For Pareto traffic, Fig. 4.4 shows that MPC-tail(2) results in energy consumption gains up to 2.12% (under $\rho = 0.4$ with $d = 16\mu s$ and $p = 0.10$) when compared with MPC-tail(1). Additionally, we observe from Fig. 4.5 that MPC-tail(2) brings the delay violation probability closer to p compared to MPC-tail(1).

Finally, we investigated MPC-tail when actual traffic traces are used in our simulations throughout which we ran the fitting algorithm that was described in subsection 4.3.2 in order to obtain the ME distributed approximation of service times. The performance of the fitting algorithm is depicted in Fig. 4.6 where we plot the

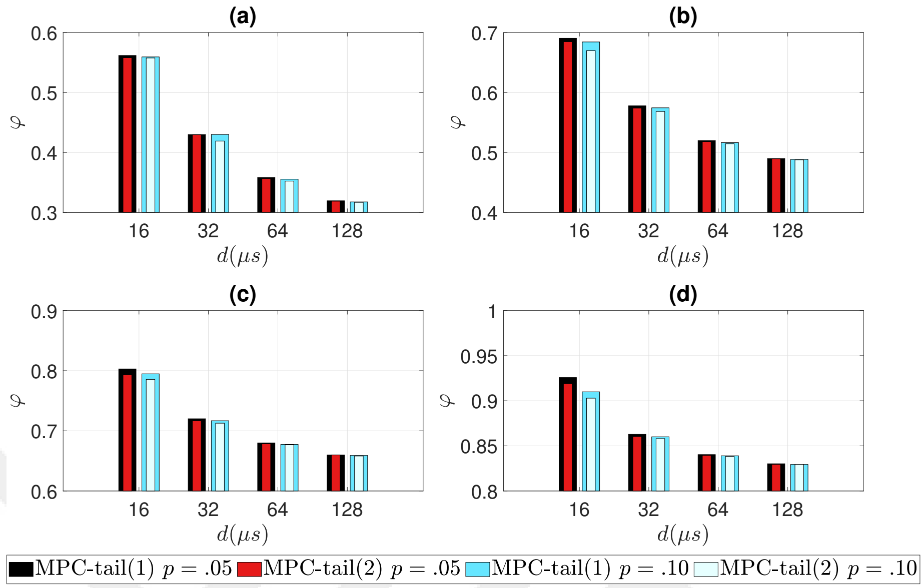

Figure 4.4: The ratio φ of the overall energy consumption of an EEE interface with MPC-tail(M), $M = 1, 2$ to that of the non-EEE interface for Pareto traffic with utilization factors (a) $\rho = 0.2$, (b) $\rho = 0.4$, (c) $\rho = 0.6$ and (d) $\rho = 0.8$.

empirical CDF obtained from service times of 1000 frames and the resulting ME distributions when the fitting algorithm uses the frame count parameter $J = 1000$. We select the number of intervals $I = 100$ and obtain results for four different pairs of order upper bounds, l_{max}, and number of bins, K. For the service time distribution of Ethernet frames, one can utilize the fitting algorithm by selecting $K = 3$ with two main bins which is particularly suitable for bi-modal distributions. However, improved fitting is possible with K and l_{max} increasing, but at the expense of increased ME orders. Table 4.1 depicts the order of the resulting ME distribution for various choices of the K and l_{max} parameters. The parameter selection of the ME fitting algorithm may depend on computational load limitations and accuracy requirements of MPC-tail. Throughout the numerical experiments, we used the fitting algorithm with parameters $K = 4$ and $l_{max} = 15$, and the offline algorithm is rerun over the first thousand of every one million frames to capture the service time distribution.

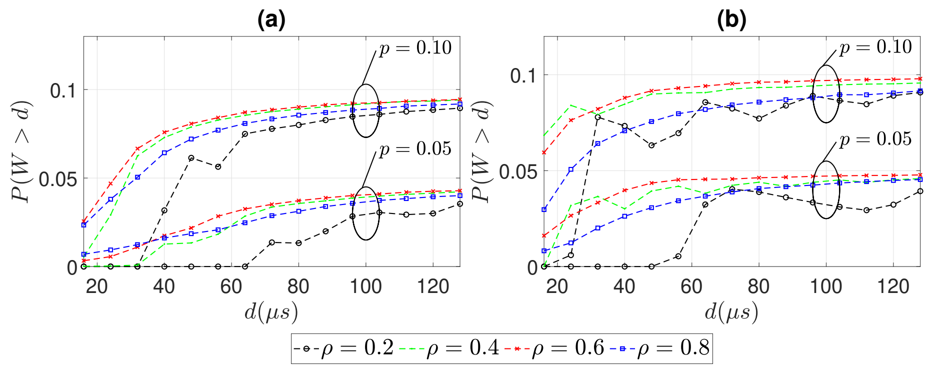

Figure 4.5: $\mathbb{P}(W > d)$ as a function of d when (a) MPC-tail(1) (b) MPC-tail(2) is employed for simulations under Pareto traffic.

Table 4.1: Order of the resulting ME distributions with the frame size fitting algorithm for different algorithm input parameters.

K	l_{max}	Order of the ME distribution
3	15	30
4	15	54
4	60	142
20	15	326

The energy consumption ratios that are plotted in Fig. 4.7 reveal that employing MPC-tail(2) gives rise to gains up to 1.40% (for trace 3 with $d = 16\mu s$ and $p = 0.05$) when compared to MPC-tail(1). Also, we observe from Fig. 4.8 that the probability $\mathbb{P}(W > d)$ exceeds p in some scenarios when MPC-tail(1) is in action whereas MPC-tail(2) manages to keep the same probability below the threshold in all the scenarios we studied. Overall, the simulations with actual traffic traces verify the effectiveness of the MPC-tail coalescers along with the service time distribution fitting algorithm we have proposed. Furthermore, the fact that ME distribution that is obtained with first 1000 frames being effective in controlling one million frames, proves that frame size behavior does not change rapidly for real traces and the effectiveness of CDF fitting even when it is run offline.

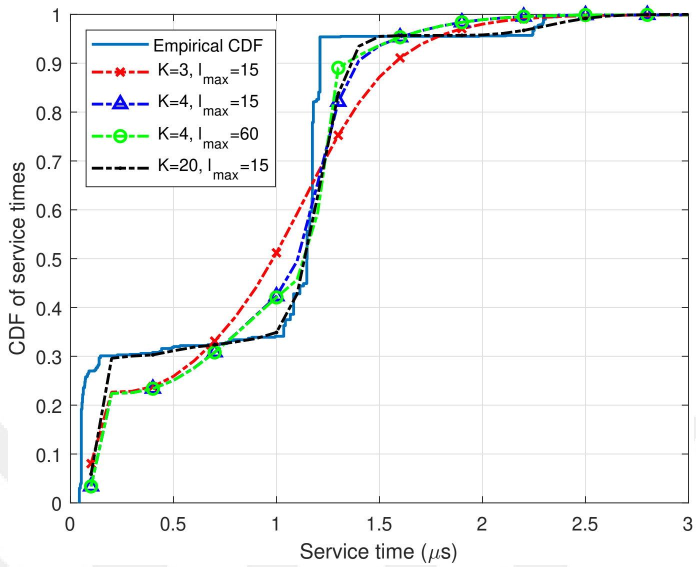

Figure 4.6: Empirical CDF of the actual trace and the obtained ME distributions using the frame size fitting algorithm.

4.5 Chapter Summary

We extend our study on MPC-based coalescing in EEE to control the tail of the queuing delays. Proposed MPC-tail(M) updates the residual timer parameter at $M \geq 1$ separate instants throughout a single coalescing cycle as opposed to determining it only once, a method used in conventional coalescers. The purpose of MPC-tail is to keep the tail of the queuing delay below a predefined threshold. The simulation results reveal that MPC-tail can effectively control the tail of the queuing delays especially with the MPC-tail(2) algorithm, with energy consumption gains up to 2.12% compared to MPC-tail(1). Furthermore, the proposed CME fitting algorithm for Ethernet frames is shown to effectively approximate the queuing delay distributions.

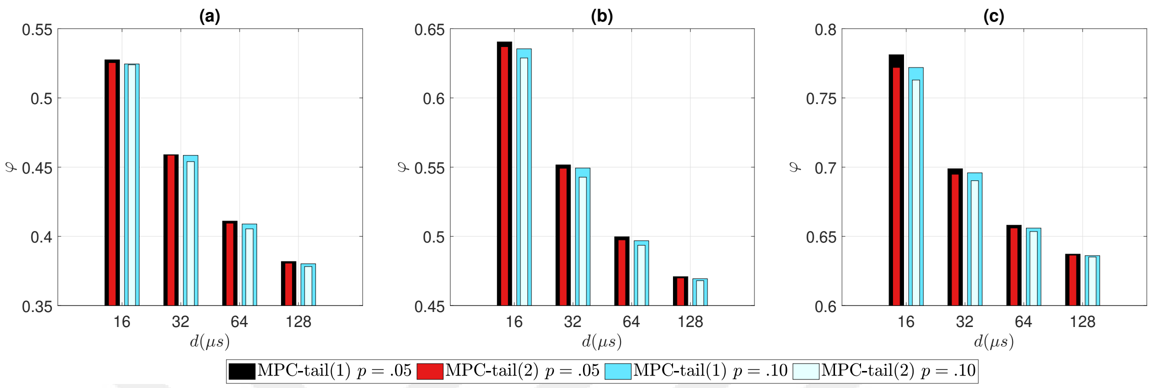

Figure 4.7: The ratio φ of the overall energy consumption of an EEE interface with MPC-tail(M), $M = 1, 2$ to that of the non-EEE interface, (a) trace 1, (b) trace 2 and (c) trace 3.

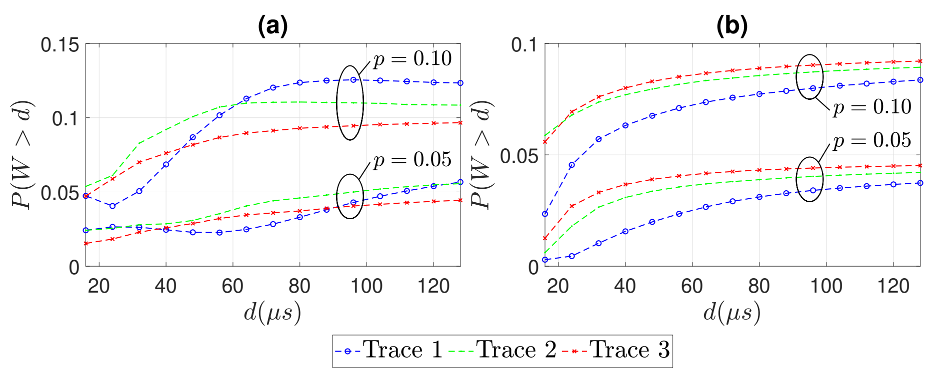

Figure 4.8: $\mathbb{P}(W > d)$ as a function of d when (a) MPC-tail(1) (b) MPC-tail(2) is employed for simulations under actual traffic traces.

Chapter 5

Energy-Age of Incorrect Information Trade-off with Timer-Based Sleep-Wake Scheduling

This chapter examines the trade-off between energy efficiency and information freshness, focusing on the transmission module of IoT sensor nodes that alternate between Sleep (or Low-power) and Active operational modes. Specifically, we study timer-based sleep-wake scheduling for a single sensor node, using a variant of the Age of Incorrect Information (AoII), referred to as AoII+ in the literature, as the metric for quantifying information freshness. We derive closed-form expressions for both the average AoII+ and average power consumption under the assumption that updates follow a Poisson arrival process, with service times modeled as either deterministic or phase-type distributions. Based on this analysis, we introduce two optimal scheduling strategies: the AoII-Idling Scheduler (AoII-IS) and the AoII-Zero Idling Scheduler

(AoII-ZIS), both designed to govern the transitions between Sleep and Active states. For these schedulers, we obtain closed-form solutions for the optimal timer values that minimize power usage while ensuring the average AoII+ remains below a specified threshold. Additionally, we propose adaptive variants of both schedulers that dynamically adjust their timer parameters under varying update rate over time.

Our results for this line of investigations have been accepted for presentation in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2025.

The chapter is organized as follows. We give a brief review of information freshness and energy trade-off in queuing systems in Section 5.1. We describe the problem definition and contributions in Section 5.2. We provide the system model in Section 5.3. We analyze the trade-off between AoII+ and energy consumption for timer-based sleep-wake scheduler in Section 5.4. We propose optimal and adaptive controllers constrained on AoII+ in Section 5.5. Section 5.6 presents numerical results. Finally, we conclude the chapter in Section 5.7.

5.1 A Brief Review of Information Freshness and Energy Trade-off in Queuing Systems

The energy trade-off attracts many of the recent research efforts on the information freshness. The authors in [76] considered the information freshness and energy trade-off from the sampler's perspective and studied the information freshness and energy trade-off in fading channels where strategies were designed to minimize the weighted sum AoI and energy consumption for sensors [77]. Energy consumption in information sensing has also been studied in [78], [79] and [80]. The authors in [81] study the energy efficiency by considering the adaptive control of both sampling and transmission of the source. Recent work of [82] studies the relation between energy and age in wireless networked control systems (WNCS) through a parameter defined as Energy Efficiency-PAoI (EE-PAoI) ratio. Similarly, Age-Energy Efficiency (AEE) metric is introduced in [83] to study the trade-off in industrial Internet of Everything (IoE) networks. The trade-off between energy and AoII is also investigated in the recent work of [84] from the sampler's perspective. Similar models to ours in discrete-time are investigated for Ao^{2}I in [85] and AoII+ in [86] without considering the energy trade-off.

There have been few attempts to consider the relation between information freshness and energy from the perspective of the sleep-wake server. The authors of [7] analyzed the relation between energy consumption and AoI for a single-server multi-source bufferless system under different sleep-wake strategies. Energy and PAoI trade-off was studied for an LCFS single-server single-source system in [10]. Also, the authors in [11] studied the relation between PAoI and energy consumption for a transmitter which utilizes DRX mechanism and proposed an optimal policy. The aforementioned works on the sleep-wake scheduling do not consider the trade-off of energy and freshness based on the information content since they solely focus on AoI. To the best of our knowledge, this is the first attempt to analyze and optimize the trade-off between energy consumption and age based on the actual content in information from the server's perspective for sleep-wake scheduling.

5.2 Problem Statement and Contributions

Reducing the energy consumption while satisfying Quality of Service (QoS) requirements is a topic of paramount importance for next generation wireless networks. Sleep-wake scheduling is one of the prominent mechanisms for energy efficiency which allows the node to save energy by temporarily turning off the components of its transmission module (transmitter or server), e.g., physical and medium access control layer components, when there are no packets to transmit.

Timer-based sleep-wake scheduling strategies are amenable to analysis and they are relatively easy to implement as a result of which they stand out among existing sleep-wake scheduling schemes. Various timer-based activation and sleeping schemes have been proposed using the tools of queuing theory [7]. Moreover, timer-based sleep-wake scheduling has been proposed for standard-based communication systems, for DRX for LTE [5] and EEE [6].

Recently, there has been increased interest in the relation between QoS metrics relying on information freshness and energy efficiency [82], [83], [84]. Only a fraction of these attempts have considered the relation between information freshness and energy from the perspective of the sleep-wake server. These studies are mostly based on AoI metric which fails to assess how long the information content has been outdated. For example, a monitor with a large AoI may still contain fresh information if the information content of the source has not changed. Moreover, the majority of previous studies consider static traffic conditions. However, IoT applications are often subject to varying update rates stemming from temporal changes in the environment.

A general information freshness metric is Age of Incorrect Information (AoII) which is defined as the product of a function which penalizes the time during which the source and the monitor are out of sync, and another penalty function that depends on both the current estimate at the monitor and the actual state of the input process [87]. AoII covers Age of Synchronization (AoS) as a sub-case where the latter is defined as the elapsed time since the content at the destination has been out of sync with the source [88–90]. For AoII and also AoS, the source and the monitor can be in sync without a reception of a status update packet which occurs when the original process transitions to the estimation value at the monitor, on its own. When the in-sync condition is incurred with a packet reception, then the receiver would exactly know that it is in sync with the source upon reception. However, when the in-sync condition occurs due to a transition of the source to the estimation value, the monitor would not know if state changes took place despite the in-sync condition. Moreover, the source will be aware of the in-sync condition but not the monitor. Therefore, we study a variation of AoII which is termed as AoII+ in [86] and Age of Outdated Information (Ao^{2}I) in [85]. AoII+ measures the elapsed time since the first time when the stored information at the destination becomes outdated in comparison to the source, but as a result of a status update packet generated by the source which is indicative of a state change. In other words, AoII+ is zero when there are no status update packets in the system. When a new status update is generated, AoII+ starts to increase with unit slope and can only be brought down to zero when the freshest update packet that carries the current state of the source is received by the monitor. For a sleep-wake scheduler that considers AoII+ as performance metric, a change in source state while in-sync will always trigger an update packet transmission. Since packet preemption is used by the source, the monitor would exactly know that its information is fresh when an update packet is successfully received. With this definition, AoII+ is also different than AoI since the age will not increase with time as long as the content currently stored at the monitor is not outdated or equivalently there are no status update packets in the system (see Fig. 5.2 for the sample path of AoII+). AoII+ has been proposed as a practical variation of AoII that is amenable to mathematical analysis and optimization. Furthermore, the independence of the AoII+ metric from the particular source dynamics and hence its simplicity similar to AoI, allow us to write closed-form expressions for average AoII+ and power consumption which facilitates the implementation of adaptive mechanisms which would be needed when changes occur in the environmental conditions.

In the proposed system, the sampler component of the IoT sensor node continuously observes a process and generates an update packet whenever a change occurs in the process; see Fig. 5.1. These update packets are transmitted by the transmitter component (or processed by the server) towards the monitor. The proposed sleep-wake strategy of the server is configured by two timers: (i) the activation timer is fired upon an incoming packet when the server is in the Low-power mode, to place a delay of T_a time units on the activation process, and (ii) the idling timer is fired as soon as the server becomes idle in the Active mode to delay the transition to the Low-power mode by T_i time units. Transitions from Low-power to Active modes of operation are incurred through a Setup mode which may involve operations such as the activation of radio frequency components, physical and medium access control layer link establishment mechanisms depending on the underlying communication protocol. We use preemption in waiting and preemption in service disciplines to preempt a packet in the system by a fresher one [91]. A successfully transmitted packet is assumed to be immediately acknowledged. The preemption and immediate acknowledgment mechanisms ensure that the source and the monitor are in sync when a packet is successfully received and the monitor is aware of the in-sync condition upon the reception of a packet.

For the proposed system, we first analytically derive expressions for average AoII+ and power consumption through a queuing model that is based on the M/D/1 or M/PH/1 preemptive queue where the packet arrival process is modeled by a Poisson arrival process and service times are deterministic or modeled by PH-type (Phase Type) distributions, respectively, and generally distributed setup times are employed. The reason for the use of PH-type service time is two-fold. First, PH-type distributions are very general and they cover the well-known exponential, hyper-exponential, Erlang, Coxian, etc. distributions as sub-cases. Secondly, this choice enables us to obtain closed-form expressions of AoII+ and energy consumption ratio which is defined as the ratio of power consumption of the sleep-wake scheduler to that of the server that is always in the Active mode. Subsequently, we propose an optimal scheduler, namely AoII-Idling Scheduler (AoII-IS), to select the optimal T_a and T_i in closed form in order to minimize the energy consumption while the average AoII+ is kept below a given desired threshold. We also propose additionally another optimal scheduler, namely AoII-Zero Idling Scheduler (AoII-ZIS), for the sub-case for which the idling timer parameter T_i is strictly forced to be zero. Note that the former scheduler type has recently been analyzed for peak AoI (PAoI) in a system that follows the Last Come First Served (LCFS) policy in [10], and the latter is widely used in the literature for its applications to standard-based mechanisms such as EEE [27].

The main contributions of the chapter are as follows.

- We derive expressions using renewal-type analysis for average AoII+ and energy consumption ratio for timer-based sleep-wake scheduling in continuous-time based on M/D/1 and M/PH/1 queuing models and a server employing a preemptive (both in waiting and in service) discipline.
- We propose two novel schedulers AoII-IS and AoII-ZIS which are shown to minimize the power consumption of the transmitter under a constraint on average AoII+. Moreover, we propose adaptive open-loop variations, namely AoII-Adaptive Idling Scheduler (AoII-AIS) and AoII-Adaptive Zero Idling Scheduler (AoII-AZIS) by which we update the optimal timer parameters under time-varying status update rates.
- We validate the analysis and the effectiveness of the proposed schedulers under various service and setup time distributions.

5.3 System Model

Table 5.1 summarizes key notations used in this chapter. We consider a status update system with a single information source which is served by a single server. Status update packets from the information source are generated according to a Poisson process with rate λ, and the transmission time (service or processing time), denoted

Table 5.1: Notations

Notation	Definition
λ	Update arrival rate
H	Service time distribution
h	Deterministic service time
α	Initial probability vector of PH-type distributed service time
A	Sub-generator matrix of PH-type distributed service time
G	Duration of busy period
T_s	Duration of setup time
T_a	Activation timer
T_i	Idling timer
T_e	Duration for which the system is empty when the server is in Low-power mode
T_{cycle}	Duration of a sleep-wake cycle
T_c	Total time elapsed while idling in the Active mode throughout a cycle
P_A	Power consumption of Active mode
P_L	Power consumption of Low-power mode
P_S	Power consumption of Setup mode
γ_L	Energy saving ratio of Low-power mode
γ_S	Energy saving ratio of Setup mode
$\Delta(t)$	AoII+ at time t
δ^*	AoII+ constraint for AoII-ZIS and AoII-IS
δ^*_{min}	Lower bound for AoII+ that is achieved when the server is always active
φ	Energy consumption ratio of timer-based strategy

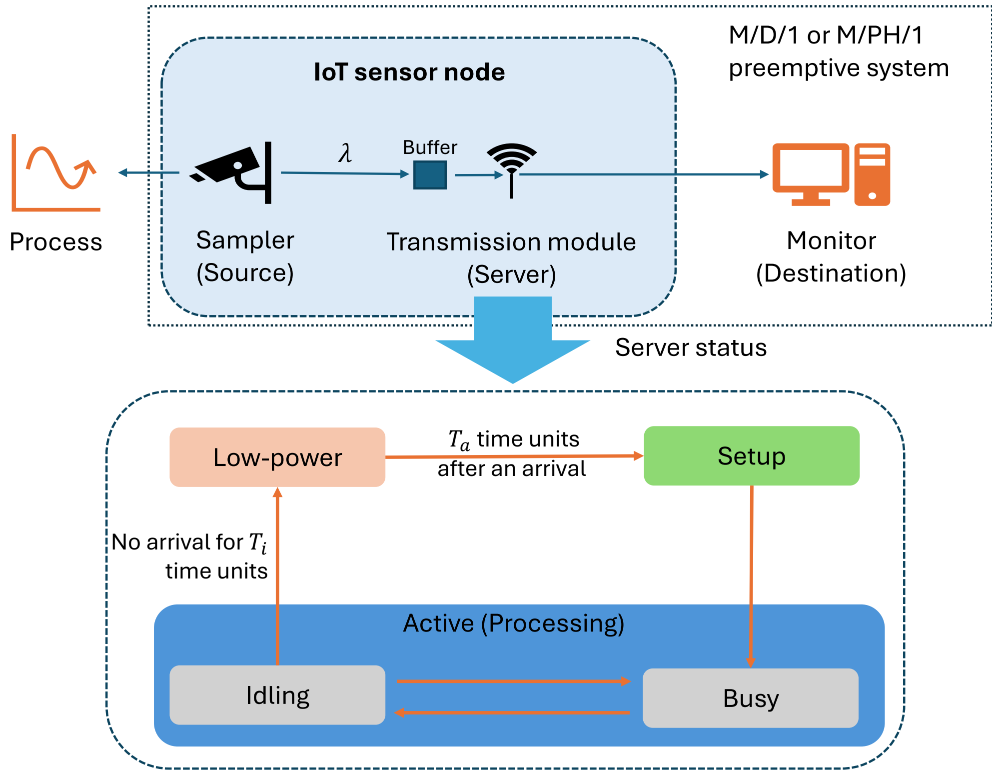

Figure 5.1: IoT sensor node with a transmission module which employs timer-based sleep-wake scheduling.

by H, for each update packet is i.i.d., and assumed to be either deterministic, or PH-type distributed characterized with the pair (α, A).

At a given time, the server operates in one of the following three modes: i) Active mode during which the server is able to process the packets in the system and consumes the highest power denoted by P_A, ii) Low-power mode in which the power consumption is reduced to the lowest, denoted by P_L, while packet processing is disabled, and iii) Setup mode which consumes moderate power, denoted by P_S and is incurred every time a transition is made from the Low-power to the Active mode of operation. The energy saving ratio for the low-power mode (resp. setup mode) is denoted by $\gamma_L = 1 - P_L/P_A$ (resp. $\gamma_S = 1 - P_S/P_A$). We assume that $0 \leq \gamma_S \leq \gamma_L \leq 1$.

The system employs a preemptive in service discipline in the Active mode of operation for which if a new packet arrival occurs while another packet is in service, the packet in service is immediately preempted by the new one. Note that the buffer of the system can only be non-empty when the server is in Low-power or Setup modes during which an incoming packet arrival replaces the one waiting in the buffer, i.e., preemption in waiting.

The duration in which the server is busy with processing packets continuously in the Active mode is denoted by G. We refer to this duration as the busy period. Note that with this definition, G amounts to the total duration elapsed while serving a packet which is processed without preemption and additionally its predecessors which are preempted consecutively.

We define a sleep-wake cycle as the period between two subsequent instants at which the server enters the Low-power mode. The sample paths of AoII+ are illustrated for two different timer configurations with non-zero and zero idling timers in figures 5.2(a) and 5.2(b), respectively. In these figures, the arrival instants are indicated with downward arrows with dashed ones corresponding to preempted arrivals. For Fig. 5.2(a), once the system becomes empty, the server would idle in Active mode for a deterministic time (denoted by T_i) before transitioning to the Low-power mode. If there is any arrival before the idling timer expires, the server will be restarted immediately upon the packet's arrival. In the given sample path only a single interrupted idling period is illustrated. The sample path in Fig. 5.2(b) differs from the former by not utilizing an idling period. In other words, T_i is selected to be zero and the server makes a transition to the Low-power mode immediately after a busy period. For both sample paths, once the server enters the Low-power mode, it remains in this mode as long as the system is empty. A deterministic activation timer T_a is triggered immediately after a packet arrives at the system when the server is in the Low-power mode, and a transition to the Active mode via a Setup mode is initiated when the timer expires. The duration of the Setup mode T_s is assumed to have a general distribution. T_e represents the exponentially distributed duration for which the system is empty when the server is in the Low-power mode. Note that the timer configuration of Fig. 5.2(a) is not feasible for AoII-ZIS and AoII-AZIS which are schedulers that strictly force the idling timer to zero.

5.4 Analysis of the Energy-Age Trade-off

In this section, for the proposed system model, we aim to derive the expressions for average power consumption and average AoII+.

5.4.1 Moments of the Busy Period

In this subsection, we derive the expressions for the first and second moments of the busy period, G, when the service time H is deterministic or PH-type distributed.

For the case when H is deterministic, i.e., $H = h$, the expected value of G is given by,

$$\mathbb{E}[G] = \mathbb{E}[N_{T_e \leq h}]\mathbb{E}[T_e | T_e \leq h] + h, \tag{5.1}$$

where $N_{T_e \leq h}$ stands for the number of preempted packets consecutively before a packet is completely processed. Actually, $N_{T_e \leq h}$ follows a geometric distribution with parameter $\mathbb{P}(T_e > h) = e^{-\lambda h}$. Hence,

$$\mathbb{E}[N_{T_e \leq h}] = \frac{1 - e^{-\lambda h}}{e^{-\lambda h}}. \tag{5.2}$$

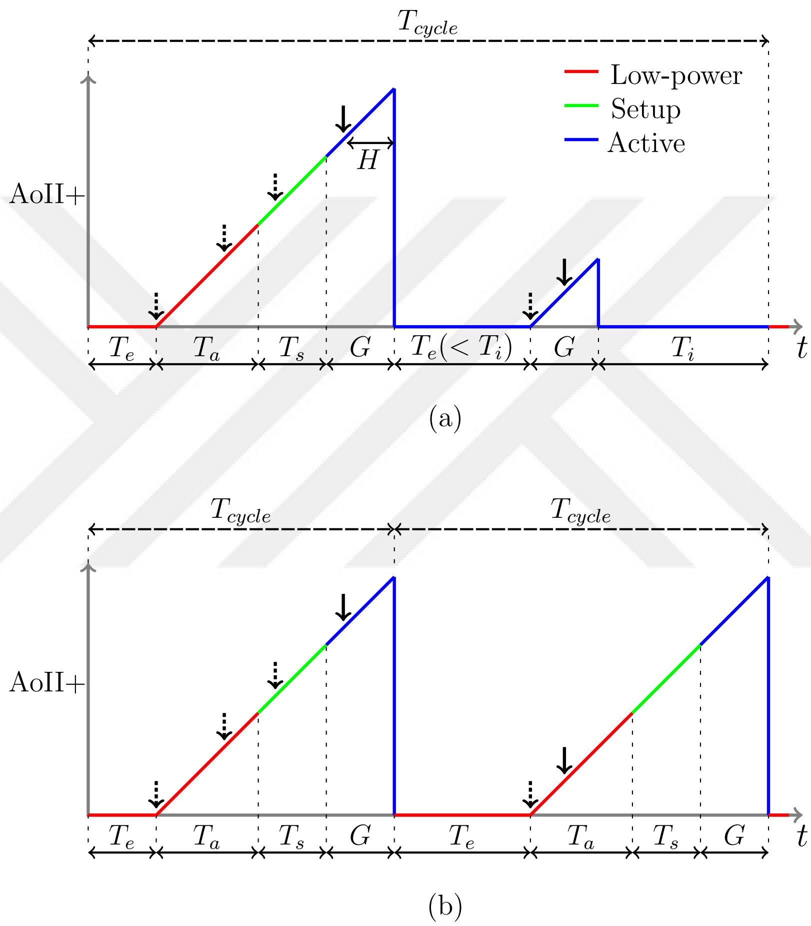

Figure 5.2: Sample path of AoII+ for a server which employs timer-based scheduling with (a) non-zero idling timer and (b) zero idling timer.

Also, $\mathbb{E}[T_e|T_e \leq h]$ denotes the conditional expectation of the exponentially distributed interarrival times which is given by,

$$\mathbb{E}[T_e|T_e \leq h] = \frac{1}{\lambda} - \frac{he^{-\lambda h}}{1-e^{-\lambda h}}. \tag{5.3}$$

Finally, $\mathbb{E}[G]$ is rewritten as

$$E[G] = \frac{1-e^{-\lambda h}}{\lambda e^{-\lambda h}}. \tag{5.4}$$

The second moment of G is given by

$$\mathbb{E}[G^2] = \mathbb{E}[N_{T_e \leq h}]\sigma^2_{T_e|T_e \leq h} + \mathbb{E}[N^2_{T_e \leq h}]\mathbb{E}[T_e|T_e \leq h]^2 + 2h\mathbb{E}[N_{T_e \leq h}]\mathbb{E}[T_e|T_e \leq h] + h^2, \tag{5.5}$$

where

$$\sigma^2_{T_e|T_e \leq h} = \frac{1}{\lambda^2} - \frac{h^2}{e^{-\lambda h} + e^{\lambda h} - 2}, \tag{5.6}$$

$$\mathbb{E}[N^2_{T_e \leq h}] = 1 + \frac{2 - 3e^{-\lambda h}}{e^{-2\lambda h}}. \tag{5.7}$$

On the other hand, when H is PH-type characterized with the matrix pair (α, A), the resulting busy period G is also PH-type characterized with the pair (α, A_G) where

$$A_G = A + \lambda e\alpha - \lambda I, \tag{5.8}$$

where I and e stand for the identity matrix and row vector of appropriate sizes, respectively. Note that the term $\lambda e\alpha - \lambda I$ in A_G resets the state of the service time upon an arrival. Consequently, the first and second moments of G which result from the PH-type distributed H are given by,

$$\mathbb{E}[G] = -\alpha A_G^{-1} e, \quad \mathbb{E}[G^2] = 2\alpha A_G^{-2} e. \tag{5.9}$$

Note that the first and second moments of G depend on the shape of service time distribution, and we only derived $\mathbb{E}[G]$ and $\mathbb{E}[G^2]$ for the deterministic and PH-Type distributed service time cases. However, the derivations in the following sections are written as functions of $\mathbb{E}[G]$ and $\mathbb{E}[G^2]$. Therefore, the following derivations are valid also for generally distributed G.

5.4.2 Energy Consumption

In this subsection, we derive the power consumption for the proposed timer-based sleep-wake scheduling. The sample path provided in Fig. 5.2(a) demonstrates that the duration of a sleep-wake cycle can be computed by summing the durations of the constituting random variables. Hence the expected duration of a sleep-wake cycle can be computed as follows,

$$\mathbb{E}[T_{cycle}] = \mathbb{E}[T_e] + T_a + \mathbb{E}[T_s] + \mathbb{E}[G] + \mathbb{E}[N_{T_e \leq T_i}](\mathbb{E}[T_e|T_e \leq T_i] + \mathbb{E}[G]) + T_i, \tag{5.10}$$

where $\mathbb{E}[N_{T_e \leq T_i}]$ and $\mathbb{E}[T_e|T_e \leq T_i]$ stand for the expected number of interrupted idling periods throughout a cycle and the expected duration of an interrupted idling period, which are given by the identities provided in (5.2) and (5.3), respectively, by replacing the term h by T_i. Subsequently, the expected duration of a sleep-wake cycle is rewritten as

$$\mathbb{E}[T_{cycle}] = T_a + \mathbb{E}[T_s] + (\mathbb{E}[N_{T_e \leq T_i}] + 1)(1/\lambda + \mathbb{E}[G]). \tag{5.11}$$

Finally, the energy consumption ratio φ (ratio of the power consumption of a server that utilizes timer-based sleep-wake scheduling to the power consumed by an always active server that does not support the Low-power mode) is given by,

$$\varphi = 1 - \frac{\gamma_L(1/\lambda + T_a) + \gamma_S \mathbb{E}[T_s]}{\mathbb{E}[T_{cycle}]}. \tag{5.12}$$

5.4.3 AoII+ Analysis

In this section, we derive the average AoII+ of a server that utilizes the timer-based sleep-wake scheduling. The AoII+ at time t is denoted by $\Delta(t)$. The average AoII+ is then given by the following expression,

$$\mathbb{E}[\Delta] = \lim_{t \to \infty} \frac{\int_0^t \Delta(\tau)\, \mathrm{d}\tau}{t}. \tag{5.13}$$

Since the AoII+ of a system with the sleep-wake scheduler is a regenerative process which renews at the beginning of each sleep-wake cycle, we can alternatively write

$$\mathbb{E}[\Delta] = \frac{\mathbb{E}[\int_0^{T_{cycle}} \Delta(\tau)\,\mathrm{d}\tau]}{\mathbb{E}[T_{cycle}]}. \tag{5.14}$$

The process $\Delta(t)$ starts to increase when an arrival occurs at an empty system. The server can either be in the Low-power mode or idling in the Active mode when the system is empty. If the arrival that triggers the increase of AoII+ occurs when the server is idling in the Active mode, then the resulting AoII+ rise will only last for G units upon which it drops down to zero. For the case when an arrival occurs to an empty server in the Low-power mode, the resulting AoII+ rise will last for $T_a + T_s + G$ units when the penalty of activation timer and Setup mode duration are also included. Based on the aforementioned behavior of AoII+ and with the PASTA property of Poisson arrivals, $\mathbb{E}[\int_0^{T_{cycle}} \Delta(\tau)\,\mathrm{d}\tau]$ is equal to the following expression,

$$\frac{\lambda(\mathbb{E}[T_e]\mathbb{E}[(T_a + T_s + G)^2] + \mathbb{E}[T_c]\mathbb{E}[G^2])}{2}, \tag{5.15}$$

where T_c is total time elapsed while idling in the Active mode throughout a cycle, and

$$\mathbb{E}[T_c] = \mathbb{E}[N_{T_e \leq T_i}]\mathbb{E}[T_e | T_e \leq T_i] + T_i. \tag{5.16}$$

Finally from equations (5.14), (5.15) and (5.16), we obtain the following expression for the average AoII+,

$$\mathbb{E}[\Delta] = \frac{T_a^2 + \mathbb{E}[T_s^2] + 2(T_a\mathbb{E}[G] + T_a\mathbb{E}[T_s] + \mathbb{E}[G]\mathbb{E}[T_s])}{2\mathbb{E}[T_{cycle}]} + \frac{(\mathbb{E}[N_{T_e \leq T_i}] + 1)\mathbb{E}[G^2]}{2\mathbb{E}[T_{cycle}]}. \tag{5.17}$$

5.5 Optimal Timer-based Schedulers for AoII+

In this section, we derive the optimal timer parameters for the schedulers AoII-ZIS and AoII-IS that minimize the energy consumption ratio under a constraint on the average AoII+. First, we denote the AoII+ constraint parameter by δ^* and we assume that

$$\delta^* \geq \delta^*_{min}, \tag{5.18}$$

where

$$\delta^*_{min} = \frac{\mathbb{E}[G^2]}{2(1/\lambda + \mathbb{E}[G])}, \tag{5.19}$$

is the resulting average AoII+ for a server that is always in the Active mode.

5.5.1 AoII-ZIS

We start by formulating the following optimization problem for AoII-ZIS,

$$\begin{aligned} \min_{T_a} \quad & \varphi \\ \text{s.t.} \quad & \mathbb{E}[\Delta] \leq \delta^* \\ & T_a \geq 0, T_i = 0. \end{aligned} \tag{5.20}$$

When $T_i = 0$, the energy consumption ratio is simplified as follows,

$$\varphi^{T_i=0} = 1 - \frac{\gamma_L(1/\lambda + T_a) + \gamma_S \mathbb{E}[T_s]}{T_a + \mathbb{E}[T_s] + 1/\lambda + \mathbb{E}[G]}. \tag{5.21}$$

Since $\varphi^{T_i=0}$ is a monotonously decreasing function of T_a, the maximum feasible value of T_a that satisfies the AoII+ constraint also minimizes the energy consumption ratio. Also, the average AoII+ expression can be simplified as,

$$\mathbb{E}[\Delta]^{T_i=0} = \frac{T_a + \mathbb{E}[G] + \mathbb{E}[T_s] - 1/\lambda}{2} + \frac{\sigma^2_{T_s} + \sigma^2_G - 1/\lambda^2}{2(T_a + \mathbb{E}[T_s] + 1/\lambda + \mathbb{E}[G])}, \tag{5.22}$$

where σ_{T_s} and σ_G stand for the standard deviation of T_s and G, respectively. If $\sigma^2_{T_s} + \sigma^2_G > 1/\lambda^2$, then $\mathbb{E}[\Delta]^{T_i=0}$ is a convex function of T_a, and if $\sigma^2_{T_s} + \sigma^2_G \leq 1/\lambda^2$, then $\mathbb{E}[\Delta]^{T_i=0}$ is a monotonously increasing function of T_a. In either case, the maximum value of T_a that satisfies the constraint $\mathbb{E}[\Delta]^{T_i=0} \leq \delta^*$ is obtained when $\mathbb{E}[\Delta]^{T_i=0} = \delta^*$ from the variable r_1 which is defined as,

$$r_1 = k_1 - \sqrt{k_1^2 + 2k_2}, \tag{5.23}$$

where

$$k_1 = \delta^* - \mathbb{E}[T_s] - \mathbb{E}[G], \tag{5.24}$$

and

$$k_2 = \delta^*(1/\lambda + \mathbb{E}[G] + \mathbb{E}[T_s]) - \frac{\mathbb{E}[T_s^2] + \mathbb{E}[G^2] + 2\mathbb{E}[T_s]\mathbb{E}[G]}{2}. \tag{5.25}$$

Note that the variable r_1 being negative or having a non-zero imaginary part implies that the age constraint δ^* is infeasible in the region $T_a > 0$. In this situation, the server must disable the sleep-wake scheduler to attain the minimum possible AoII+, δ^*_{min}. Therefore, the optimal activation timer parameter for AoII-ZIS is given as follows,

$$T_a^{T_i=0^*} = \max(r_1 \mathbb{1}_{\Im(r_1)=0}, 0), \tag{5.26}$$

where $\Im(\cdot)$ operator returns the imaginary part of its input argument, $\max(\cdot,\cdot)$ returns the larger of its two input arguments and

$$\mathbb{1}_{x=0} = \begin{cases} 1, & \text{if } x = 0, \\ 0, & \text{if } x \neq 0, \end{cases} \tag{5.27}$$

is the indicator function which tests whether the input argument is equal to zero or not.

5.5.2 AoII-IS

We formulate the following optimization problem for the AoII-IS,

$$\begin{aligned} \min_{T_a,T_i} \quad & \varphi \\ \text{s.t.} \quad & \mathbb{E}[\Delta] \leq \delta^* \\ & T_a, T_i \geq 0. \end{aligned} \tag{5.28}$$

For the rest of this section, we will proceed by assuming that $\mathbb{E}[\Delta] = \delta^*$, and later we will prove that for the optimization problem in (5.28), the optimal values of T_a and T_i are obtained when $\mathbb{E}[\Delta] = \delta^*$. The condition $T_i \geq 0$ is equivalent to $\mathbb{E}[N_{T_e \leq T_i}] \geq 0$. By rearranging the terms of (5.17), for a given value of δ^*, we can alternatively write $\mathbb{E}[N_{T_e \leq T_i}]$ as,

$$\begin{aligned} \mathbb{E}[N_{T_e \leq T_i}] = &-\frac{2\delta^*(1/\lambda + \mathbb{E}[G] + T_a + \mathbb{E}[T_s])}{2\delta^*(1/\lambda + \mathbb{E}[G]) - \mathbb{E}[G^2]} + \frac{2(T_a\mathbb{E}[G] + T_a\mathbb{E}[T_s] + \mathbb{E}[G]\mathbb{E}[T_s])}{2\delta^*(1/\lambda + \mathbb{E}[G]) - \mathbb{E}[G^2]} \\ &+ \frac{T_a^2 + \mathbb{E}[T_s^2] + \mathbb{E}[G^2]}{2\delta^*(1/\lambda + \mathbb{E}[G]) - \mathbb{E}[G^2]} \end{aligned} \tag{5.29}$$

which is a convex function of T_a with the assumption provided in (5.18). Therefore, the condition $T_i \geq 0$ (or equivalently $\mathbb{E}[N_{T_e \leq T_i}] \geq 0$) is only satisfied when T_a is greater than the root that satisfies $\mathbb{E}[N_{T_e \leq T_i}] = 0$. Note that this root is equal to the root of $\mathbb{E}[\Delta]^{T_i=0}$ given by equation (5.23), and an imaginary or negative result implies that $T_i > 0$ in the region $T_a \geq 0$. In light of these observations, the constraints $T_a, T_i \geq 0$ in (5.28) can equivalently be written as $T_a \geq T_a^{T_i=0^*}$ where $T_a^{T_i=0^*}$ is provided in (5.26). By substituting the identity provided in (5.29) in place of (5.12), we obtain the objective function of (5.28), φ, as

$$\varphi = 1 - \frac{j_1(2\delta^*(1/\lambda + \mathbb{E}[G]) - \mathbb{E}[G^2])}{j_2}, \tag{5.30}$$

where

$$j_1 = \gamma_L(1/\lambda + T_a) + \gamma_S\mathbb{E}[T_s], \tag{5.31}$$

and

$$j_2 = (1/\lambda + \mathbb{E}[G])(T_a^2 + \mathbb{E}[T_s^2] + 2(T_a\mathbb{E}[G] + T_a\mathbb{E}[T_s] + \mathbb{E}[G]\mathbb{E}[T_s])) - (T_a + \mathbb{E}[T_s])\mathbb{E}[G^2]. \tag{5.32}$$

The term $2\delta^*(1/\lambda+\mathbb{E}[G])-\mathbb{E}[G^2]$ in (5.30) does not alter the optimal T_a value (if there is any) which minimizes φ, and it is non-negative due to the assumption provided in (5.18). Moreover, (5.30) is a non-convex identity, and acquiring the optimal values of T_a and T_i is an arduous task. Instead, we define an equivalent optimization problem

$$\begin{aligned} \min_{T_a} \quad & y = \frac{j_2}{j_1} \\ \text{s.t.} \quad & \mathbb{E}[\Delta] = \delta^* \\ & T_a \geq T_a^{T_i=0^*}. \end{aligned} \tag{5.33}$$

Note that the objective function of the equivalent optimization problem is not a function of δ^*, and given by the following identity,

$$y = (1/\lambda + \mathbb{E}[G])(T_a + 2\mathbb{E}[G] + 2\mathbb{E}[T_s] - l_2) - \mathbb{E}[G^2] + \frac{l_1}{T_a + l_2}, \tag{5.34}$$

where

$$l_1 = (1/\lambda + \mathbb{E}[G])(\mathbb{E}[T_s^2] + 2\mathbb{E}[T_s]\mathbb{E}[G] + l_2^2 - (2\mathbb{E}[G] + 2\mathbb{E}[T_s])l_2) - (\mathbb{E}[T_s] - l_2)\mathbb{E}[G^2] \tag{5.35}$$

and

$$l_2 = 1/\lambda + \frac{\gamma_S \mathbb{E}[T_s]}{\gamma_L}. \tag{5.36}$$

If $l_1 \leq 0$, then y is a monotonously increasing function of T_a. Therefore, the optimal value for the objective function is achieved when $T_a^* = T_a^{T_i=0^*}$. On the other hand, when $l_1 > 0$, y is a convex function of T_a with a global minimum at the point

$$r_2 = -l_2 + \sqrt{\frac{l_1}{1/\lambda + \mathbb{E}[G]}}. \tag{5.37}$$

Thus, for $l_1 > 0$, the optimal value of T_a for (5.33) is written as,

$$T_a^* = \max(r_2 \mathbb{1}_{l_1>0}, T_a^{T_i=0^*}). \tag{5.38}$$

Note that the optimal value, T_a^*, is either equal to $T_a^{T_i=0^*}$ or r_2. We have previously proved in Section 5.5.1 that as δ^* increases, φ at the optimal value $T_a^{T_i=0^*}$ will decrease. Moreover, r_2 is a fixed global minimum of φ that does not vary with δ^*, and (5.30) shows that φ at any fixed value will decrease as δ^* increases. Therefore, as δ^* increases, φ at the optimal activation timer parameter, T_a^*, will decrease. This behavior also implies that for the optimization problem under the constraint $\mathbb{E}[\Delta] \leq \delta^*$ in (5.28), the minimum φ is achieved when $\mathbb{E}[\Delta] = \delta^*$. This completes the proof of the expression for T_a^* provided in (5.38) being the optimal activation timer parameter for AoII-IS.

After obtaining the optimal activation timer, T_a^*, of AoII-IS, one can obtain the optimal idling timer as,

$$T_i^* = -\frac{1}{\lambda} \log\left(\frac{1}{1 + \mathbb{E}[N_{T_e \leq T_i}]^*}\right), \tag{5.39}$$

where $\mathbb{E}[N_{T_e \leq T_i}]^*$ is obtained from (5.29) by substituting $T_a = T_a^*$.

Note that the identities provided in (5.26) and (5.38) for both strategies are valid for deterministic or PH-type distributed processing times with generally distributed setup times.

5.5.3 Adaptive Control of Timer Parameters

In this subsection, we propose low-complexity adaptive schedulers AoII-AIS and AoII-AZIS to update the timer parameters based on the time-varying arrival rate. The aim of AoII-AZIS and AoII-AIS is to adaptively tune the timer parameters to

Algorithm 1 Pseudocode for AoII-AIS and AoII-AZIS that is compiled at each update arrival

Update $\bar{\lambda}$ with Robbins-Monro algorithm with coefficient β
if H is deterministic **then**
  Update $\mathbb{E}[G]$ and $\mathbb{E}[G^2]$ using equations (5.4) and (5.5)
else if H is PH-type distributed **then**
  Update $\mathbb{E}[G]$ and $\mathbb{E}[G^2]$ using (5.9)
end if
if AoII-AIS is employed **then**
  Update T_a^* using (5.38)
  Update T_i^* using (5.39)
else if AoII-AZIS is employed **then**
  Update $T_a^{T_i=0^*}$ using (5.26)
end if

minimize the energy consumption ratio under constraints on the AoII+ based on the varying traffic conditions. For this purpose, we only need the timely approximation of arrival rate $\bar{\lambda}$, which we propose to update at each arrival using the last interarrival time with the Robbins–Monro algorithm with update coefficient β [72]. Subsequently, we update $\mathbb{E}[G]$ and $\mathbb{E}[G^2]$ through equations (5.4) and (5.5), respectively, when the service times are deterministic. For the case of PH-Type distributed service times, we alternatively employ the equations provided in (5.9) to obtain the first two moments of the busy period. Subsequently, we employ the identities (5.38) and (5.39) to obtain the optimal activation and idling timers, respectively, for AoII-AIS. Alternatively, for AoII-AZIS, the optimal activation timer can be found by (5.26). Note that when service times are deterministic the whole timer update process only includes basic mathematical operations (summation, subtraction, multiplication etc.). However, when the service times are PH-Type distributed, the process also involves a matrix inversion operation. Both of the proposed AoII-AIS and AoII-AZIS can be implemented with low complexity due to the underlying closed-form expressions to compute the optimal timer parameters. The pseudocode for AoII-AIS and AoII-AZIS is given in Algorithm 1.

Table 5.2: Processing and setup time distributions for each traffic setting

Setting ID	Setup times	Processing times
I	Exponential	Exponential
II	Deterministic	Exponential
III	Deterministic	Deterministic
IV	Deterministic	Erlang
V	Deterministic	Hyper-Exponential

5.6 Numerical Examples

In this section, we first validate our analytical results by obtaining the average AoII+ and energy consumption ratio under different traffic settings and comparing them with the simulation results. Throughout this section, we set the expected processing time $\mathbb{E}[H] = 1$, and we conduct our analysis with the traffic settings provided in Table 5.2. For Erlang and hyper-exponentially distributed processing times, we select PH-type distributions characterized with the pairs

$$\alpha = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -2 & 2 \\ 0 & -2 \end{bmatrix}, \tag{5.40}$$

and

$$\alpha = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}, \quad A = \begin{bmatrix} -2 & 0 \\ 0 & -2/3 \end{bmatrix}, \tag{5.41}$$

respectively.

For the purpose of validating the analytical results, we select $\mathbb{E}[T_s] = 2$, $T_a = 5$ and $T_i = 1$. We consider three different arrival rates with $\lambda = \{0.3, 1, 3\}$, for light, medium and heavy traffic, respectively. The analytical results that are obtained for $\mathbb{E}[\Delta]$ and φ with the identities provided in (5.17) and (5.12), respectively, are presented with the simulation results under five different traffic settings in Table 5.3.

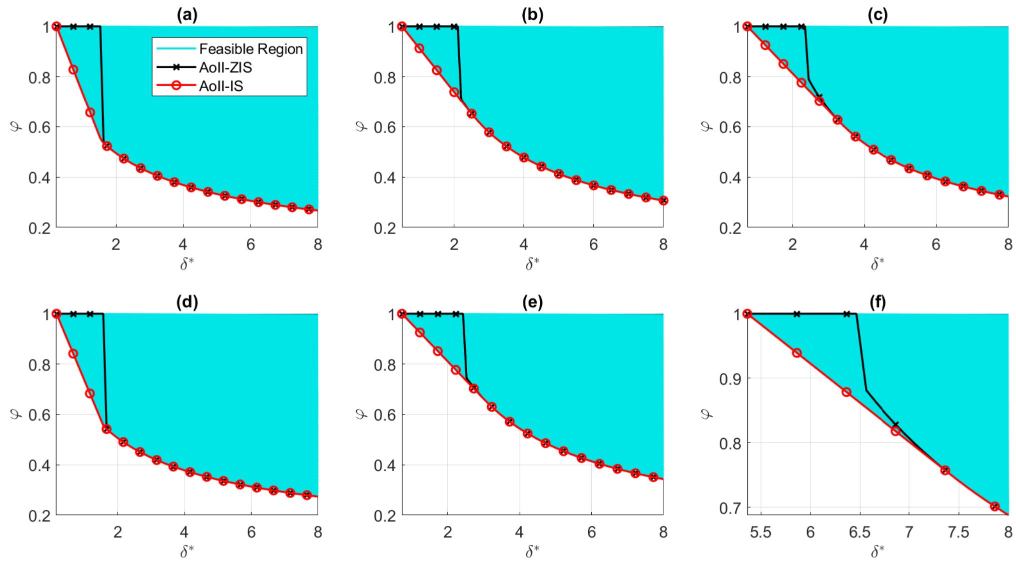

Figure 5.3: AoII+ constraint versus the achieved energy consumption ratio graphs for AoII-IS and AoII-ZIS under traffic setting II with arrival rates (a) $\lambda = 0.3$, (b) $\lambda = 1$ and (c) $\lambda = 3$, and under traffic setting III with arrival rates (d) $\lambda = 0.3$, (e) $\lambda = 1$ and (f) $\lambda = 3$. The feasible region for timer-based sleep-wake scheduling is also plotted for each subplot.

The simulation results for each traffic setting are achieved with accuracy better than 0.0050 verifying the analytical expressions. Under heavy traffic, as the variance of the service time distribution gets smaller, larger busy periods occur due to the preemptive policy. As a result, under heavy traffic, we observe larger average AoII+ values as variance of the service time decreases.

Next, we focus on the effectiveness of AoII-IS and AoII-ZIS. To the best of our knowledge, AoII-IS and AoII-ZIS are the only schedulers in the literature that are proposed to control sleep-wake scheduling timers under the average AoII+ constraint. Therefore, we select the scenario of the server being always in the Active mode as the benchmark scenario against which we compare our proposed schedulers in terms of the energy consumption ratio φ. We also compare the energy consumption ratios of AoII-IS and AoII-ZIS with respect to each other. For this purpose, we select $\mathbb{E}[T_s] = 3$ to better demonstrate the gain of AoII-IS over AoII-ZIS. We select the same arrival rate set to demonstrate the effectiveness under various load conditions. We plot the achieved energy consumption ratio, φ, versus the AoII+ constraint, δ^*, in Fig. 5.3 for the traffic settings II (in subplots a, b and c) and III (in subplots d, e and f). The graph of each scenario is plotted beginning from δ^*_{min} to $\delta^* = 8$. For each subplot, we also obtain the feasible region that is achievable with timer-based sleep-wake scheduling by conducting dense simulations with all possible timer configurations in the region $T_a, T_i \geq 0$. As can be observed in each subplot, the energy consumed with AoII-IS constitutes a lower bound on the feasible region validating the optimality of the scheduler. Furthermore, AoII-IS is shown to achieve lower φ values when compared to AoII-ZIS under stringent AoII+ requirements. As previously stated, for stringent AoII+ constraints that AoII-ZIS is unable to achieve, the only option is to disable sleep-wake scheduling which exhibits an abrupt jump in energy consumption ratio to one at each subplot. On the other hand, The AoII-IS is able to achieve any AoII+ value as long as $\delta^* > \delta^*_{min}$ without disabling the scheduling and thus, does not make an abrupt energy consumption ratio jump.

Finally, we test the effectiveness of the adaptive sleep-wake schedulers AoII-AIS and AoII-AZIS proposed in subsection 5.5.3. In this simulation setting, we only change the arrival rate in time and we update the arrival rate approximation at each arrival with the Robbins–Monro algorithm with update coefficient $\beta = 0.01$. Then, we use the updated arrival rate approximation to compute the timer parameters of AoII-AIS and AoII-AZIS. For this simulation, we select $\mathbb{E}[T_s] = 2$ and traffic setting II where the setup times are deterministic and processing times are distributed exponentially. In order to observe the performance of the proposed strategies in a dynamic environment, we define $\bar{\Delta}$ and $\bar{\varphi}$ as the estimated averages of AoII+ and energy consumption ratio, respectively, obtained from the average of last hundred sleep-wake cycles at an instant. We plot the varying arrival rate in time along with the resulting $\bar{\Delta}$ and $\bar{\varphi}$ when AoII-AIS and AoII-AZIS are employed in Fig. 5.4 for

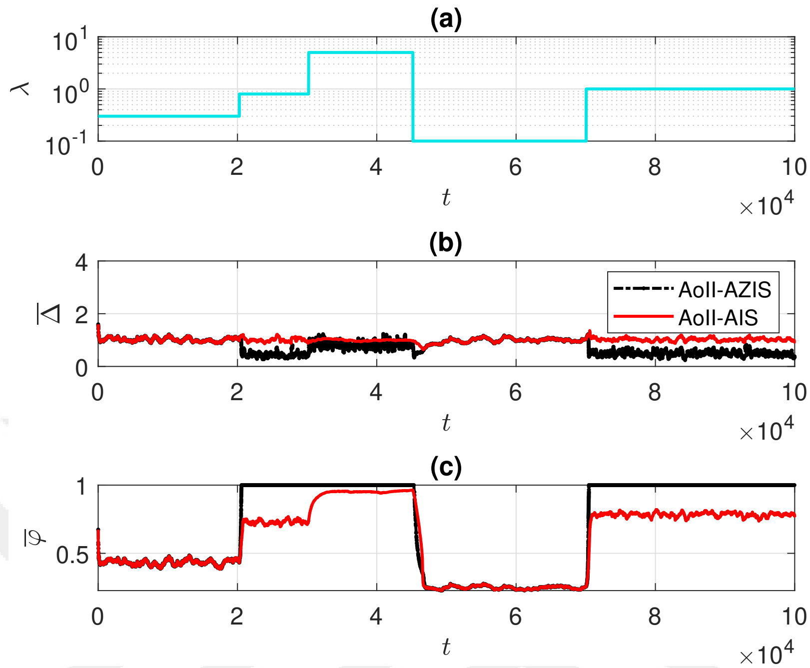

Figure 5.4: The estimated average AoII+ and energy consumption ratio values obtained in time when AoII-AIS and AoII-AZIS are employed. Dynamic arrival rate in time is plotted in (a), $\bar{\Delta}$ in time is plotted in (b), and $\bar{\varphi}$ in time is plotted in (c) for $\delta^* = 1$. .

$\delta^* = 1$. The $\bar{\Delta}$ versus time plots demonstrate the effectiveness of both schedulers for keeping AoII+ around the desired value under varying arrival rates. When the traffic gets heavy, AoII-AZIS disables the sleep-wake scheduling to stay below the desired age parameter. When this situation occurs, $\bar{\varphi}$ plots demonstrate the advantage of AoII-AIS over AoII-AZIS by still achieving a certain gain in the energy consumption ratio while satisfying the AoII+ constraint.

5.7 Chapter Summary

In this chapter, we investigate timer-based control of sleep-wake scheduling of the transmitter of IoT sensor nodes for which We analytically derive the average AoII+ and energy consumption ratio. We propose two optimal schedulers AoII-IS and AoII-ZIS which employ the closed-form expressions for the optimal timer parameters to minimize the power consumption under a constraint on average AoII+. These expressions enable the computation of the optimal timer parameters with low complexity when the model for the underlying update traffic is given. In simulations, we observed that, especially under stringent AoII+ constraints, AoII-IS is able to save significantly more energy compared to AoII-ZIS. Furthermore, we also propose open-loop adaptive schedulers AoII-AIS and AoII-AZIS to control AoII+ under varying update rates, and we validated their effectiveness via simulations.

Table 5.3: Average AoII+ ($\mathbb{E}[\Delta]$) and energy consumption ratio (φ) achieved by the timer-based sleep-wake scheduling for $\mathbb{E}[T_s] = 2$, $T_a = 5$ and $T_i = 1$ under different arrival rates, service and setup time distributions with Analytical (A) and Simulation (S) results.

		Average AoII+ ($\mathbb{E}[\Delta]$)					Energy consumption ratio (φ)				
λ		I	II	III	IV	V	I	II	III	IV	V
.3	A	2.7122	2.5565	2.5756	2.5672	2.5443	0.3852	0.3852	0.3957	0.3900	0.3814
	S	2.7110	2.5564	2.5759	2.5674	2.5445	0.3851	0.3851	0.3958	0.3901	0.3814
1	A	2.9122	2.7514	2.9074	2.8071	2.7178	0.5336	0.5336	0.5969	0.5578	0.5205
	S	2.9100	2.7517	2.9090	2.8092	2.7182	0.5338	0.5336	0.5967	0.5576	0.5202
3	A	1.5863	1.5271	5.5845	1.9089	1.4580	0.8461	0.8461	0.9632	0.8935	0.8268
	S	1.5858	1.5310	5.5867	1.9115	1.4609	0.8458	0.8458	0.9630	0.8932	0.8258

Chapter 6

Summary and Conclusions

Energy efficiency solutions in the design of NGN technologies is considered as one of the key enablers to meet the increasing demands [4]. Investigation of complex control mechanisms and novel QoS metrics for sleep-wake scheduling has the potential to offer more sustainable and energy efficient NGNs. With this motivation, we consider various applications of novel sleep-wake scheduling mechanisms with a focus on delay and age related metrics for communication systems within the scope of this thesis. In particular, we propose a novel open-loop dynamic coalescing technique for EEE that is based on MPC and queuing theory. For this purpose, we first propose the MPC-mean method which attempts to minimize the energy consumption of the Ethernet link, under constraints on mean of the queue waiting time. We also propose another MPC-based coalescer, termed as MPC-tail, that operates under constraints on the tail of the queue waiting time. Finally, we investigate information freshness for IoT sensor nodes by studying the energy-age of incorrect information trade-off with timer-based sleep-wake scheduling. In the following, we summarize the outcomes of the thesis in more detail.

In Chapter 3, We propose a novel open-loop adaptive frame coalescer based on the theory of MPC and queuing theory which hinges upon the idea of updating the residual timer parameter separate instants throughout a single coalescing cycle as opposed to determining it only once, a method used in conventional coalescers. The proposed MPC-mean coalescer keeps the average queuing delay below a target value. Using synthetic traffic and actual trace-driven traffic in exhaustive simulation studies, we have observed consistent energy consumption gains with MPC-mean with better control over average queuing delays, when compared to the open-loop dynamic timer-based coalescer from the existing literature.

In Chapter 4, we extend our study on MPC-based coalescing in EEE to control the tail of the queuing delays. Similar to MPC-mean, proposed MPC-tail updates the residual timer parameter at separate instants throughout a single coalescing cycle. The purpose of MPC-tail is to keep the tail of the queuing delay below a predefined threshold. The simulation results reveal that MPC-tail can effectively control the tail of the queuing delays especially and increasing the update count results in better energy consumption figures. Furthermore, the proposed CME fitting algorithm for Ethernet frames is shown to effectively approximate the queuing delay distributions.

In Chapter 5, we investigate timer-based control of sleep-wake scheduling of the transmitter of IoT sensor nodes for which we analytically derive the average AoII+ and energy consumption ratio. We propose two optimal schedulers AoII-IS and AoII-ZIS which employ the closed-form expressions for the optimal timer parameters to minimize the power consumption under a constraint on average AoII+. These expressions enable the computation of the optimal timer parameters with low complexity when the model for the underlying update traffic is given. Furthermore, we also propose open-loop adaptive schedulers AoII-AIS and AoII-AZIS to control AoII+ under varying update rates, and we validated their effectiveness via simulations.

In this thesis, we considered different QoS metrics for the control of sleep-wake scheduling in various wireless communication problems. Even if there is no common rule for the selection of QoS metric for a specific communication network, the underlying needs of the potential service to be provided, gives significant clues. For example, we employed average of the queuing delays for EEE interfaces, which is a common QoS metric used extensively in the literature. We extended this investigation further for the tail of the queuing delays and we propose the first open-loop adaptive EEE controller for the tail of the queuing delays, MPC-tail. Furthermore, we utilized AoII+ for IoT sensor networks for which the nodes often transmit the samples of the same process making it suitable for information freshness related metrics.

The studies presented in this thesis demonstrate the effectiveness of employing modern control mechanisms for sleep-wake scheduling. Through the use of MPC, we proposed an EEE control mechanism that updates the timer parameter of frame coalescing multiple times throughout a coalescing cycle. Consequently, overall system performance have been improved both in terms of the employed QoS metric and energy efficiency at the expense of increased computational load.

Next generation communication networks are often designed with various performance objectives such as information freshness, latency, energy efficiency and outage probability requirements under certain scenarios. As communication networks evolve the list of demands gets broader and requires more rigorous and stable control. Sleep-wake scheduling, as one of many mechanisms that save energy by trading off QoS, is essentially based on a simple approach of electrical on-off switching making it a feasible candidate for almost all of communication networks. 3GPP utilize sleep-wake scheduling via DRX mechanism. On-off duty cycling is also a critical topic for Low Earth Orbit (LEO) constellations such as Starlink for the availability of the service with energy harvesting satellites. IoT sensor networks often rely on the energy saving capabilities of the nodes in the network. As the primary source of energy consumption in terrestrial communication networks, the power usage of 5G base station (BS) is a significant concern. The Sleep mode of BS can be utilized to reduce mobile network energy consumption during periods of low traffic demand. Although the control mechanisms proposed in this thesis extend the scope of QoS metrics addressed via sleep-wake scheduling, need for more robust, smart and comprehensive mechanisms cease to exist.

We now identify several research directions following the study presented in this thesis. Advanced control techniques such as MPC continue their rapid improvement through increasing data and computing power, and easy-to-use toolboxes. Incorporation of these sophisticated mechanisms to sleep-wake scheduling clearly has the potential to improve the performance especially for complex problems in which network-level evaluation is necessary for control, such as LEO constellations or EEE networks. These techniques may also decrease the model and assumption dependency, which is a significant drawback of most existing sleep-wake strategies.

The control mechanisms developed in this thesis are based on conventional queuing models such as M/G/1, M/PH/1 and M/D/1. While these models are mostly accepted as effective means for analysis in the literature and employing them mostly result in low-complexity control mechanisms, the mechanisms proposed in this thesis may further be investigated with different queuing models which assume Markov modulated or Pareto arrivals etc. [92]. Furthermore, sleep-wake scheduling for IoT sensor networks should further be explored for multi-server and multi-source settings.

Bibliography

[1] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Dynamic EEE coalescing: Techniques and bounds," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 1, pp. 205–214, 2019.

[2] C. X. Wang, X. You, X. Gao *et al.*, "On the road to 6G: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys and Tutorials*, vol. 25, no. 2, pp. 905–974, May 2023.

[3] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, Mar. 2020.

[4] R. Coomonte, C. Feijóo, S. Ramos, and J.-L. Gómez-Barroso, "How much energy will your NGN consume? a model for energy consumption in next generation access networks: The case of Spain," *Telecommunications Policy*, vol. 37, no. 10, pp. 981–1003, 2013.

[5] 3GPP, "5G NR: Radio Resource Control (RRC) Protocol Specification," *3rd Generation Partnership Project (3GPP)), Technical Specification (TS) 38.331, version 17.1.0*, 2022.

[6] "IEEE Standard for Information Technology– Local and Metropolitan Area Networks– Specific Requirements– part 3: CSMA/CD Access Method and Physical layer Specifications Amendment 5: Media Access Control Parameters, Physical Layers, and Management Parameters for energy-efficient Ethernet," *IEEE Standard 802.3az-2010*, pp. 1–302, 2010.

[7] J. Xu, X. Wu, Q. Huang, and P. Sun, "How should the server sleep?—age-energy tradeoff in sleep-wake server systems," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 4, pp. 1620–1634, 2023.

[8] W. Liu, Y. Ma, and J. Li, "Equilibrium threshold strategies in observable queueing systems under single vacation policy," *Applied Mathematical Modelling*, vol. 36, no. 12, pp. 6186–6202, 2012.

[9] O. Gursoy and N. Akar, "Adaptive frame coalescing in energy efficient Ethernet with model predictive control and queuing theory," *IEEE Transactions on Green Communications and Networking*, vol. 8, no. 4, pp. 1574–1585, 2024.

[10] J. Xu and Y. Chen, "Information freshness in sleep-wake server systems," *IEEE Communications Letters*, vol. 25, no. 7, pp. 2186–2190, 2021.

[11] H.-C. Lin, K.-H. Lin, and H.-Y. Wei, "Age of information for power-saving devices with DRX mechanism," in *IEEE International Conference on Communications (ICC)*, 2023.

[12] P. Rafiee, Z. Ju, and M. Doroslovački, "Adaptive on/off scheduling to minimize age of information in an energy-harvesting receiver," *IEEE Sensors Journal*, vol. 24, no. 3, pp. 3898–3911, 2024.

[13] D. A. Hayes, D. Ros, Özgü Alay, P. Teymoori, and T. M. Vister, "Investigating predictive model-based control to achieve reliable consistent multipath mmWave communication," *Computer Communications*, vol. 194, pp. 29–43, 2022.

[14] N. Lazic, C. Boutilier, T. Lu, E. Wong, B. Roy, M. Ryu, and G. Imwalle, "Data center cooling using model-predictive control," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[15] O. Gursoy and N. Akar, "Model predictive control based adaptive frame coalescing in energy efficient Ethernet," in *2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR)*, 2024, pp. 227–232.

[16] I. Horváth, O. Sáfár, M. Telek, and B. Zámbó, "Concentrated matrix exponential distributions," in *Computer Performance Engineering*. Springer International Publishing, 2016, pp. 18–31.

[17] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: an engineering perspective," *The International Journal of Advanced Manufacturing Technology*, vol. 117, pp. 1–23, 11 2021.

[18] C. Cutler and B. Ramakar, "Dynamic matrix control — a computer control algorithm," 01 1980.

[19] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066102001867

[20] F. Oldewurtel, A. Parisio, C. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, "Energy efficient building climate control using stochastic model predictive control and weather predictions," in *Proceedings of the 2010 American Control Conference*. IEEE, Jun. 2010, pp. 5100–5105.

[21] N. Lazic, T. Lu, C. Boutilier, M. Ryu, E. Wong, B. Roy, and G. Imwalle, "Data center cooling using model-predictive control," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 3818–3827.

[22] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[23] S. Yoon, H. Jeon, and D. Kum, "Predictive cruise control using radial basis function network-based vehicle motion prediction and chance constrained model predictive control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3832–3843, 2019.

[24] C. Shen, Y. Shi, and B. Buckham, "Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5796–5805, 2018.

[25] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2017.

[26] A. Cenedese, F. Tramarin, and S. Vitturi, "An energy efficient Ethernet strategy based on traffic prediction and shaping," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 270–282, 2017.

[27] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro, "IEEE 802.3az: the road to energy efficient Ethernet," *Communications Magazine, IEEE*, vol. 48, no. 11, pp. 50–56, 2010.

[28] P. Reviriego, J. Maestro, J. Hernandez, and D. Larrabeiti, "Burst transmission for energy-efficient Ethernet," *Internet Computing, IEEE*, vol. 14, no. 4, pp. 50–57, 2010.

[29] S. Herreria-Alonso, M. Rodriguez-Perez, M. Fernandez-Veiga, and C. Lopez-Garcia, "A GI/G/1 model for 10 Gb/s energy efficient Ethernet links," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3386–3395, 2012.

[30] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models.* Baltimore, MD: Johns Hopkins University Press, 1981.

[31] ——, *Phase-type Probability Distributions.* Boston, MA: Springer US, 2013, pp. 1132–1134.

[32] S. Asmussen and M. Bladt, "Renewal theory and queueing algorithms for matrix-exponential distributions," in *Matrix-analytic Methods in Stochastic Models.* Marcel Dekker, 1996, pp. 313–341.

[33] Q.-M. He and H. Zhang, "On matrix exponential distributions," *Advances in Applied Probability*, vol. 39, no. 1, pp. pp. 271–292, 2007.

[34] A. Horvath, M. Scarpa, and M. Telek, *Phase Type and Matrix Exponential Distributions in Stochastic Modeling.* Springer International Publishing, 2016, pp. 3–25.

[35] N. Bean and B. F. Nielsen, "Quasi-birth-and-death processes with rational arrival process components," *Stochastic Models*, vol. 26, pp. 309–334, 2010.

[36] P. Buchholz and M. Telek, "Stochastic Petri nets with matrix exponentially distributed firing times," *Performance Evaluation*, vol. 67, pp. 1373–1385, 2010.

[37] A. David and S. Larry, "The least variable phase type distribution is Erlang," *Communications in Statistics. Stochastic Models*, vol. 3, no. 3, pp. 467–473, 1987.

[38] J.-C. Ke, "The optimal control of an M/G/1 queueing system with server vacations, startup and breakdowns," *Computers and Industrial Engineering*, vol. 44, no. 4, pp. 567–579, 2003.

[39] J. Artalejo, "On the M/G/1 queue with D-policy," *Applied Mathematical Modelling*, vol. 25, no. 12, pp. 1055–1069, 2001.

[40] O. Onireti, A. Mohamed, H. Pervaiz, and M. Imran, "Analytical approach to base station sleep mode power consumption and sleep depth," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2017.

[41] M. Yadin and P. Naor, "Queueing systems with a removable service station," *OR*, vol. 14, no. 4, pp. 393–405, 1963. [Online]. Available: http://www.jstor.org/stable/3006802

[42] D. P. Heyman, "The T-Policy for the M/G/1 Queue," *Management Science*, vol. 23, no. 7, pp. 775–778, 1977. [Online]. Available: https://doi.org/10.1287/mnsc.23.7.775

[43] D. Frankel, "brief review: Queueing analysis: A foundation of performance evaluation. volume 1: Vacation and priority systems, part 1 by H. Takagi (North-Holland, 1991)," *SIGMETRICS Perform. Eval. Rev.*, vol. 19, no. 2, p. 13, Sep. 1991. [Online]. Available: https://doi.org/10.1145/122564.1045501

[44] J. Teghem, "Optimal control of a removable server in an M/G/1 queue with finite capacity," *European Journal of Operational Research*, vol. 31, no. 3, pp. 358–367, 1987, methodology for Public Decision-Making Interactive Decision Support Systems Queue and Game Theory. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377221787900452

[45] K. Gakis, H. Rhee, and B. S. and, "Distributions and first moments of the busy and idle periods in controllable M/G/1 Queueing Models with Simple

and Dyadic Policies," *Stochastic Analysis and Applications*, vol. 13, no. 1, pp. 47–81, 1995. [Online]. Available: https://doi.org/10.1080/07362999508809382

[46] J.-C. Ke, "On M/G/1 system under NT policies with breakdowns, startup and closedown," *Applied Mathematical Modelling*, vol. 30, no. 1, pp. 49–66, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0307904X05000478

[47] Z. Niu, X. Guo, S. Zhou, and P. R. Kumar, "Characterizing energy–delay tradeoff in hyper-cellular networks with base station sleeping control," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 641–650, 2015.

[48] X. Guo, Z. Niu, S. Zhou, and P. R. Kumar, "Delay-constrained energy-optimal base station sleeping control," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1073–1085, 2016.

[49] J. Wu and Z. Lian, "A single-server retrial G-queue with priority and unreliable server under Bernoulli vacation schedule," *Computers and Industrial Engineering*, vol. 64, no. 1, pp. 84–93, 2013.

[50] S. Javed, P. Mach, Z. Becvar, and J. Gazda, "An energy-efficient sleeping strategy for multi-access edge computing," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, 2025, pp. 1–6.

[51] N. Akar, "Delay analysis of timer-based frame coalescing in energy efficient Ethernet," *IEEE Communications Letters*, vol. 17, no. 7, pp. 1459–1462, 2013.

[52] M. Mostowfi and K. Christensen, "An energy-delay model for a packet coalescer," in *2012 Proceedings of IEEE Southeastcon*, 2012, pp. 1–6.

[53] N. Maksić and M. Bjelica, "M/M/1 model of energy-efficient Ethernet with byte-based coalescing," *Annals of Telecommunications*, vol. 75, no. 7-8, pp. 291–305, 2020.

[54] J. Gong and J. Huang, "Age-energy analysis in multi-source systems with wake-up control and packet management," *IEEE Transactions on Mobile Computing*, vol. 24, no. 10, pp. 10 696–10 709, 2025.

[55] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Adaptive DRX scheme to improve energy efficiency in LTE networks with bounded delay," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2963–2973, 2015.

[56] J.-C. Ke, "Optimal NT policies for M/G/1 system with a startup and unreliable server," *Computers and Industrial Engineering*, vol. 50, no. 3, pp. 248–262, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835206000301

[57] S. Herreria-Alonso, M. Rodriguez-Perez, M. Fernandez-Veiga, and C. Lopez-Garcia, "A power saving model for burst transmission in energy-efficient Ethernet," *IEEE Communications Letters*, vol. 15, no. 5, pp. 584–586, 2011.

[58] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "How efficient is energy-efficient Ethernet?" in *2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2011, pp. 1–7.

[59] M. Rodríguez-Pérez, S. Herrería-Alonso, M. Fernández-Veiga, and C. López-García, "Delay properties of energy efficient Ethernet networks," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2194–2197, 2017.

[60] X. Pan, T. Ye, T. T. Lee, and W. Hu, "Power efficiency and delay tradeoff of 10GBASE-T energy efficient Ethernet protocol," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2773–2787, 2017.

[61] K. J. Kim, S. Jin, N. Tian, and B. D. Choi, "Mathematical analysis of burst transmission scheme for IEEE 802.3az energy efficient Ethernet," *Performance Evaluation*, vol. 70, no. 5, pp. 350–363, 2013.

[62] S. Herreria-Alonso, M. Rodriguez-Pérez, M. Fernández-Veiga, and C. López-Garcia, "Optimal configuration of energy-efficient Ethernet," *Computer Networks*, vol. 56, no. 10, pp. 2456–2467, 2012.

[63] M. Rodríguez-Pérez, S. Herrería-Alonso, R. F. Rodríguez-Rubio, and J. C. López-Ardao, "Leveraging energy saving capabilities of current EEE interfaces via pre-coalescing," *Journal of Network and Computer Applications*, vol. 166, p. 102734, 2020.

[64] M. Rodríguez-Pérez, M. Fernández-Veiga, S. Herrería-Alonso, M. Hmila, and C. López-García, "Optimum traffic allocation in bundled energy-efficient Ethernet links," *IEEE Systems Journal*, vol. 12, no. 1, pp. 593–603, 2018.

[65] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Bounded energy consumption with dynamic packet coalescing," in *2012 17th European Conference on Networks and Optical Communications*, 2012, pp. 1–5.

[66] A. Chatzipapas and V. Mancuso, "Measurement-based coalescing control for 802.3az," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, 2016, pp. 270–278.

[67] ——, "Modelling and real-trace-based evaluation of static and dynamic coalescing for energy efficient Ethernet," in *Proceedings AMC e-Energy*, Berkeley, CA, USA, 2013, pp. 161–172.

[68] W. Jiang, J. Wang, L. Peng, and J. Wang, "Generalized predictive control of the time window for energy efficient Ethernet with prediction," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1491–1495, 2021.

[69] J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, 2013.

[70] W. Jiang, K. Liao, Y. Yan, W. Cheng, L. Zhang, and J. Wang, "Modeling and analysis of latency distribution in the 40-100Gbps dual-mode energy efficient Ethernet," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1426–1437, 2021.

[71] S. W. Fuhrmann and R. B. Cooper, "Stochastic decompositions in the M/G/1 queue with generalized vacations," *Operations Research*, vol. 33, no. 5, pp. 1117–1129, 1985.

[72] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[73] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," in *2000 USENIX Annual Technical Conference (USENIX ATC 00)*. San Diego, CA: USENIX Association, 2000. [Online]. Available: http://mawi.wide.ad.jp/mawi

[74] N. Akar, O. Gursoy, G. Horvath, and M. Telek, "Transient and first passage time distributions of first- and second-order multi-regime Markov fluid queues via ME-fication," *Methodology and Computing in Applied Probability*, vol. 23, pp. 1257–1283, 2021.

[75] M. F. Neuts, "Explicit steady-state solutions to some elementary queueing models," *Operations Research*, vol. 30, no. 3, pp. 480–489, 1982.

[76] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, "Low-power status updates via sleep-wake scheduling," *IEEE/ACM Transactions on Networking*, vol. 29, no. 5, pp. 2129–2141, 2021.

[77] H. Huang, D. Qiao, and M. C. Gursoy, "Age-energy tradeoff optimization for packet delivery in fading channels," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 179–190, 2022.

[78] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7468–7482, 2019.

[79] S. Leng and A. Yener, "Age of information minimization for an energy harvesting cognitive radio," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 427–439, 2019.

[80] H. Zheng, K. Xiong, P. Fan, Z. Zhong, and K. Ben Letaief, "Age-based utility maximization for wireless powered networks: A Stackelberg game approach," in *IEEE Global Communications Conference (GLOBECOM)*, 2019.

[81] Y. Xiao, Q. Du, W. Cheng, and W. Zhang, "Adaptive sampling and transmission for minimizing age of information in metaverse," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 3, pp. 588–602, 2024.

[82] J. Cao, X. Zhu, S. Sun, E. Kurniawan, and A. Boonkajay, "Risk-aware and energy-efficient AoI optimization for multiconnectivity WNCS with short-packet transmissions," *IEEE Internet of Things Journal*, vol. 11, no. 12, pp. 21 474–21 485, 2024.

[83] H. Zheng, K. Xiong, M. Sun, H. Wu, Z. Zhong, and X. Shen, "Maximizing age-energy efficiency in wireless powered industrial IoE networks: A dual-layer DQN-based approach," *IEEE Transactions on Wireless Communications*, vol. 23, no. 2, pp. 1276–1292, 2024.

[84] Y. Chen and A. Ephremides, "Minimizing age of incorrect information for unreliable channel with power constraint," in *IEEE Global Communications Conference (GLOBECOM)*, 2021.

[85] Q. Liu, C. Li, Y. T. Hou, W. Lou, J. H. Reed, and S. Kompella, "Ao^{2}I: Minimizing age of outdated information to improve freshness in data collection," in *IEEE INFOCOM*, 2022.

[86] ——, "Wireless scheduling to optimize age of information based on earliest update time," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6352–6366, 2023.

[87] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2215–2228, 2020.

[88] J. Zhong, R. D. Yates, and E. Soljanin, "Two freshness metrics for local cache refresh," in *IEEE ISIT*, June 2018.

[89] H. Tang, J. Wang, Z. Tang, and J. Song, "Scheduling to minimize age of synchronization in wireless broadcast networks with random updates," in *IEEE International Symposium on Information Theory (ISIT)*, July 2019.

[90] C. Deng, J. Yang, and C. Pan, "Timely synchronization with sporadic status changes," in *IEEE International Conference on Communications (ICC)*, June 2020.

[91] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[92] F. Fiorini, M. Cococcioni, and M. Pagano, "Extending the applicability of the Pollaczek- Khinchin formula to the case of infinite service moments," *IEEE Transactions on Communications*, vol. 73, no. 4, pp. 2313–2328, 2025.