

Proceedings of the third Korea-Netherlands conference on queueing theory and its applications to telecommunication systems, July 12, 2007, Eindhoven

Citation for published version (APA): Kim, B., & Resing, J. A. C. (Eds.) (2007). *Proceedings of the third Korea-Netherlands conference on queueing theory and its applications to telecommunication systems, July 12, 2007, Eindhoven.* Eurandom.

Document status and date: Published: 01/01/2007

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.

• The final author version and the galley proof are versions of the publication after peer review.

 The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
 You may not further distribute the material or use it for any profit-making activity or commercial gain
 You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Proceedings of the Third Korea-Netherlands Conference on Queueing Theory and its Applications to Telecommunication Systems

July 12, 2007 EURANDOM, Eindhoven

Editors

Bara Kim, Korea University Jacques Resing, Eindhoven University of Technology

Organized by

- EURANDOM
- Telecommunication Mathematics Research Center (TMRC), Korea University

Preface

These proceedings contain the papers which are being presented at the *Third Korea-Netherlands Conference on Queueing Theory and its Applications to Telecommunication Systems,* to be held July 12, 2007 in the research institute EURANDOM in Eindhoven, The Netherlands.

The initiative for this series of Korea-Netherlands conferences was taken by Professor Bong Dae Choi, who also organized the first edition in Seoul, Korea, in June 2005. The second one took place in Amsterdam, The Netherlands, in October 2006. Exploiting the fact that a strong Korean delegation will be visiting the 14th INFORMS Applied Probability Conference in Eindhoven University of Technology, July 9-11, 2007, we have decided to organize the third conference edition right after this INFORMS conference.

The Korea-Netherlands conference series aim to bring together leading specialists in queueing theory and its applications to the performance analysis of telecommunication systems from Korea and The Netherlands, thus stimulating scientific interaction and exchange of knowledge between the two very active and prominent research communities of both countries. As in the previous two editions, the talks range from fundamental queueing theory to relevant applications to modern telecommunication networks.

In the first paper, **Hwang**, **Kim**, **Son** and **Choi** consider IEEE 802.16e, which is designed to support high capacity, high data rate and multimedia services as an emerging broadband wireless access system. The authors propose a new sleep-mode mechanism called the *power saving mechanism with periodic traffic indications*, and study its performance.

Núñez-Queija and **Prabhu** study the broadcast time of files in a Peer-to-Peer network with a large number N of initial nodes. In a network of altruistic nodes, they show the mean broadcast time to be $O(\log(N))$; in a network with free-riding nodes, a similar order of mean broadcast time may be achieved if nodes remain connected to the network for the duration of at least one more contact after downloading the file.

Park and **Chong** present an overview of wireless mesh networking technologies. They discuss a few promising standards for these technologies, and mention some challenging research issues associated with those standards.

De Haan, Boucherie and **van Ommeren** study a polling model with a so-called autonomous server. More precisely, they assume that the server spends an exponentially distributed period of time at a queue (independent of the number of customers present at each queue) before moving to the next queue. Applications of the model arise for instance in the context of wireless ad hoc networks. Their analysis is based on considering embedded Markov chains at specific instants.

The next two papers are devoted to retrial queues. Kim and Kim consider an M/G/1 retrial queue. By relating its waiting time to the waiting time in an ordinary M/G/1 queue with

random order of service, they are able to show that the waiting time distribution for the retrial queue is regularly varying if the service time distribution is regularly varying. **Nobel** considers a retrial queue with an unlimited number of servers, of which only a finite number is not dormant but active. Using Markov decision theory, he tackles the problem of determining when to activate or shut down servers in order to minimize the long-run average costs per unit time.

Hwang and **Ishizaki** study a cross-layer design problem for a wireless network with adaptive modulation and coding. In their cross-layer design, they consider both the physical layer and the medium access control layer. To capture the joint effect of the performance of both layers (packet transmission error rate at the PHY layer and packet overflow probability at the MAC layer) they introduce and study the effective bandwidth function of the packet service process at the MAC.

In the final paper, **Verloop**, **Borst** and **Núñez-Queija** investigate the delay-optimization problem for flows in bandwidth-sharing networks. For a class of simple linear bandwidth-sharing networks, they compare the performance of the optimal bandwidth-sharing policy with that of various α -fair strategies. They conclude that (optimization within) the family of α -fair strategies is likely to be adequate for most practical purposes.

Acknowledgment

We are grateful to the speakers and authors for submitting very interesting and state-of-the-art contributions. We gratefully acknowledge the fact that Professor Onno Boxma has actively participated in organizing the workshop, and that Professor Bong Dae Choi generously provided support and ideas. We also like to thank Dr. Ad Ridder who kindly provided us with the source files of the proceedings of the previous Korea-Netherlands conference (this saved us a lot of time in making these proceedings). Finally, we are indebted to EURANDOM for hosting and sponsoring the meeting, and to its staff – and in particular Mrs. Lucienne Coolen – for their tremendous help in organizing this meeting while simultaneously organizing the above-mentioned INFORMS conference.

Bara Kim, Korea University Jacques Resing, Eindhoven University of Technology

Contents

The Power Saving Mechanism with Periodic Traffic Indications: A New Sleep Mode Scheme in the IEEE 802.16e Eunju Hwang, Kyung Jae Kim, Jung Je Son and Bong Dae Choi	1
Scaling Laws for File Dissemination in P2P Networks with Random Contacts Rudesindo Núñez-Queija and Balakrishna Prabhu	16
Overviews on Wireless Mesh Networking Technologies JinSoo Park and HakJin Chong	26
A Polling Model with an Autonomous Server Roland de Haan, Richard Boucherie and Jan-Kees van Ommeren	35
Regularly Varying Tail of the Waiting Time Distribution in an $M/G/1$ Retrial Queue Jerim Kim and Bara Kim	56
A Retrial Queueing System with a Variable Number of Active Servers Rein Nobel	72
Cross-Layer Design with Adaptive Modulation and Coding Scheme for QoS Support and its Theoretical Analysis Gang Uk Hwang and Fumio Ishizaki	85
Performance Optimization in Bandwidth-Sharing Networks	101

The Power Saving Mechanism with Periodic Traffic Indications: A New Sleep Mode Scheme in the IEEE 802.16e

Eunju Hwang^a, Kyung Jae Kim^a, Jung Je Son^b and Bong Dae Choi^a ^a Department of Mathematics and Telecommunication Mathematics Research Center, Korea University, Seoul, KOREA ^b Telecom R&D Center, Samsung Electronics, Suwon, KOREA

This research is supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

Abstract

We propose a new sleep-mode scheme called the *power saving mechanism with periodic traffic indications* in the IEEE 802.16e. In the proposed scheme, traffic indication (TRF-IND) messages are regularly sent at every constant time to initiate transmission. Bandwidth and energy can be saved by not sending sleep request (MOB-SLP-REQ) and sleep response (MOB-SLP-RSP) messages, which are required in the original power saving classes in the 802.16e standard. We derive the Laplace Stieltjes transforms (LSTs) of the lengths of awake interval and sleep interval as well as the LST of queueing delay of a message. As performance measures we obtain sleep-mode ratio, power-consumption ratio, and mean total-delay. We show that our analytic results match with simulation results very well. Using our performance analysis we find the optimal system parameters such as a TRF-IND interval and a close-down time, which minimize the power consumption of MS while satisfying the required quality of service (QoS) on mean total-delay. The numerical results show that the proposed scheme performs better than the original power saving class of type 1 in the standard.

Keywords : IEEE 802.16e; power saving scheme; performance analysis; M/G/1 queueing model.

1 Introduction

The IEEE 802.16e is designed to support high capacity, high data rate and multimedia services as an emerging broadband wireless access system for fixed and mobile subscriber stations. Originally the IEEE 802.16 [1] has been designed for fixed subscriber stations, and the latest version IEEE 802.16e [2] has enhanced the original standard with mobility so that Mobile Stations (MSs) can move during services. Due to the promising mobility capability in the IEEE 802.16e, a power saving scheme is one of the significant issues for the battery-powered MSs to extend their lifetime before recharging. The IEEE 802.16e standard [2] introduces three kinds of sleep-mode operations called *power saving classes of type I*, *II and III*. Power saving class of type I is recommended for Best Effort (BE) and non-real-time variable rate (NRT-VR) traffics, power saving class of type II for unsolicited grant service (UGS) and real-time variable rate (RT-VR)

traffics, and power saving class of type III for management operation and multicast connections, respectively. These three types differ by their parameter sets, procedures of activation or deactivation, and policies of MS availability for data transmission.

Under the sleep-mode operation of the power saving classes in the IEEE 802.16e standard, in order to go to sleep, a MS sends a sleep request (MOB-SLP-REQ) message to its serving Base Station (BS) and obtains its approval through a sleep response (MOB-SLP-RSP) message from the BS. The sleep-mode in the power saving class of type I and type II involves two operational windows, namely, a sleep window and a listening window. After the MS receives a MOB-SLP-RSP message, it enters the sleep-mode from the awake-mode, and sleeps during a certain sleep window. At the following listening window the MS wakes up to listen to a traffic indication (TRF-IND) message from the BS, which tells whether the BS has any buffered downlink service-data-units (called packets in this paper) destined to it. If there is such traffic arrived at BS for the MS, the MS goes to the awake-mode. Otherwise, the MS remains in the sleep-mode and gets sleep for another sleep window. During the listening window, the MS listens to the TRF-IND message broadcasted from the BS, in the same way as in the state of awake-mode.

The MOB-SLP-REQ/RSP messages contain power saving class-type, and the sizes of initialsleep window, final-sleep window, and listening window. Each traffic can choose different power saving class-types by sending MOB-SLP-REQ/RSP messages. In the power saving class of type I, after sleeping during the initial-sleep window, if there is no buffered traffic at the BS at the following listening window, the MS doubles the sleep window size, and sleeps until the next listening window. These sleeping-and-listening events repeat with updated sleep window sizes until the MS is notified of the buffered packets destined to itself via a TRF-IND message. If the sleep window size reaches up to a maximum value of the final-sleep window, then the sleep window size is not doubled, but fixed.

Unlike the power saving class of type I, the power saving class of type II uses constant sleeping window size instead of doubling sleep window. Moreover, as opposite of type I, during the listening window of power saving type II, the MS may send or receive any packets.

There have been some recent studies that evaluate power saving schemes in the IEEE 802.16e. Han and Choi [3] analyzed the sleep mode operation of IEEE 802.16e in view of downlink traffic through semi-Markov chain, assuming that the bandwidth is always allocated one packet per one frame. Xiao [7] [8] proposed an analytical model of the power saving class of type I and investigated the energy consumption of IEEE 802.16e, including both incoming and outgoing frames. Zhang and Fujise [10] performed the energy and delay analysis on uplink and downlink traffic separately in the IEEE 802.16e. Xu, et al.[9] and Jang, et al. [4] proposed adaptive energy saving mechanisms in the IEEE 802.16e system, where the initial-sleep window and final-sleep window are adjusted according to the average traffic overload dynamically. Kong, et al. [5] evaluated and compared the sleep-mode operations of the power saving classes of type I and type II using an embedded Markov chain.

One of the problems in the power saving classes in IEEE 802.16e standard [2] is that they need overhead on the resources to transmit MOB-SLP-REQ/RSP messages. For example, during the switching time from awake-mode to sleep-mode, a MS sends a MOB-SLP-REQ message and receives a MOB-SLP-RSP message, and also needs actions for requesting and granting resource to exchange such signaling messages. The switching time between awake/sleep-modes excluding resource request period takes at least 4 frames as follows:

The switching time from awake-mode to sleep-mode =

- time to send a MOB-SLP-REQ message from MS to BS, 1 frame
- + time for BS to decode the MOB-SLP-REQ message, 1 frame
- + time to check data buffered and to generate a MOB-SLP-RSP message, 1 frame

+ time to send the MOB-SLP-RSP message from BS to MS, 1 frame.

Moreover, the problem is that during the switching time, MSs also consume the energy by staying in the awake-mode. In other words, the overhead to transmit the MAC management messages and the additional overhead for the resource request are necessary for the mode transition. The data link time to handle such overheads will be entirely wasted time in the awake mode, and will affect the power consumption efficiency. These problems motivate us to introduce a new sleep-mode scheme for the IEEE 802.16e.

In this paper, we propose a new sleep-mode scheme called *the power saving mechanism with periodic traffic indications* in order to save the resources and the power consumption by not sending MOB-SLP-REQ/RSP messages. The main characteristic of the power saving mechanism with periodic traffic indications is to send periodically a TRF-IND message by the BS at every constant interval, called *the TRF-IND interval*. This power saving mechanism with periodic traffic indications resolves the problem of the MOB-SLP-REQ/RSP in the IEEE 802.16e mentioned above, by saving their bandwidth as well as the switching time between awake/sleepmodes, and furthermore the energy consumption. The proposed power saving mechanism with periodic traffic indications unifies the power saving classes of type I and type II, and so it can be applicable to all traffics such as BE, NRT-VR, UGS and RT-VR traffics, by adjusting the length of a TRF-IND interval depending on traffics. The algorithm of the new sleep-mode scheme is so simple and convenient for the actual implementation. We will describe in details the power saving mechanism with periodic traffic indications with periodic traffic indications in Section 2 of this paper.

We provide an analytic performance study for the downlink traffic under the proposed scheme, assuming that arrivals of messages follow a Poisson process, which is applied for bursty traffic like BE traffic.

We organize the rest of this paper as follows. In Section 2, we introduce the power saving mechanism with periodic traffic indications for the 802.16e. In Section 8, the probabilistic characteristics of the lengths of awake/sleep intervals in the proposed sleep-mode operation are investigated by their Laplace-Stieltjes transforms (LSTs). As a result, we obtain sleep-mode ratio and power-consumption ratio. The LST and the mean of queueing-delay of a message are derived in Section 8. In Section 4, we show that our analytic results match with simulation results very well. The power saving mechanism with periodic traffic indications and the power saving class of type I in the IEEE 802.16e standard are compared with power-consumption ratio under the constraint that the two schemes have the same mean total-delay bound. As we expect, the proposed scheme in this paper performs better than the power saving class of type I in the IEEE 802.16e standard. We conclude the paper in Section 5.

2 The power saving mechanism with periodic traffic indications

The power saving mechanism with periodic traffic indications has a TRF-IND interval of a constant length *C*. The TRF-IND interval consists of a listening interval, an awake interval and a sleep interval. Under this new sleep-mode operation, the BS sends out periodically a TRF-IND message at the beginning of the listening interval in TRF-IND interval (see Fig.1). During the listening interval, a MS synchronizes with the serving BS downlink, listens to a TRF-IND message, and decides whether to go to an awake-mode or remain in a sleep-mode.

If there are data traffics in the buffer for the tagged MS, the BS sends a positive TRF-IND message and the MS enters an awake-mode. The BS transmits data during the awake-mode, and the awake-mode terminates if no traffic arrives during a fixed time, called *a close-down time* (referred to as *time-out* in some papers), of a constant length *T*, since the buffer of the BS has become empty. The close-down time is given to provide with services before entering a



Figure 1: Sleep-mode operation of the power saving mechanism with periodic traffic indications.

sleep-mode, to the data which arrive shortly after the buffer has become empty. If any data traffic arrives during the close-down time *T*, the BS keeps on the awake-mode and transmits the data. Otherwise, the MS goes to a sleep-mode from the awake-mode without exchanging MOB-SLP-REQ/RSP messages. The sleep-mode continues up to the next listening interval, where the BS sends out a TRF-IND message. If there are no data for the downlink traffic at the beginning of the listening interval, the BS sends a negative TRF-IND message, the MS stays in a sleep-mode, and a sleep interval becomes a whole TRF-IND interval.

To illustrate the sleep-mode operation of the power saving mechanism with periodic traffic indications, we give an example in Fig.1, where TRF-IND interval C = 12, close-down time T = 4, and listening interval L = 2 frames. Since a message composed of 3 packets arrives during a previous sleep interval, the BS sends the first positive TRF-IND message in the listening interval. The awake interval lasts from 3rd frame to 9th frame, where the transmission of 3 packets takes 3 frames and the close-down time is the last 4 frames. Because no data arrive during the close-down time, the MS enters a sleep-mode from the 10th to the 12th. The 13th frame is the beginning of a listening interval at which the next TRF-IND message arrives. The second TRF-IND message is negative, and then the MS restarts a sleep-mode.

Now we explain why we can use our proposed power saving mechanism with periodic traffic indications for all traffics. Every traffic has its own arrival process of packets, and its own requirement on packet delay or packet loss. For example, BE traffic like Web browsing arrives with series of packets, (which we regard as a message in this paper), and it demands less stringent requirement on delay. A good candidate for BE traffic is the power saving mechanism with periodic traffic indications having a sufficiently large value of C. On the other hand, UGS traffic such as VoIP or streaming service produces packets of fixed bytes at every fixed milliseconds, and it demands stringent requirement on delay. Thus a good candidate for UGS traffic is the power saving mechanism with periodic traffic indications having a sufficient on delay.

In this paper we consider bursty traffics of messages consisting of a random number of packets, which are not sensitive to delay. Thus we assume that in the power saving mechanism with periodic traffic indications, the length C of a TRF-IND interval is chosen so large that an awake-mode terminates within C and the requirement on delay is still satisfied. A sleep interval is the remaining part of a TRF-IND interval after the awake-mode terminates. As a system parameter, the length of a TRF-IND interval determines delay and power consumption.

3 Mathematical analysis

3.1 Assumptions

We make the following assumptions:

(1) The downlink message arrival toward the tagged MS follows a Poisson process with arrival rate λ .

(2) A message of random length is divided into packets of fixed length, and only one single packet is transmitted during every frame, i.e., MAC layer always allocates a resource to the tagged MS at every frame and the resource can transmit one packet. Thus the transmission time (service time in queueing terminology) of one message is a random multiple of one frame and so we may assume that the service time of a message has a general distribution B(x).

Mathematical analysis will be proceeded with the techniques developed in the M/G/1 *queuing model with a vacation* by regarding messages as customers and a sleep-mode as a vacation. We assume that the BS has an infinite buffer for the tagged MS.

It is assumed that the BS does not send downlink messages during the listening interval. The mathematical analysis will be done by classifying two intervals: awake interval and sleep interval. Since the listening interval is very short as 2 frames, for mathematical simplicity we assume that the listening interval is included in the sleep interval. However, in reality the listening interval is on the awake-mode, and so it will be subtracted from the mean length of sleep interval and added to the mean length of awake interval for obtaining power-consumption ratio in 3.2. Let

• A = the length of an *awake interval*,

• V = C - A, the length of a *sleep interval*, assuming that the length *C* of a TRF-IND interval is chosen so large that $A \le C$, as mentioned earlier.

Remark. If a traffic has the lowest priority among traffic classes in the IEEE 802.16e system, like BE traffic, according to the traffic conditions of BS, the bandwidth for BE traffic may not be allocated to the pending packets destined to the tagged MS at every frame. The time (called *HoL-delay*) to be allocated the bandwidth for the packet of BE traffic will be a random amount of time (see [2]). Assuming that the packet arrivals follow a Poisson process with arrival rate λ , the results of this paper can be also applied to the sleep mode operation initiated by BS by regarding the service time as HoL-delay + transmission time, and a sleep-mode as a vacation. In M/G/1 queueing, packets correspond to customers, and HoL-delay + transmission time correspond to a general service time.

3.2 Awake interval and sleep interval

In this subsection, first we describe the awake interval in terms of the busy periods generated by messages buffered at the BS. The LST of the length of the awake interval is computed in terms of the LST of the length of the sleep interval, and we obtain a relation between mean lengths of the awake/sleep intervals. We derive the LST and the mean of the length of the sleep interval. Sleep mode ratio and power consumption ratio are given.



Figure 2: Awake interval and sleep interval.

Laplace transforms of the lengths of awake interval and sleep interval

Let *Y* be the number of messages which arrive during a previous sleep interval *V*. We consider two cases of $Y(\omega) = 0$ and $Y(\omega) > 0$, separately. Note that $P(Y = 0) = \int_0^\infty P(Y = 0|V = x) dV(x) = \int_0^\infty e^{-\lambda x} dV(x) = V^*(\lambda)$.

(1) Case $1: Y(\omega) > 0$, and so the BS sends a positive TRF-IND message. The probability of this case is given by $P(case 1) = 1 - V^*(\lambda)$.

Let Θ_c be the length of the busy period generated by messages which have arrived during the previous sleep interval. As the busy period is over, the MS waits for a close-down time Tbefore it enters the sleep-mode. If there is no arrival message during the constant close-down time T, the MS goes from the awake-mode to the sleep-mode, and the length of the awake interval in case 1 is equal to $\Theta_c + T$ with probability $e^{-\lambda T}$. Otherwise, the awake interval is added to the busy period, which is generated by the messages arrived during the present close-down time. The MS stays on the awake-mode and waits again for another close-down time T. In the case that there is no message in the second close-down time T, the MS enters the sleep-mode, and the length of the awake interval in case 1 becomes $\Theta_c + T + (\Theta_T^{(1)} + T)$ with probability $e^{-\lambda T}(1 - e^{-\lambda T})$, where $\Theta_T^{(1)}$ represents the length of the (first) busy period generated by messages arrived during the first close-down time T. In such a way, the length of the awake interval given Y > 0 is expressed as follows (*w.p.* below stands for *with probability*). Fig.2 indicates the awake interval and the sleep interval in case 1.

The length *A* of the awake interval, conditionally on $\{Y > 0\}$ in case 1:

$$A = \begin{cases} \Theta_{c} + T & w.p. \ e^{-\lambda T} \\ \Theta_{c} + T + (\Theta_{T}^{(1)} + T) & w.p. \ e^{-\lambda T} (1 - e^{-\lambda T}) \\ \cdots & \cdots & \cdots \\ \Theta_{c} + T + (\Theta_{T}^{(1)} + T) + \cdots + (\Theta_{T}^{(n)} + T) & w.p. \ e^{-\lambda T} (1 - e^{-\lambda T})^{n} \\ \cdots & \cdots & \cdots \end{cases}$$
(1)

The LST $A^*(s|Y > 0)$ of *A* given Y > 0 is computed as follows.

$$\begin{aligned} A^*(s|Y>0) &= \sum_{n=0}^{\infty} E[e^{-sA}|Y>0, \ A = \Theta_c + \Theta_T^{(1)} + \dots + \Theta_T^{(n)} + (n+1)T]e^{-\lambda T}(1 - e^{-\lambda T})^n \\ &= \frac{e^{-(\lambda+s)T}\Theta_c^*(s)}{1 - (1 - e^{-\lambda T})e^{-sT}\Theta_T^*(s)} \end{aligned}$$



Figure 3: Approximation of $V^*(\lambda)$ by $e^{-\lambda E[V]}$.

where

$$\Theta_c^*(s) = \frac{V^*[\lambda - \lambda\theta^*(s)] - V^*(\lambda)}{1 - V^*(\lambda)}, \qquad \Theta_T^*(s) = \frac{e^{-\lambda T}}{1 - e^{-\lambda T}} [e^{\lambda T\theta^*(s)} - 1]$$

 $\Theta_c^*(s)$ and $\Theta_T^*(s)$ are the LSTs of the busy periods Θ_c and Θ_T generated by messages arrived during the previous sleep interval, and during the close-down time *T*, respectively, and $\theta^*(s)$ is the LST of the busy period θ of one message in the M/G/1 queueing system, with $\theta^*(s) = B^*(s + \lambda - \lambda \theta^*(s))$. Hence,

$$A^*(s|Y>0) = \frac{e^{-(\lambda+s)T}[V^*[\lambda-\lambda\theta^*(s)] - V^*(\lambda)]}{[1-V^*(\lambda)][1-e^{-(\lambda+s)T}(e^{\lambda T\theta^*(s)}-1)]}.$$

(2) Case 2: $Y(\omega) = 0$, and so the BS sends a negative TRF-IND message. Since we assume for mathematical simplicity that the listening interval is contained in the sleep interval, A is equal to zero in case 2. The probability of this case is $P(case 2) = V^*(\lambda)$.

Therefore, the LST of the length *A* of an awake interval is as follows:

$$A^{*}(s) = A^{*}(s|Y=0)P(Y=0) + A^{*}(s|Y>0)P(Y>0) = V^{*}(\lambda) + A^{*}(s|Y>0)[1-V^{*}(\lambda)],$$
$$A^{*}(s) = \frac{V^{*}(\lambda) + e^{-(\lambda+s)T}[V^{*}(\lambda-\lambda\theta^{*}(s)) - V^{*}(\lambda)e^{\lambda T\theta^{*}(s)}]}{1 + e^{-(\lambda+s)T}[1-e^{\lambda T\theta^{*}(s)}]}.$$
(2)

This equation gives a relation between $A^*(s)$ and $V^*(s)$, the LSTs of A and V. By differentiating it at zero, the relation of the mean lengths of an awake interval and a sleep interval can be obtained. Since $V^*(\lambda)$ is the probability that there is no message in the sleep interval V, we approximate $V^*(\lambda)$ by the probability that there is no message in the interval whose length is E[V], i.e., $V^*(\lambda) \cong e^{-\lambda E[V]}$. Fig.3 shows that this approximation is verified by simulation. The equalities $\rho = \lambda E[B]$ and $E[\theta] = \frac{E[B]}{1-\rho}$ are used to yield the equation of E[A] and E[V]:

$$E[A] = \frac{1}{1 - \rho} [\rho E[V] + T e^{\lambda T} - T e^{\lambda T} e^{-\lambda E[V]}].$$
(3)

This equation is a relation between E[A] and E[V] obtained from the relation of the LSTs of A and V in a general case with no assumption on the system parameters.

However, we assumed that the TRF-IND interval *C* is chosen so large that $A \leq C$ holds. Then V = C - A and the LST of *V*, $V^*(s) = A^*(-s)e^{-sC}$ and from (2) we have the following equation of $V^*(s)$:

$$V^{*}(s) = \frac{e^{-sC}V^{*}(\lambda)[1 - e^{-(\lambda - s)T}e^{\lambda T\theta^{*}(-s)}]}{1 + e^{-(\lambda - s)T}[1 - e^{\lambda T\theta^{*}(-s)}]} + \frac{e^{-sC}e^{-(\lambda - s)T}}{1 + e^{-(\lambda - s)T}[1 - e^{\lambda T\theta^{*}(-s)}]}V^{*}(\lambda - \lambda\theta^{*}(-s)).$$

This expression is a functional equation for the LST $V^*(s)$ of V. We solve for $V^*(s)$ for any given value s through the following iterative equation:

$$V_{n+1}^{*}(s) = R(s) + Q(s)V_{n}^{*}(D(s))$$

where $V_0^*(s)$ is some initial function and

$$R(s) = \frac{e^{-sC}V^*(\lambda)[1 - e^{-(\lambda - s)T}e^{\lambda T\theta^*(-s)}]}{1 + e^{-(\lambda - s)T}[1 - e^{\lambda T\theta^*(-s)}]}, \qquad Q(s) = \frac{e^{-sC}e^{-(\lambda - s)T}}{1 + e^{-(\lambda - s)T}[1 - e^{\lambda T\theta^*(-s)}]}$$

and $D(s) = \lambda - \lambda \theta^*(-s)$, then we obtain

$$V^{*}(s) = V^{*}(\lambda) \sum_{k=0}^{\infty} \left[e^{(\lambda - D^{k}(s))T} - e^{\lambda T \theta^{*}(-D^{k}(s))} \right] \prod_{j=0}^{k} Q(D^{j}(s)) + \lim_{n \to \infty} V_{0}^{*}(D^{n}(s)) \prod_{j=0}^{n-1} Q(D^{j}(s))$$
(4)

where $D^{j}(s) = D(D(\cdots D(s)))$, *j*-fold composition of D(s). The equation (4) is an expression in terms of known quantities and initial function V_0^* , which may be chosen as the LST of V_0 where V_0 is some proper random variable. Assuming (4) is convergent, E[V] can be obtained as in (5) below by differentiating (4), but in such a differentiating calculation, the term of V_0 will vanish to zero with a term of ρ . E[A] = C - E[V] < C and using (3),

$$E[V] = Te^{\lambda T}e^{-\lambda E[V]} - Te^{\lambda T} + (1-\rho)C.$$
(5)

E[V] is obtained by solving (5) numerically.

Sleep mode ratio and power consumption ratio

In this subsection, we are interested in the performance measures such as sleep-mode ratio and power-consumption ratio under the sleep mode operation:

• *Sleep-mode ratio* is defined by the ratio of mean length of a sleep interval to the total mean length of a sleep interval and an awake interval.

Sleep Mode Ratio =
$$\frac{E[V]}{E[A] + E[V]}$$
.

• *Power-consumption ratio* is defined by the ratio of power-consumption per time unit as MS is applied to the sleep mode operation to that as MS stays always awake.

Power Consumption Ratio =
$$\frac{\alpha_A(E[A] + L) + \alpha_V(E[V] - L) + \alpha_{on}}{\alpha_A(E[A] + E[V])}$$

where α_A and α_V are energies consumed per time unit at the awake interval and the sleep interval, respectively, and α_{on} is extra energy needed for the switchings from awake-mode to sleep-mode and from sleep-mode to awake-mode, and *L* is the length of a listening interval. In our calculation, the listening interval is included in the sleep interval for mathematical simplicity, but in reality the listening interval is on the awake-mode. This is why we add *L* to the mean length of the awake interval and subtract it from the mean length of the sleep interval.

3.3 Laplace transform of queueing delay

As the BS transmits messages in the system with a sleep-mode operation, the sleep-mode operation will give vast affects on the delay of the messages. The total delay of a tagged message is the sum of queueing delay (= the length of the remaining sleep interval and the transmission times for the messages queued in front of the message) and the transmission time of the tagged message itself if arrival of the tagged message occurs during the sleep interval. Since queueing delay and transmission time are independent, as we find either total delay or queueing delay, we obtain the other. Here we will find the queueing delay.

Let *W* be the queueing delay of a tagged message. We obtain the LST $W^*(s)$ of *W* by using the technique in Takagi [6] for the delay in the M/G/1 queueing model with a vacation (sleep interval).

$$W^*(s) = E[e^{-sW}|sleep]P(sleep) + E[e^{-sW}|awake]P(awake)$$
$$P(sleep) = \frac{E[V]}{E[A] + E[V]}, \quad P(awake) = \frac{E[A]}{E[A] + E[V]}.$$

Calculations for $E[e^{-sW}|sleep]$ and $E[e^{-sW}|awake]$ are given in Section 6 of the appendix.

$$W^*(s) = \frac{E[V]}{E[A] + E[V]} \frac{\int_{\lambda - \lambda B^*(s)}^{\infty} V^*(s) ds - \int_s^{\infty} V^*(s) ds}{s - \lambda + \lambda B^*(s)} + \frac{E[A]}{E[A] + E[V]} \frac{e^{-\lambda T} [\alpha \Psi(s) + \beta \Phi(s)]}{s - \lambda + \lambda B^*(s)}$$

where

$$\begin{split} \Psi(s) &= \frac{1 - V^*(\lambda - \lambda B^*(s))}{1 - V^*(\lambda)} + e^{-\lambda T[1 - B^*(s)]} - e^{-sT}, \\ \Phi(s) &= \frac{1 - e^{-\lambda T(1 - B^*(s))}}{1 - e^{-\lambda T}} + e^{-\lambda T[1 - B^*(s)]} - e^{-sT}, \\ \alpha &= \sum_{n=0}^{\infty} \frac{(1 - e^{-\lambda T})^n}{E[\Theta_c] + T + n(E[\Theta_T] + T)}, \quad \beta = \sum_{n=0}^{\infty} \frac{n(1 - e^{-\lambda T})^n}{E[\Theta_c] + T + n(E[\Theta_T] + T)}, \\ E[\Theta_c] &= \frac{\rho E[V]}{(1 - \rho)(1 - V^*(\lambda))}, \quad E[\Theta_T] = \frac{\rho T}{(1 - \rho)(1 - e^{-\lambda T})}. \end{split}$$

It is differentiated to derive the mean queuing delay E[W] in the queue as follows.

$$\begin{split} E[W] &= \frac{\lambda E[B^2]}{2(1-\rho)} \frac{E[A]}{E[A] + E[V]} \left[1 + \frac{\alpha e^{-\lambda T} E[V]}{1 - V^*(\lambda)} + \frac{\beta e^{-\lambda T} T}{1 - e^{-\lambda T}} - (\alpha + \beta) e^{-\lambda T} T \right] \\ &+ \frac{e^{-\lambda T}}{2(1-\rho)} \frac{E[A]}{E[A] + E[V]} \left[\frac{\rho^2 \alpha E[V^2]}{1 - V^*(\lambda)} + \frac{\rho^2 \beta T^2}{1 - e^{-\lambda T}} + (1-\rho^2)(\alpha + \beta) T^2 \right] + \frac{(1+\rho)(E[V])^2}{2(E[A] + E[V])} \right] \end{split}$$

The mean total delay is E[W] + E[B]. The mathematical calculation in this subsection is in a general case with no assumption on the system parameters. However, the delay which we will see as a numerical result in the next section is obtained under our assumption that the TRF-IND interval *C* is chosen so that $A \le C$ holds.

4 Numerical results and performance analysis

For numerical analysis we assume that a message consists of only one packet of fixed size, thus the service time is 5 ms, one frame long. The length of a listening interval *L* is 10 ms as 2 frames.



Figure 4: Comparison between numerical results and simulations.

In order to compare mathematical analysis and simulation results, we choose the following values of the parameters: Close-down time T = 30 ms, TRF-IND interval C = 200 ms. As mean interarrival time $\frac{1}{\lambda}$ varies (on the *x*-axes), delay, power-consumption ratio and sleep-mode ratio are compared between numerical analysis and simulation results in Fig. 4 (a)(b)(c), respectively. In obtaining the power-consumption ratio, $\alpha_A = 10$, $\alpha_V = 1$, and $\alpha_{on} = 40$ are chosen for example. Currently, we do not have any power consumption information of IEEE 802.16e as [3] mentioned.

Fig. 4 shows the analysis and the simulation match very well. The reason that the results on analysis differ from those on simulation for small values of mean interarrival time is that for our assumption $A \leq C$, large values of mean interarrival times are assumed so that all messages which arrived during the previous sleep interval are served within one TRF-IND interval, as we mentioned. This is why two results match very well for large values of mean interarrival times and have a little gap for small ones.

As the TRF-IND interval is longer, the sleep interval is longer and the delay is larger since the messages which arrived during the sleep interval have to wait until the next listening interval. On the other hand, as the TRF-IND interval is shorter, the delay is shorter, but the energy consumption increases because of the frequent switchings between awake/sleep-modes. Fig. 5 (a)(b)(c) depict comparisons of delay, power-consumption ratio and sleep-mode ratio, respectively, in 6 cases of TRF-IND intervals ($C = 100 \, ms$, $140 \, ms$, \cdots , $300 \, ms$) as mean interarrival time varies and close-down time is 30 ms. We see that the delay and the sleep-mode ratio increase as the TRF-IND interval and the mean interarrival time increase, whereas the power-consumption ratio decreases.

If a mean total-delay bound is given, the optimal TRF-IND interval minimizing the powerconsumption ratio can be chosen using the graphs in Fig. 5(a)(b)(c). For example, if the mean total-delay is less than 70ms and the mean interarrival time is 50ms, TRF-IND intervals are among 100, 140, 180, and 220ms as seen in Fig. 5(a). From Fig. 5(b) we find the optimal TRF-IND interval (= 220ms) which minimizes the power-consumption ratio while satisfying the



Figure 5: Performance measures vs. mean interarrival time.

required QoS on delay. In this case the power-consumption ratio and the sleep-mode ratio are 0.50 and 0.65, respectively, as shown in Fig. 5(b)(c).

The power saving mechanism with periodic traffic indications (abbreviated the periodic-TRF-IND-method below) in the present paper is proposed to improve the power saving classes in IEEE 802.16e standard [2]. Now we compare our scheme and the power saving class of type 1 with the optimal power-consumption ratio. The optimal power-consumption ratio (on y-axis) in the graphs of Fig. 6 is defined as follows: First, for a given equal mean total-delay bound in the type 1 of IEEE 802.16e standard and the periodic-TRF-IND-method, we select all system parameters satisfying the given mean total-delay bound. Secondly, we find minimum power-consumption ratio among all power-consumption ratios which have the system parameters selected in the first step. In details, for the graphs of the power saving class of type 1 in IEEE 802.16e standard (called type 1 on the graphs), as given a mean total-delay bound, we select all sets of initial-sleep windows, final-sleep windows and close-down times satisfying the mean total-delay bound, and then find the minimum (optimal) power-consumption ratio among all power-consumption ratios with the selected sets. For the graphs of the periodic-TRF-IND-method, as given a mean total-delay bound, we choose all pairs of TRF-IND intervals and close-down times satisfying the given mean total-delay bound, and among these pairs we take the optimal parameters minimizing power-consumption ratio. The optimal powerconsumption ratio is the minimum power-consumption ratio with the optimal parameters.

In Fig. 6(a)(b), we see the optimal power-consumption ratios which satisfy mean totaldelay bounds as the mean interarrival time is 30ms and as the mean interarrival time is 300ms. As seen in Fig. 6(a)(b), under the same delay bound, the optimal power-consumption ratio of the type 1 in the standard is larger than that of the periodic-TRF-IND-method. For example, in Fig. 6(a) with mean interarrival time 30ms, if the delay bound is 50ms, the optimal powerconsumption ratio of the type 1 is 0.78 and that of the periodic-TRF-IND-method is 0.59. In Fig.



Figure 6: Comparison between the power saving class of type 1 in IEEE 802.16e standard and the power saving mechanism with periodic traffic indications.

6(b) with mean interarrival time 300ms, under the delay bound 120ms, the optimal powerconsumption ratios of the type 1 and of the periodic-TRF-IND-method are 0.34 and 0.27, respectively. Now we consider delays of two schemes with the same power-consumption ratio. In the case of the mean interarrival time 30ms (see Fig. 6(a)), under the power-consumption ratio 0.6, the type 1 has delay 80ms and the periodic-TRF-IND-method has delay 50ms. If the mean interarrival time is 300ms (see Fig. 6(b)), under the power-consumption ratio 0.29, the type 1 has delay 160ms and the periodic-TRF-IND-method has delay 100ms.

5 Conclusion

In this paper, we propose the power saving mechanism with periodic traffic indications to support efficient power saving and to save resources for IEEE 802.16e. The proposed scheme has a constant TRF-IND interval, and TRF-IND messages are regularly sent by BS at the beginning of the listening interval in every TRF-IND interval. We analyze the probabilistic characteristics of the lengths of the awake interval and the sleep interval in the proposed scheme with a sufficiently large length of a TRF-IND interval for burst traffics. Sleep-mode ratio, power-consumption ratio, and mean total-delay of a message are computed to obtain the optimal system parameters while satisfying the QoS delay. We compare the performance between the original power saving class of type 1 in IEEE 802.16e standard and our power saving mechanism with periodic traffic indications. In conclusion, the proposed scheme in this paper improves the standard scheme in view of saving the resources as well as the switching time between the awake/sleep-modes, and furthermore the energy consumption. In the present paper, we show mathematical analysis and performance evaluation under the assumption of Poisson arrival for BE traffic in the IEEE 802.16e. However, the power saving mechanism with periodic traffic indications can be also applied for UGS and RT-VR traffics.

6 Appendix

In this appendix we calculate $E[e^{-sW}|sleep]$ and $E[e^{-sW}|awake]$.

Let
$$W_V^*(s) = E[e^{-sW}|sleep]$$
. If $V = x$,
 $E[e^{-sW}|V = x] = \int_0^x E[e^{-sW}|tagged message arrives at y] \frac{1}{x} dy = \frac{e^{-sx} - e^{-\lambda x[1-B^*(s)]}}{x[\lambda - \lambda B^*(s) - s]}$

Unconditioning on V = x, we use the integration of Laplace transform: since

$$\frac{1}{x}e^{-\lambda x[1-B^*(s)]} = \int_{\lambda-\lambda B^*(s)}^{\infty} e^{-sx}ds, \qquad \int_0^{\infty} \frac{1}{x}e^{-\lambda x[1-B^*(s)]}dV(x) = \int_{\lambda-\lambda B^*(s)}^{\infty} V^*(s)ds$$

thus

$$W_V^*(s) = \frac{\int_{\lambda-\lambda B^*(s)}^{\infty} V^*(s) ds - \int_s^{\infty} V^*(s) ds}{s - \lambda + \lambda B^*(s)}.$$

Let $W_A^*(s) = E[e^{-sW}|awake]$. By (1),

$$W_A^*(s) = \sum_{n=0}^{\infty} E[e^{-sW}|\Theta_c + T + (\Theta_T^{(1)} + T) + \dots + (\Theta_T^{(n)} + T)]e^{-\lambda T}(1 - e^{-\lambda T})^n.$$
(6)

 $E[e^{-sW}|\Theta_c + T + (\Theta_T^{(1)} + T) + \dots + (\Theta_T^{(n)} + T)]$

 $= E[e^{-sW}|tagged message arrives in (\Theta_c + T)]q_0 + \sum_{l=1}^{n} E[e^{-sW}|tagged message arrives in (\Theta_T^{(l)} + T)]q_l$

where

$$q_0 = \frac{E[\Theta_c] + T}{R_n}, \quad q_l = \frac{E[\Theta_T] + T}{R_n} \quad \text{and} \quad R_n = E[\Theta_c] + T + n(E[\Theta_T] + T).$$

We compute $W_c^*(s) = E[e^{-sW}|tagged message in (\Theta_c + T)]$ and $W_T^*(s) = E[e^{-sW}|tagged message in (\Theta_T^{(l)} + T)].$

$$W_{c}^{*}(s) = E[e^{-sW}|\Theta_{c}] \frac{E[\Theta_{c}]}{E[\Theta_{c}] + T} + E[e^{-sW}|T] \frac{T}{E[\Theta_{c}] + T}.$$
(7)

In order to find $E[e^{-sW}|\Theta_c]$ and $E[e^{-sW}|T]$: Let τ_0 be the service time for messages arriving in V, $\tau_0 = B_1 + \cdots + B_Y$. We call τ_0 the 0th generation of the busy period and the period for serving all messages that arrive during the m-1 st generation the *m*th generation, where $m = 1, 2, \cdots$. The length of the *m*th generation is denoted by τ_m . Then $\Theta_c = \sum_{m=0}^{\infty} \tau_m$,

$$E[e^{-sW}|\Theta_c] = \sum_{m=0}^{\infty} E[e^{-sW}|\Theta_c, \ \tau_m] \frac{E[\tau_m]}{E[\Theta_c]} = \sum_{m=0}^{\infty} \int_{t=0}^{\infty} E[e^{-sW}|\tau_m = t] \frac{t \ d\tau_m(t)}{E[\tau_m]} \frac{E[\tau_m]}{E[\Theta_c]}$$

where the probability distribution function of τ_m , during which the tagged message arrives, is given by $\frac{t d\tau_m(t)}{E[\tau_m]}$.

$$E[e^{-sW}|\tau_m=t] = \frac{e^{-\lambda t[1-B^*(s)]} - e^{-st}}{t[s-\lambda+\lambda B^*(s)]}.$$

Thus

$$E[e^{-sW}|\Theta_c] = \frac{\sum_{m=0}^{\infty} [\tau_m^*(\lambda - \lambda B^*(s)) - \tau_m^*(s)]}{E[\Theta_c][s - \lambda + \lambda B^*(s)]} = \frac{1 - V^*(\lambda - \lambda B^*(s))}{(1 - V^*(\lambda))E[\Theta_c][s - \lambda + \lambda B^*(s)]}$$

because

$$\tau_{m+1}^{*}(s) = \tau_{m}^{*}(\lambda - \lambda B^{*}(s)), \quad \tau_{0}^{*}(s) = \frac{V^{*}(\lambda - \lambda B^{*}(s)) - V^{*}(\lambda)}{1 - V^{*}(\lambda)}.$$

As for $E[e^{-sW}|T]$:

$$E[e^{-sW}|T] = \int_0^T [e^{-sW}|tagged \ packet \ at \ y] \frac{1}{T} dy = \frac{e^{-\lambda T[1-B^*(s)]} - e^{-sT}}{T[s-\lambda+\lambda B^*(s)]}$$

Hence, in (7),

$$W_{c}^{*}(s) = \frac{1}{(E[\Theta_{c}] + T)[s - \lambda + \lambda B^{*}(s)]} \left[\frac{1 - V^{*}(\lambda - \lambda B^{*}(s))}{1 - V^{*}(\lambda)} + e^{-\lambda T[1 - B^{*}(s)]} - e^{-sT} \right]$$

Similarly,

$$W_T^*(s) = \frac{1}{(E[\Theta_T] + T)[s - \lambda + \lambda B^*(s)]} \left[\frac{1 - e^{-\lambda T(1 - B^*(s))}}{1 - e^{-\lambda T}} + e^{-\lambda T[1 - B^*(s)]} - e^{-sT} \right]$$

Hence, from (6)

$$W_A^*(s) = \frac{e^{-\lambda T}}{s - \lambda + \lambda B^*(s)} [\alpha \Psi(s) + \beta \Phi(s)]$$

where $\Psi(s)$, $\Phi(s)$, α and β are given in subsection 3.3.

References

- [1] IEEE 802.16-2004, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Standard for Local and Metropolitan Area Networks, October 2004.
- [2] IEEE 802.16e-2006, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, February 2006.
- [3] K. Han and S. Choi, "Performance analysis of sleep mode operation in IEEE 802.16e mobile broadband wireless access systems", Proc. IEEE VTC'2006 Spring, Melbourne, Australia, May 7-10, 2006.
- [4] J. Jang, K. Han and S. Choi, "Adaptive power saving strategies for IEEE 802.16e mobile broadband wireless access," APCC 2006.
- [5] L. Kong and D. H. K. Tsang, "Performance study of power saving classes of type I and II in IEEE 802.16e," Proceedings 2006 31st IEEE Conference on Local Computer Networks.
- [6] H. Takagi, *Queueing analysis: a foundation of performance evaluation.* vol. 1, North-Holland, 1991.
- [7] Y. Xiao, "Energy saving mechansm in the IEEE 802.16e Wireless MAN", IEEE Communication Letters, vol 9, no. 7, pp. 595-597, July 2005.
- [8] Y. Xiao, "Performance analysis of an energy saving mechanism in the IEEE 802.16e Wireless MAN," *Proceedings of IEEE CCNC 2006*, January 2006.

- [9] F. Xu, W. Zhong and Z. Zhou, "A novel adaptive saving mode in IEEE 802.16e system," *MILCOM* 2006.
- [10] Y. Zhang and M. Fujise, "Energy management in the IEEE 802.16e MAC," IEEE Communication Letter vol. 10, no. 4, April 2006, pp. 311-313.

Scaling Laws for File Dissemination in P2P Networks with Random Contacts

Rudesindo Núñez-Queija^{a,b} and Balakrishna Prabhu^{a,c,d} ^aCWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands ^bTNO Information and Communication Technology, Delft, The Netherlands ^cDepartment of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands ^dEURANDOM, P.O. Box 513, 5600 MB Eindhoven, The Netherlands Email: {sindo,b.j.prabhu}@cwi.nl

This research is financially supported by the SCALP project of the Network of Excellence EURO-FGI and by the Dutch Bsik/BRICKS project.

Abstract

In this paper we obtain the scaling law for the mean broadcast time of a file in a P2P network with an initial population of N nodes. In the model, at Poisson rate λ a node initiates a contact with another node chosen uniformly at random. This contact is said to be successful if the contacted node possesses the file, in which case the initiator downloads the file and can later upload it to other nodes. In a network with altruistic nodes (i.e., nodes do not leave the network) we show that the mean broadcast time is $O(\log(N))$. In a network with free-riding nodes, our main result shows that a $O(\log(N))$ mean broadcast time can be achieved if nodes remain connected to the network for the duration of at least one more contact after downloading the file, otherwise a significantly worse O(N) time is required to broadcast the file.

1 Introduction

Traffic measurements in the Internet suggest that Peer-to-Peer (P2P) networks are becoming increasingly popular among Internet users for sharing and distributing files. The salient features of a P2P architecture are the vast possible improvements in scalability and robustness compared to the traditional client-server architecture. In the best-case scenario, a P2P network can broadcast a file in a time which scales only logarithmically with the number of nodes in the network, which compares favourably with the linear scaling for a client-server network. This vast improvement in the distribution time can be explained as follows. After downloading the file, a client node acts as a server and uploads the file to other client nodes. Thus, the service capacity of the network actually increases with the number of the nodes in the network. The presence of several simultaneous servers in the network significantly reduces the vulnerability of the file distribution process to attacks on the central server.

Although the P2P architecture is very promising in terms of scalability, there are several factors which are critical to achieving the promised performance gains. The foremost factor is the willingness of each client node to become a server node. A failure on the part of client nodes to do so (also called *free-riding*) would impact both scalability and robustness. As a simple example, if each client node departs immediately after having downloaded the entire file, the network will behave as a client-server network with the broadcast time scaling linearly in

the number of client nodes. Thus, the impact of free-riding (i.e., downloading but not uploading) on the broadcast time needs a detailed investigation. Another factor which is critical to achieving the performance gains is the connectivity of the underlying network graph. Again as a simple example, if the network would be configured in a star topology then again the time to broadcast would be linear in the number of nodes which is significantly worse than the logarithmic scaling possible on a hypercube topology [4].

A detailed study of the impact of these factors on the performance of a P2P network is thus essential in obtaining conditions under which P2P architecture can outperform the clientserver architecture. In this paper, we study a closed P2P network in which N client nodes and one seed node form a fully connected file sharing network. The purpose of this network is to broadcast the file which is available at the seed node. Each node (except the seed node) can leave the network after downloading the file. The model described above is suited to study the behaviour of P2P network when subjected to *flash-crowds*, i.e., a large population of nodes joins the network in a very short interval of time [2]. One of the main performance measures in such networks is the time required to broadcast the file. The focus of this paper is to study the impact of free-riding on the mean broadcast time. Our main result states that a $O(\log(N))$ mean broadcast time is achievable in P2P networks with free-riding provided that nodes stay long enough in the network after having downloaded the file, otherwise a significantly worse O(N) time is required, thereby implying a phase transition phenomenon for the scaling law of the mean broadcast time.

1.1 Related work

The availability of free P2P software such as BitTorrent [1] has contributed significantly to the increased popularity of P2P networks among Internet users, and has also motivated research in several aspects of the P2P networks. The BitTorrent P2P algorithm achieves a significant improvement in performance by dividing the file into several chunks. Instead of downloading a large file from one server, nodes can download smaller chunks from different servers. A file download is said to be complete when a node has downloaded all the corresponding chunks. Previously, low bandwidth nodes were reluctant to upload because of large file sizes, and thus reluctant to participate in P2P networks. However, breaking the file into smaller chunks provides such users an incentive to upload data and join a P2P network. In [4], the authors studied the problem of the optimal broadcast of a set of *C* messages to *N* nodes over a complete graph in a *deterministic* setting. They showed that the optimal broadcast time is $O(C + \log_2(N))$. In our present paper, we give an analogous result in a stochastic setting for the one chunk case and with the more realistic assumption of nodes being able to leave the network.

In general, the analysis of P2P networks in a stochastic setting (i.e., random node arrival and node departures) is too complex to permit an exact analysis. Hence approximate models have been constructed to obtain some insights into the performance of P2P networks. For example, using a fluid model Qiu and Srikant [6] have studied the behaviour of the number of servers and clients in a BitTorrent network in which there are external arrivals, and servers leave the network at a certain rate. The emphasis is on studying the number of servers and clients in the equilibrium state. In [3], the authors generalized the above model to be able to study the spread of chunks within networks. One of their results was to show that chunk selection policies (like *rarest first* or *random selection*) had negligible impact on the performance of a P2P network. In practice, arrivals to a network may not occur at a constant rate, and the so-called *flash crowd* phenomenon has often been observed [2]. For example, the latest version of a popular software is solicited by a large number of users (a flash crowd) close to the release date. Usually, the interest in this version may taper off as time progresses, and the critical period of

operation is during the first few days when the interest is large. We note that the interest may increase again when a new version of the software is released, for example. Unlike the above mentioned work, our objective in this paper is to characterize the mean *broadcast time* in a *closed* network. In that respect, our work is an extension of [4] to stochastic setting with *free-riding*. However, the analytical tools (Markov chains and fluid limits) are similar to those in [6] and [3].

As a first step, we present the analysis for the one chunk case, i.e., the file is not divided as in BitTorrent. From the insights obtained using this model, we intend to extend this analysis to the multiple chunk case and for different network topologies.

The rest of the paper is organised as follows. In section 2, we describe the model, give the assumptions, and formulate the problem in terms of the input parameters. The analysis for a network without free-riding is presented in section 3. In section 4 our main result on the mean broadcast time in a network with free-riding users is derived. Using simulations, similar results for general values of C are given in section 5. Finally, we conclude with possible research directions in section 6.

2 Problem Formulation

Consider a population of *N* nodes who want to download a file which is available at the seed node at time 0. We assume that the underlying network topology is fully connected, and that a node, which is present in the network and has the file, is willing to upload the file to other nodes. In order to download a file, a node initiates a contact with another node chosen uniformly at random among the existing nodes. These contacts are initiated at Poisson rate λ . If the contacted node has the file then the file transfer is assumed to take place in a time which is negligible compared to the mean time between contacts. This model of a contact process for file dissemination is based on the one analysed in [3] and [5].

In order to model the impatient behaviour of nodes in a real network, we shall assume that, after having downloaded the file, a node leaves the network at a Poisson rate μ . The case $\mu = 0$ corresponds to altruistic nodes who remain in the network for the duration of the broadcast whereas the case $\mu = \infty$ corresponds to nodes who leave the network immediately after downloading the file. Finally, we shall assume that the seed node remains in the network for the duration of the broadcast. This assumption guarantees that all the nodes will be able to download the file eventually. One could possibly study the number of unsuccessful nodes if the seed node also had the possibility of leaving the network. Such an analysis could give clues to the vulnerability of the network to malicious attacks on the seed node.

Given the above setting, our main interest in this paper is to study the impact of the departure rate, μ , on the mean time to broadcast the file to all *N* nodes. Intuitively, a higher departure rate of the nodes would translate into fewer servers present in the network which would then increase the mean broadcast time. We shall formalize this intuitive result by showing that, depending on the departure rate, different scaling laws are possible for the mean broadcast time.

3 Mean broadcast time with altruistic nodes ($\mu = 0$)

We first take a look at the case $\mu = 0$. Through this analysis we expect to obtain a lower bound on the mean broadcast time for $\mu > 0$. In a deterministic setting when the sequence in which file downloads take place is determined at time 0, file broadcast can be achieved in $O(\log(N))$ time units. We now show that this is also the case in the stochastic contact process model we described earlier. Thus, the mean broadcast time in a random contact based P2P network is of the same order as the optimal broadcast time.

For the case of $\mu = 0$, we shall study the network in discrete time where each time step corresponds to the time between two contacts. Since no nodes leave the network, contacts are initiated at rate $N\lambda$ (we assume that the seed does not initiate any contacts). The mean broadcast time can be obtained by multiplying the mean number of contacts by $(N\lambda)^{-1}$.

Let Y_n denote the number of servers in the network after the *n*th contact. The dynamics of the process $\{Y_n, n > 0\}$ can be described as follows.

$$Y_{n+1} = \begin{cases} Y_n & \text{w.p.} & p(Y_n) \\ Y_n + 1 & \text{w.p.} & 1 - p(Y_n) \end{cases}$$

where $p(i) = 1 - \frac{N-i}{N} \frac{i+1}{N}$. The probability p(i) describes the probability of an unsuccessful contact when there are *i* servers and one seed present in the network.

Let A_i denote the number of contacts made in state *i*. The random variable A_i is geometrically distributed with

Prob.
$$(A_i = k) = (1 - p(i))p(i)^{k-1}, \ k \ge 1.$$



Figure 1: The relation between S and dynamics of Y

Let $S_j = \sum_{i=0}^{j} A_i$. The random variable S_j is the number of contacts needed to distribute the file to j + 1 nodes. This relation between the processes S_j and Y_n is illustrated in figure 1 from which we can infer that $P(Y_n < j) = P(S_{j-1} > n)$. Since A_j s are independent random variables,

$$E[S_j] = \sum_{i=0}^{j} E[A_i], \quad j = 0, 1, ..., N - 1,$$

$$Var[S_j] = \sum_{i=0}^{j} Var[A_i], \quad j = 0, 1, ..., N - 1.$$

Also, since A_i s are geometrically distributed,

$$E[A_i] = \frac{1}{1-p(i)} = \frac{N^2}{(i+1)(N-i)},$$

$$Var[A_i] = \frac{p(i)}{(1-p(i))^2} = \left(1 - \frac{i+1}{N}\left(\frac{N-i}{N}\right)\right) \left(\frac{N^2}{(i+1)(N-i)}\right)^2.$$

Therefore, the mean number of contacts to broadcast the file (i.e., to distribute the file to N nodes) is

$$E[S_{N-1}] = \sum_{i=0}^{N-1} E[A_i]$$

= $\frac{N^2}{N+1} (2\log(N) + o(\log(N))).$

Let T_j denote the time needed to distribute the file to j nodes. Then,

$$T_j = \sum_{k=0}^{S_{j-1}} \tau_k,$$
 (1)

where the random variable τ_k denotes the time between the *k*th and the (k+1)th contact. Since $\tau_1, \tau_2, ...$ is a sequence of i.i.d. exponential random variables with mean $(N\lambda)^{-1}$, we can use Wald's lemma and obtain the mean broadcast time as

$$E[T_N] = E[S_{N-1}]E[\tau_1]$$
(2)

$$= 2\frac{N}{\lambda(N+1)}\log(N) + o(\log(N)).$$
(3)

4 Mean broadcast time with free riding nodes ($\mu > 0$)

In the previous section we obtained a mean broadcast time of $O(\log(N))$ for $\mu = 0$. For the other extreme case of $\mu = \infty$, we can see that the broadcast time would be O(N) because the seed would be the only server present in the network, and every user will have to download the file from the seed node, which will take O(N) encounters.

In this section we shall obtain the scaling law when $0 < \mu < \infty$, i.e., nodes leave the network at rate μ after downloading the file. Let Y(t) (resp. X(t)) denote the number of servers (resp. downloaders) present in the network at time t. The joint process $\{X(t), Y(t)\}_{t\geq 0}$ is a two-dimensional Markov process on $\{0, 1, 2, ..., N\} \times \{0, 1, 2, ..., N\}$ whose dynamics can be described as follows

$$Y(t) \rightarrow \begin{cases} Y(t) + 1 & \text{at rate} \quad \lambda X(t) \frac{Y(t) + 1}{X(t) + Y(t)} \\ Y(t) - 1 & \text{at rate} \quad \mu Y(t) \end{cases},$$
(4)

$$X(t) \rightarrow X(t) - 1$$
 at rate $\lambda X(t) \frac{Y(t)+1}{X(t)+Y(t)}$, (5)

with (X(0), Y(0)) = (N, 0). The increase in Y(t) only happens when downloaders make a successful contact (the +1 in the numerator is due to the presence of the seed). The rate of decrease of Y(t) is $\mu Y(t)$ independent of the number of downloaders.

We now study this process in the large initial population limit i.e., $N \to \infty$. Let $(x(t), y(t)) \equiv \left(\frac{X(t)}{N}, \frac{Y(t)}{N}\right)$ be the rescaled process. Then, y(t) (resp., x(t)) is the fraction of nodes at time t who do (resp., do not) have the file. For $0 < \mu < \infty$, we can write the following fluid equations for the dynamics x and y,

$$\frac{dy}{dt} = -\mu y + \lambda x \frac{y}{x+y},\tag{6}$$

$$\frac{dx}{dt} = -\lambda x \frac{y}{x+y}.$$
(7)

Balakrishna Prabhu

Therefore,

$$\frac{d(x+y)}{dt} = -\mu y. \tag{8}$$

Combining equations (7) and (8), we get

$$\frac{d}{dx}(x+y) = \frac{\mu}{\lambda} \frac{x+y}{x},\tag{9}$$

which can be solved to obtain

$$x + y = c_0 x^{\frac{\mu}{\lambda}}.\tag{10}$$

We can determine c_0 by noting that y = 0 when x = 1. Thus, we can characterize the evolution of the number of servers as a function of the number of downloaders in the network as follows

$$y = -x + x^{\rho}, \ x \in (0, 1),$$
 (11)

where $\rho = \mu/\lambda$. In figure 2, we plot solutions of (10) for different values of $\rho = \mu/\lambda$. As $\mu \to 0$, the solution approaches the line x + y = 1, which is the case when no nodes leave the network. The solution to the differential equation obtained above is valid only for $\rho < 1$. For $\rho > 1$ we



Figure 2: Solutions of $y = -x + x^{\rho}$ for various values of ρ .

obtain negative values for y, which makes the solution infeasible. We now have the following result.

Theorem 4.1. The mean broadcast time for a file in a P2P network with free-riding users scales as

- O(N) if $\rho > 1$;
- $O(\log(N))$ if $\rho < 1$.

Thus, there is phase transition in the scaling law at $\rho = 1$. This suggests that if nodes stay for the duration of one more contact after downloading the file then a significantly improved scaling law for the broadcast time prevails even in the presence of free-riding nodes.

Proof. We first prove that the mean broadcast time is O(N) for $\rho > 1$. For this case we upper bound the Y(t) process by another process which is easier to analyse. Let $\{Z(t)\}_{t\geq 0}$ be defined as

$$Z(t) \rightarrow \begin{cases} Z(t) + 1 & \text{at rate } \lambda Z(t), \\ Z(t) - 1 & \text{at rate } \mu Z(t). \end{cases}$$
(12)

For the same initial conditions, $\{Y(t)\}$ is stochastically smaller than $\{Z(t)\}$. For $\rho > 1$, we have $\operatorname{Prob}(Z(t) > N) \to 0$ as $N \to \infty$. Hence, for large N, we can conclude that Z(t) will not reach O(N) and, consequently, Y(t) will remain o(N). From (5), when X(t) is on linear scale it will decrease at a constant rate. Thus, for large N, the mean time required for X(t) to go from $\alpha_1 N$ to $\alpha_2 N$ will be linear in N. Hence, the mean broadcast time will be O(N).

For $\rho < 1$, we will follow similar arguments. In order to determine the mean broadcast time, we divide the analysis in three phases. The first phase corresponds to the time required for the number of servers to reach O(N). In the second phase, both the number of downloaders and the number of servers are O(N) and the fluid analysis is valid. In the final phase, the number of nodes goes to zero.

For the time spent in the first phase, we find the mean time required for Y(t) to exceed level $\epsilon_{\rho}N$. This level will depend on ρ as not all values of $y(t) \in (0, 1)$ are feasible for a given ρ . First, we find a lower bound for the rate of increase of Y(t). Let γ be the maximal solution of the equation $-x + x^{\rho} = \epsilon_{\rho}$ in (0, 1). Then

$$\begin{split} \lambda X(t) \frac{Y(t)}{X(t) + Y(t)} &> \lambda Y(t) \frac{X(t)}{X(t) + \epsilon_{\rho}} \\ &> \lambda Y(t) \frac{\gamma}{\gamma + \epsilon_{\rho}}. \end{split}$$

The second inequality follows from the fact that x/(x + 1) is an increasing function in x, and that if $Y(t) < \epsilon_{\rho}N$ then $X(t) > \gamma N$. We now bound Y(t) by $\hat{Z}(t)$ described by

$$\hat{Z}(t) = \begin{cases} \hat{Z}(t) + 1 & \text{at rate} \quad \lambda \frac{\gamma}{\gamma + \epsilon_{\rho}} \hat{Z}(t), \\ \hat{Z}(t) - 1 & \text{at rate} \quad \mu \hat{Z}(t). \end{cases}$$
(13)

We choose a $\gamma > \rho^{\frac{1}{1-\rho}}$ which then determines ϵ_{ρ} . For this choice of γ , $\lambda \frac{\gamma}{\gamma+\epsilon_{\rho}} = \lambda \frac{\gamma}{\gamma^{\rho}} > \lambda \rho = \mu$. For such a choice of parameters, $\hat{Z}(t)$ and, consequentially, Y(t) grow exponentially with time. Hence, the time for Y(t) to reach $\epsilon_{\rho}N$, say t_1 , is $O(\log(N))$.

For the time spent in the second phase, we first solve (7) to obtain

$$t(x) = \frac{1}{\lambda(1-r)} \log\left(\frac{x+x^{\rho}}{2x}\right).$$
(14)

From this equation, the time for x to start from a fraction γ and reach a fraction γ^* is a constant independent of N. Hence the time spent in the second phase is O(1).

For the time spent in the third phase, we shall bound the time required for x(t) starting from $x(\tau) = \gamma *$ to reach 0. For a given ρ , y > x if $x < \frac{1}{2^{(1-\rho)}}$. We first fix a $\gamma^* < \frac{1}{2^{(1-\rho)}}$. For $x < \gamma^*$,

$$\lambda x \frac{y}{x+y} > \frac{1}{2} \lambda x. \tag{15}$$

Since x is non-increasing, if $x(t_2) < \gamma^*$ then $x(t) < \gamma^*$ and $y(t) > x(t), \forall t > t_2$. Hence, the above inequality will remain valid once x is smaller than γ^* . Let $\{\hat{X}(t)\}$ be described by

 $\hat{X}(t) \rightarrow \hat{X}(t) - 1$ at rate $\frac{1}{2}\lambda \hat{X}(t)$.

From this definition, the process $\{X(t)\}$ is stochastically smaller than $\{\hat{X}(t)\}$. Since $\hat{X}(t)$ decreases exponentially, we can conclude that X(t) also decreases to 0 in logarithmic time.

From the above analysis, the time spent in the first phase is upper bounded by $O(\log(N))$, the time spent in the second phase is O(1), and the time spent in the final phase is upper bounded by $O(\log(N))$. Since the time to broadcast cannot be lower than $\log(N)$, we can conclude that the mean broadcast time is $O(\log(N))$ for $\rho < 1$.

5 Simulations

Using simulations, in this section we will observe that the above phase transition holds when C is greater than unity as well. We simulate the model with C = 10 and C = 50 and obtain the mean broadcast time as a function of N. In figure 3, the mean broadcast time is shown as a function of $\log_2(N)$ for C = 10 and two different values of ρ smaller than 1. Figure 4 shows the mean broadcast time as a function of N for two different values of ρ larger than 1.



Figure 3: Mean broadcast time versus $\log_2(N)$. C = 10.



Figure 4: Mean broadcast time versus N. C = 10.

In figures 5 and 6, we plot the mean broadcast time versus $\log_2(N)$ and N, respectively, for C = 50. From these plots we observe that the phase transition at $\rho = 1$ appears to be true for larger values of C as well.



Figure 5: Mean broadcast time versus $\log_2(N)$. C = 50.



Figure 6: Mean broadcast time versus N. C = 50.

6 Conclusions and future work

In this paper we quantified the effect of free-riding users on the mean broadcast time of a file in a P2P network. Our main result showed that a logarithmic broadcast time can be achieved if nodes stay in the network for the duration of one more contact, i.e., if they upload the file at least once. Otherwise a significantly worse linear scaling is achieved. Thus, if nodes stay in the network for the duration of one more contact, a random contact based P2P network can broadcast a file in a time which is of the same order as the optimal time.

Our future work will seek to extend these results to the multiple chunk case, and also to study the effect of the network topology on the mean broadcast time.

References

- [1] BitTorrent. [Online]. Available: http://www.bittorrent.com.
- [2] Xuan Chen and John Heidemann. Flash crowd mitigation via adaptive admission control based on application-level observations. *ACM Trans. Inter. Tech.*, 5(3):532–569, 2005.

- [3] L. Massoulié and M. Vojnović. Coupon Replication Systems. In Proc. ACM SIGMETRICS, Banff, Canada, 2005.
- [4] J. Mundinger, R. Weber, and G. Weiss. Analysis of peer-to-peer file dissemination. *To appear in Performance Evaluation Review, Special Issue on MAMA* 2006.
- [5] I. Norros, B.J. Prabhu, and H. Reittu. Flash crowd in a file sharing system based on random encounters. In *Proc. InterPerf*, Pisa, 2006.
- [6] D. Qiu and R. Srikant. Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In *Proc. ACM Sigcomm*, Portland, OR, 2004.

Overviews on Wireless Mesh Networking Technologies

JinSoo Park^a and HakJin Chong^b

^aResearch Director, Advanced Research Laboratory, KT, Korea, e-mail: vtjinsoo@paran.com
 ^bAssistant Vice President, Advanced Research Laboratory, KT, Korea, e-mail: hjchong@kt.co.kr

Abstract

Wireless mesh networking (WMN) technologies have captured attention from both industry and academia recently, and they can be the ultimate wireless networking solution for the next decade. WMNs are characterized as having self-configuring and self-healing nature along with flexible interoperability with other networks. In this paper, we discuss a few promising wireless mesh networking technologies, especially those from IEEE 802.15.5, IEEE 802.11s, 6LoWPAN standards, and address a few crucial research issues that are associated with the standards.

1 Introduction

WMN has emerged as one of the wireless networking technologies to realize ubiquitous society. Today, there are various types of wireless networking technologies such as WCDMA, WiMAX, Wibro, HSDPA, HSUPA, etc. These types of networks are in general developed as a backbone of networks, providing broadband networking services. Also these networks involve considerable amount of investments for their deployment and elaborate network planning. In order to construct the ubiquitous society where peer-to-peer communications between devices and human beings are common, networks that can operate in pico- and nano- cell environment are inevitable. People have made a lot of effort to develop proper networking technologies that meet this requirement. Those include Wireless Local Area Network(WLAN)s, Wireless Personal Network(WPAN)s, Body Area Network(BAN)s, and Wireless Sensor Network(WSN)s. Among these networking technologies, WLANs, WSNs and WPANs can be the most important ones to enable communications among devices and humans to operate. Mesh networking issues lies at the center of these technologies for the peer-to-peer communications. The mesh network can play as an information collector from neighboring sensor nodes as well as a bridge between sensor nodes and backbone networks.

There are three major standardization bodies to drive the international standard for wireless mesh networks, IEEE 802.11s, IEEE 802.15.5, and 6LoWPAN. IEEE 802.11s strives to lead WLAN-based mesh networking technologies, while IEEE 802.15.5 WPAN-based ones. IEEE 802.11s adopts two procedures to implement the mesh networks among nodes, which includes formation of topology and routing. The topology formation relates to collecting local information among nodes and finding its neighbor nodes. The routing algorithm finds the best route to a destination based on the collected local information from the topology formation. All these procedures are executed in layer two of network stack. IEEE 802.15.5 and 6LoWPAN study the mesh networking technologies based on WPAN MAC and PHY standards. IEEE 802.15.5 leads its standard focusing on MAC and PHY issues, while 6LoWPAN higher layer ones. Those issues from 6LowPAN include packet header compression and format, routing, security, etc., in order to transmit IPv6 packets over IEEE 802.15.4 networks. In this article, we discuss several mesh networking technologies, mostly from recent publications and standard documents.



Figure 1: WMN architecture

This document is organized as follows. In section 2, we define generic structure of wireless mesh networks and present details of topology and routing algorithms, along with a network employing multi-channel schemes. In section 3, we explain one example routing method discussed in 6LoWPAN as well as a promising MAC protocol, adopting a new beacon scheduling algorithm in superframe. Section 4 discusses several major research issues associated with wireless mesh networks. Finally, a conclusion of this article is given in section 5.

2 WMN based on IEEE 802.11s standard

WMN is conceived to achieve a network infrastructure by relaxing the major constraint of mobile ad hoc networks "infrastructureless", and it introduces a physical hierarchy in the network by adopting static wireless relay nodes and mobile client nodes. Figure 1 depicts generic architecture of a WMN where a group of mesh routers sits in the middle where various client nodes, Wi-Fi nodes, and sensor networks are connected.

Current 802.11 ad hoc mode is not sufficient to implement the multi-hop nature of wireless mesh networks, and recent advance on 802.11 standard such as 11e and 11n also has their inherent dependency on wireline infrastructure backbones and the last, single-hop communication structure. In this sense, IEEE 802.11s is designed to provide an IEEE 802.11 wireless DS that supports both broadcast/multicast and unicast delivery at the MAC layer using radioaware metrics over self-configuring multihop topologies. Objectives of the standard are summarized as: increased range/coverage & flexibility in use, possibility of increased throughput, reliable performance, seamless security, power efficient operation, multimedia transport between devices, backward compatibility and interoperability for interworking. IEEE 802.11s defines three types of nodes that constitute a mesh network, mesh point(MP), mesh access point(MAP), and mesh portal(MPP). MPs are nodes that relay frames each other in a router-

- MP (Mesh Point) : Relay frames each other in a router-like hop-by-hop fashion
- . MAP (Mesh Access Point) : Mesh relaying + AP service for clients
- * MPP (Mesh Portal) : Acting as a bridge to other networks



Figure 2: Definition of nodes for an IEEE 802.11s WMN

like hop-by-hop fashion. MAP performs two roles; that is, it works as a mesh relay node as well as AP for clients. MPP acts as a bridge to other nodes.

2.1 Topology formation and routing

802.11s mesh network is composed of two major procedures called topology formation and routing[1]. In a topology formation, nodes exchange their own information with their neighbor nodes to find their neighbors. Figure 3 summarizes the procedure to form a topology in a network.

First, when a mesh point running simple channel unification protocol(SCUP) is powered up, the system goes into a neighbor discovery phase where periodical advertisements of beacons are exchanged among nodes in the neighbor. SCUP is a protocol that assigns a common channel to a subset of MPs belonging to the same mesh network. Request and response frames are invoked on demand by a mesh point to find the neighbor. If neighbors are found to exist, connection is established between nodes by assigning all channels to their own network interface. This is also accompanied by authentication and association among neighbor nodes. If not found, the node selects one channel at random and assigns itself Channel Precedence Indicator (CPI). CPI is an indicator to differentiate mesh networks, and it is used to merge a group of network nodes into one mesh network group, having the same CPI. In general, CPI is computed as a sum of a random number and the time spent by an MP in the WLAN mesh[1].

After the phase of topology formation is over, the routing algorithm begins to operate. As mentioned before, the routing in the WLAN mesh networks is executed in the layer two of protocol stack. Based on the local information collected in the topology formation procedure, the routing algorithm determines a route to a destination. A few well-known ad hoc routing



Figure 3: Procedure for topology formation

algorithms are employed such as Ad hoc On demand Distance Vector (AODV) and Dynamic Source Routing (DSR)[2]. When running these algorithms, MAC address is used to identify each node to a destination. To secure a more stable route to the destination, multiple paths can be selected to the destination based on a metric. The metric is used to indicate status of channel between nodes. One example of the metric is given below

$$C_a = \left[O_{ca} + O_p + \frac{B_t}{r}\right] \frac{1}{1 - e^{pt}} \tag{1}$$

where O_{ca} , O_p , B_t , r, and e^{pt} represent channel access overhead, protocol overhead, number of bits in test frame, transmission bit rate for B_t , and error rate for B_t respectively.

2.2 Multichannel and multiradio for WMN

One of the hurdles in successful implementation of wireless mesh networks is the fact that most existing WLAN systems are using single channel APs, which results in high probability of packet collision. Consequently, severe performance degradation is inevitable in terms of throughput, latency, etc., with this type of technology [2]. One way to avoid this problem is to adopt multichannel and multiradio schemes into implementing WMNs[3]. Figure 4 illustrates a WMN where multichannel multiradio technology is used to implement WMNs[3].

Figure 4 illustrates a route from a source to a destination based on multichannel technologies where each node has two different channels. First, each node exchanges ratio metrics with its neighboring nodes(1). The radio metric can differ even between the same adjacent nodes if different frequency links are in use, which is because channel state is dependent on active frequency used. When the exchange of the radio metric information is over, a node sends Route_Request message to a destination to all WLAN interface. This message passes through all WLAN nodes connected, and transmitted to the destination (2). Now, the destination selects a path having the least radio metric sums(3) and sends Route_Reply message back to the source node(4). Eventually, the source node sends data packets to the destination node following the chosen route path. This routing scheme operates in layer 2.5 of protocol stack in



Figure 4: Multichannel scheme is employed to implement a WMN

general, therefore it can run independently of multiple access schemes. Figure 5 shows one of WMNs using different channels within a wireless distributed system(WDS)[4]. By assigning different channels for ingress and egress packets, the system can reduce the probability of packet collision significantly, and thus increase system throughput. Consequently, overall latency of packets can also be significantly reduced. Figure 6 shows an experiment where system throughput is measured with respect to varying number of hops with or without noise.

As you can see in the figure, the system can maintain a fairly stable throughput even in the presence of noise and increasing hops when a multi-channel scheme is employed.

3 WMN based on IEEE 802.15.5 and 6LoWPAN standard

Unlike the previous 802.11s mesh networks, WPAN-based mesh networks have more constraints, which is summarized as "3L" constraints, Low power, Low price, and Low transmission power. There are two standardization bodies that are involved in the development of WPAN-based mesh, 6LoWPAN and IEEE 802.15.5. 6LowPAN strives to develop standards to transmit IPv6 packet over IEEE 802.15.4 WPAN networks. It mostly focuses on the standardization above the layer 2 of protocol stack. 6LoWPAN has been viewed as more promising technology than ZigBee [5] because it has loose constraints in terms of mobility of nodes. It can be the best WPAN solution for a stationary environment where the mobility of the node is negligible. A list of major work items includes IP adaptation/Packet Format, interoperability, addressing schemes and address management, network management, routing in dynamically adaptive topologies, security including setup and maintenance, application programming interface, discovery (of devices, of services, etc.), and implementation considerations. Many contributions have been presented so far for the standard, mostly assuming that lower layer can support the high layer in reliable fashion. However, with the current IEEE 802.15.4-like MAC



Figure 5: Example of wireless mesh network based on multichannel scheme



Figure 6: Throughput analysis of a WMN depending on different number of hops


Figure 7: Mesh router assisted routing

protocols, it is hard to realize WMNs because of inherent difficulties of beacon scheduling and shortage of address space. We address this later with more details. Here we introduce one of the routing methods that have been presented by a Korean delegate to 6 LoWPAN. Figure 7 shows a diagram of a WMN where there are four mesh routers and groups of sensor nodes surrounding them.

The idea is to have a wireline connection among the mesh routers, and each sensor node is controlled by one specific mesh router. This idea has several problems as follows. First when we have a large number of sensor nodes, address space will quickly run out. Second, nodes near the routers have high rate of communication with lower layer nodes because packets from lower layer nodes must pass through higher layer nodes inevitably and therefore run out battery faster than lower layer nodes. This can create a domino effect on lower layer nodes. Nodes in the lower layer keep sending packets to the high layer nodes until they get responses from them. Third, without a proper beacon scheduling, it is unlikely to have normal communications between nodes because of collision of the beacons in the case of multi-hop communications. All these issues relate to how to design MAC and PHY layers efficiently as mentioned before.

Meanwhile, the IEEE 802.15.5 Mesh Networking Task Group was formed to derive PHY and MAC standards to enable WPAN mesh networking. The use of mesh routing technologies can be used to overcome inherent power limitation of WPANs. The mesh networking environment can increase the coverage of WPANs and provides shorter links to nodes. This is particularly advantageous for ultra wideband (UWB) communication that is significantly sensitive to distance due to high data rate. The shorter links from the mesh networking environment significantly increase throughput.

Recently, a proposal is presented, addressing how to improve the performance of IEEE 802.15.4-like MAC [6]. The author maintains a basic paradigm of IEEE 802.15.4 MAC and PHY, and suggests several novel ideas to improve the performance of the MAC. Those ideas include a new beacon scheduling method, a new short addressing scheme, and a method to support seamless mobility. We explain this new beacon scheduling method here because it is the key to improving the performance of MAC. The new beacon scheduling algorithm is



Figure 8: Structure of the superframe and an example of scheduling table

based on the idea that a node could avoid collision of beacons if the node knows the timings of beacon transmissions of its neighbor nodes and those of the neighbor's neighbor nodes. A node keeps updating a scheduling table that is composed of its neighbor nodes, neighbor's neighbor nodes, depths from mother node, beacon timing to avoid, and its own beacon timing. We illustrate one example to explain the operation of the beacon scheduling in Figure 8.

In Figure 8, node 2 knows that it has node 1 and 3 as its neighbors, and its neighbors do not have anymore neighbor nodes(denoted as "x" in the table). Therefore it is one depth away from the parent node 1. The node 2 also knows it can avoid the collision of beacons from node 1 and 3 by setting 2 as its beacon timing. As shown in Figure 8, the author adopts a Beacon Only Period (BOP) in the superframe where the beacon timings of all nodes are scheduled. Figure 9 further illustrates the beacon scheduling when there are 20 nodes in the network.

This algorithm is recently adopted as a candidate standard for WPAN mesh in the U-city forum in Korea.

4 Research Issues

As studied so far, there are many research issues with regards to wireless mesh networking technologies, and most of them are quite challenging. In this section, we summarize those major research issues. First, wireless mesh networking technologies are characterized as having a dynamic topology that enables us to construct self-configuring and self-healing wireless mesh networks. Given this fact, it is imperative to develop an efficient routing protocol with a novel link evaluation metric. Second, as in [6], an effort to achieve efficient routing protocol can be far-fetched without a proper beacon scheduling scheme in the MAC layer. More contributions on this are highly anticipated. Meanwhile the performance of [6] needs to be further evaluated along with simulation and analytical results. Third, nodes can be movable, so one needs to take mobility issues into consideration as well in the design step of MAC protocols. Fourth, more studies on proper radio technologies are required. There are lots of different types of radio technologies to date. As many people have agreed, new approaches like cross-layer approach



Figure 9: Example of beacon scheduling in the case of 20 nodes

can be more efficient.

5 Concluding Remark

This article discusses technical issues related to wireless mesh networking technologies, which includes issues on topology formation, routing, MAC, and PHY schemes. Research activities of three standardization bodies, IEEE 802.11s, IEEE 802.15.5, and 6 LoWPAN, are also discussed. One point to make is that it is not easy to realize a proper mesh without advances in the MAC and PHY technologies, which is underestimated in comparison to those from the higher layers.

References

- D. M. Shrestha and Y.-B. Ko, "Enhanced Topology Formation Protocol for IEEE 802.11 WLAN based Mesh Networks," COMSWARE2007, Jan. 2007.
- [2] Q. Shen and X. Fang," A Multi-metric AODV Routing in IEEE 802.11s", ICCT06
- [3] H. Aoki, Sh Takeda, K. Yagyu and A. Yamada, "IEEE 802.11s Wireless LAN Mesh Network Technology," NTT DoCoMo Technical Journal Vol. 8 No.2, pp13-21
- [4] www.strixsystems.com
- [5] ZigBee Draft Version 1.0
- [6] Ho-In Jeon, "An Analysis of 802.15.4-Based Mesh Network Architecture," 15-06-0266-00-0005-analysis-802-15-4-based-mesh-network-architecture, IEEE 802.15.5 Wireless PAN Mesh Network Task Group Face-to-Face meeting, April 18, 2006.

A Polling Model with an Autonomous Server

Roland de Haan, Richard J. Boucherie and Jan-Kees van Ommeren University of Twente, Enschede, The Netherlands

Abstract

Polling models are used as an analytical performance tool in several application areas. In these models, the focus often is on controlling the operation of the server as to optimize some performance measure. For several applications, controlling the server is not an issue as the server moves independently in the system. We present the analysis for such a polling model with a so-called autonomous server. In this model, the server remains for an exogenous random time at a queue, which also implies that service is preemptive. Moreover, in contrast to most of the previous research on polling models, the server does not immediately switch to a next queue when the current queue becomes empty, but rather remains for an exponentially distributed time at a queue. The analysis is based on considering imbedded Markov chains at specific instants. A system of equations for the queue-length distributions at these instant is given and solved for. Besides, we study to which extent the queues in the polling model are independent and identify parameter settings for which this is indeed the case. These results may be used to approximate performance measures for complex multi-queue models by analyzing a simple singlequeue model.

1 Introduction

Polling systems are multi-queue systems with a single server. Typically, the server visits a queue, offers service to (a part of) the customers present at this queue, and then moves to a next queue. The specific details of the system may lead to quite distinct polling models. Polling models are typically characterized by: (i) the arrival process of the customers to the system (Poisson or more general), (ii) the service requirements of the customers, (iii) the servicing policy of the server (exhaustive, gated, k-limited, etc.), (iv) the visit order of the server, (v) the switch-over times of the server between visits to the queues. An excellent survey on a broad class of polling models is [1]. Applications of polling models are ubiquitous. For instance, traffic light systems, multiple-access protocols for communication networks (e.g., IEEE 802.11) and product-assembly systems can be modelled as a polling system.

In most of the (applications of) polling models, the server is assumed controllable. The goal is then to limit the time a server spends idle at a queue while there is still work in the system. Contrary, in this article we assume that the server behaves autonomously (and thus is uncontrollable). More precisely, we assume that the server spends an exponentially distributed period of time at a queue independent of the distribution of the customers present at each queue. Another consequence of the autonomous server is that the services are subject to preemption. Applications of such polling models arise for instance in the context of wireless ad hoc networks in which cars, pedestrians or other moving objects which carry wireless equipment are used as communication hop.

The class of polling models that is most closely related to our model is the class of so-called time-limited polling models [2, 3, 4, 5]. Leung [2] analyzes a time-limited model in which the server remains an exponential time at a queue but service is non-preemptive. Preemption is

considered for a deterministic time-limited model by De Souza e Silva et al. [4] for Poisson arrivals and by Frigui and Alfa [3] for Markovian Arrival Processes. In each of these models the server is *impatient* and leaves a queue as soon as it becomes empty. A specific application of a time-limited model to a timed token protocol^{*} can be found in [5].

Standard polling models assume that the server moves to a next queue once the queue becomes empty. However, there also exists analytical work on models with a server that remains at a queue even when it becomes empty. These models are often referred to as patient server models or stopping server models. The works of Eisenberg [6] and Borst [7] analyze several strategies for the server once the complete system becomes empty as to optimize some system performance measure. More recently, Boxma et al. [8, 9] consider a single-queue vacation model and a two-queue polling model in which the server upon arriving at an empty queue waits patiently for a certain duration before leaving again. We note that in the latter two-queue polling model (contrary to the models in [6] and [7]) there is no notion of work conservation anymore, since the server may wait patiently at one queue while the other queue is nonempty.

The only work we know of that includes both a given (random) visit time and a patient server that does not leave before the end of the visit time is [10]. This work considers the workload process for the autonomous server model with deterministic visit times. The authors of [10] analyze each queue in isolation by considering them as an M/G/1 model with server vacations. Using an approximate analysis, several performance measures for the system are derived.

For the case of a single queue, the polling model that we will consider boils down to the unreliable server model (USM) [11]. The extension of the analysis to a two-queue polling model appears feasible when the approach of, e.g., [12] or [13] would be followed. This approach requires to solve a boundary value problem. This solution method appears an extremely difficult task for the two-queue model already, while for three or more queues analytical solutions along this direction are not anticipated.

In the first part of this article, we study a single-server polling model with $M \ge 1$ stations with infinite buffer in a stable environment. The main characteristics of the model are that the server visits a queue for a random amount of time (irrespective of the number of customers present at a queue) and that the service is preemptive. Our interest is in the queue-length distribution at various instants in time. We note that if the interest would only be in mean performance measures, then the queues could be considered in isolation. Our analytical approach builds on the work of Eisenberg [14]. We set-up a system of equations which relate the queue-length distributions at various specific instants. The solution of this system is obtained by the explicit determination of the distribution at visit completion instants via an iterative approach. This approach is similar to the approach introduced by Leung for probabilistically-limited polling models [15]. In the second part of this article, we study to which extent the queues in the polling system are independent. To this end, we consider a single queue in isolation by analyzing a USM. Next, we perform several numerical experiments to compare the results from the polling system with results based on the USM. In this way, we identify for which system parameters the queues appear "reasonably" independent.

This article is organized as follows. In Sect. 2 we describe the polling model. The analyses for the single-queue model and multi-queue model are given in Sect. 3 and Sect. 4, respectively. In Sect. 5, we study an approximation approach for the multi-queue model. The article is concluded in Sect. 6.

2 Model

We denote queue *i* by Q_i , i = 1, ..., M. Customers arrive to Q_i according to a Poisson process with arrival rate λ_i . We will throughout use the subscript *i* to refer to a queue and for convenience leave out its range (i = 1, ..., M) whenever this does not lead to ambiguity. We denote the interarrival-time distribution by I_i , with Laplace-Stieltjes Transform (LST) $\tilde{I}_i(s) = \lambda_i/(\lambda_i + s)$. A customer arriving to Q_i requires an amount of service with a general distribution X_i , with LST $\tilde{X}_i(s)$, and mean $1/\mu_i$.

A single server serves the queues at unit rate. For ease of presentation, we assume a fixed cyclic visit schedule $Q_1, Q_2, \ldots, Q_M, Q_1, Q_2$, etc., but assuming other fixed cyclic schedules (e.g., in which queues are visited multiple times per cycle) would not significantly change the analysis. The server visits Q_i for an exponential amount of time denoted by Y_i , with LST $\tilde{Y}_i(s) = \xi_i/(\xi_i + s)$. The server always remains at a queue until the (random) visit time ends, even when the queue becomes empty. In other words, the dynamics of the server are independent of the current state of the system. We assume that switch-over times of the server from Q_{i-1} to Q_i follow a general distribution C_i , with LST $\tilde{C}_i(s)$, and mean c_i . Due to the patient nature of the server, (possibly multiple) idle periods can occur during a visit. The duration of each of these periods is distributed as the interarrival time.

We assume that customers are served according to the First-In-First-Out discipline. The service (but also the idle periods) at a queue will be preempted at the end of a visit. At the beginning of the next visit, the service time will be redrawn from the original distribution; thus, we adopt the so-called *preemptive-repeat* strategy with independent repetitions.

The sequences of random variables generated from C_i , I_i , X_i and Y_i are assumed independently and identically distributed. Besides, the random variables C_i , I_i , X_i and Y_i are assumed to be mutually independent.

3 Analysis of the single-queue model

The single-queue model is in fact an unreliable server model. Alternatively, it may be considered as a vacation model with preemptive service. The first to analyze this specific model was Gaver [11] by introducing high priority (i.e., interrupting) and low priority (i.e., arriving) customers.

Here, we analyze this unreliable server model by considering a sequence of alternating *processing* and *non-processing periods*. During a processing period, the server serves customers, while during a non-processing period no customers are served. The server may break down (and thus need repair) at random points in time both during processing and non-processing periods. These *repair periods* follow a general distribution D with LST $\tilde{D}(s)$ and mean $\mathbb{E}D$. The periods between consecutive repairs, the so-called *availability periods*, are assumed exponentially distributed with mean $1/\xi$. Customers arrive to the system according to a Poisson process with rate λ . We assume further that a *preemptive-repeat* servicing strategy with independent repetitions is followed, i.e., if a service is interrupted, then the next availability period the service requirement is redrawn from the original service-time distribution.

Let us introduce some notation. We denote by $\bar{X}_G(s)$ and $\mathbb{E}[X_G]$ the LST and the mean of the generalized service time of a customer, respectively. The latter period of time is defined as the period that starts when a customer receives service for the first time and ends when the customer leaves the system. We let $\mathbb{E}[K]$ refer to the mean number of customers served during a processing period. Further, we denote by $\hat{U}(z)$ the p.g.f. of the number of customers arriving during the service time of a customer that arrives to an empty system during a repair time. The latter customer (service time) will be referred to as an exceptional first customer (service time). The load of the system is defined as ρ_G . Let us finally denote the queue-length distribution at departure instants (which equals the time-equilibrium distribution) by d_n , n = 0, 1, 2, ...

. Then, the probability generating function $P_{L_d}(z)$ of this distribution is known and given by the following theorem (see, e.g., [16]).

Theorem 1.

$$P_{L_d}(z) = \frac{1}{\mathbb{E}[K]} \cdot \frac{\tilde{X}_G(\lambda(1-z)) - z\hat{U}(z)}{\tilde{X}_G(\lambda(1-z)) - z} , \qquad (1)$$

where

$$\begin{split} \tilde{X}_G(s) &= \frac{X(\xi+s) \cdot (\xi+s)}{(\xi+s) - \xi(1-\tilde{X}(\xi+s))\tilde{D}(s)} ,\\ \hat{U}(z) &= \tilde{X}_G(\lambda(1-z)) \cdot \frac{\lambda z + \xi(\tilde{D}(\lambda(1-z)) - \tilde{D}(\lambda))}{z(\lambda + \xi(1-\tilde{D}(\lambda)))} \\ \mathbb{E}[K] &= \frac{1}{1-\rho_G} \cdot \frac{\lambda(1+\xi\mathbb{E}[D])}{\lambda + \xi(1-\tilde{D}(\lambda))} . \end{split}$$

Let us denote by V^* the processing time given that the service is interrupted and by D the repair time. Further, we denote by X^* the service time given that the service is successful. Let V_i^* be i.i.d. copies of V^* , D_i be i.i.d. copies of D, and N a random variable denoting the number of interruptions during a service. Then, the generalized service time X_G satisfies:

$$X_G = X^* + \sum_{i=1}^N (V_i^* + D_i)$$

Let $\tilde{X}(s)$ be the LSTs of the original service time.

Lemma 1.

$$\tilde{X}_G(s) = \mathbb{E}[e^{-sX_G}] = \frac{X(\xi+s) \cdot (\xi+s)}{(\xi+s) - \xi(1-\tilde{X}(\xi+s))\tilde{D}(s)},$$

Proof. The random variable *N* is geometrically distributed with success probability $\xi/(\mu + \xi)$. The result for $\tilde{X}_G(s)$ follows by conditioning on *N* and some elementary calculus.

The service time U of an exceptional first customer is given by

$$U = X_G + R_D \cdot \mathbf{1}_{\{R_D\}},\tag{2}$$

where R_D denotes the residual repair time when the first customer arrives to the queue and $\mathbf{1}_{\{R_D\}}$ is the indicator function of the event that a customer which arrives to an empty system arrives during a repair time.

Lemma 2. The p.g.f. of the number of arrivals during a service time U of an exceptional first customer is given by

$$\hat{U}(z) = \mathbb{E}[z^{N(U)}] = \mathbb{E}[z^{N(X_G)}] \cdot \mathbb{E}[z^{N(R_D)} \mathbf{1}_{\{R_D\}}],$$

where

$$\begin{split} \mathbb{E}[z^{N(X_G)}] &= \tilde{X}_G(\lambda(1-z)) ,\\ \mathbb{E}[z^{N(R_D)} \mathbf{1}_{\{R_D\}}] &= 1 - (1 - \mathbb{E}[z^{N(R_D)}]) \cdot \frac{\xi \cdot (1 - \tilde{D}(\lambda))}{(\lambda + \xi) - \xi \cdot \tilde{D}(\lambda)} . \end{split}$$

Proof. By Eq. (2), we can directly write for the p.g.f. of the number of arrivals during U, $\hat{U}(z)$,

$$\hat{U}(z) = \mathbb{E}[z^{N(U)}] = \mathbb{E}[z^{N(X_G) + N(R_D) \cdot \mathbf{1}_{\{R_D\}}}] = \mathbb{E}[z^{N(X_G)}] \cdot \mathbb{E}[z^{N(R_D)}\mathbf{1}_{\{R_D\}}].$$

Due to the Poisson arrival process, we have:

$$\mathbb{E}[z^{N(X_G)}] = \tilde{X}_G(\lambda(1-z)) \; .$$

Let us denote by $\mathbb{P}(XFS)$ the probability that an arbitrary arriving customer is indeed an exceptional first customer. Then, we can write:

$$\mathbb{E}[z^{N(R_D)} \mathbf{1}_{\{R_D\}}] = \mathbb{E}[z^{N(R_D)}] \cdot \mathbb{P}(\mathsf{XFS}) + 1 \cdot (1 - \mathbb{P}(\mathsf{XFS}))$$

= 1 - (1 - \mathbb{E}[z^{N(R_D)}]) \cdot \mathbb{P}(\mathsf{XFS}).

The p.g.f. $\mathbb{E}[z^{N(R_D)}]$ can be found by conditioning on the event of at least one arrival during the repair time and is given by:

$$\mathbb{E}[z^{N(R_D)}] = \frac{\mathbb{E}[z^{N(D)} \mid N(D) \ge 1]}{z} = \frac{\tilde{D}(\lambda(1-z)) - \tilde{D}(\lambda)}{z(1-\tilde{D}(\lambda))} .$$

The probability $\mathbb{P}(XFS)$ is obtained by considering its counterpart $\mathbb{P}(\overline{XFS}) = 1 - \mathbb{P}(XFS)$. The sequence of instants at which the queue becomes empty forms a renewal process. Note that the queue becomes empty only during an availability period and that the residual availability time is still exponentially distributed. Thus, by considering the first customer arriving after a renewal point, we can write for $\mathbb{P}(\overline{XFS})$:

$$\begin{split} \mathbb{P}(\overline{XFS}) &= \mathbb{P}(\text{arrival in processing period}) \\ &+ (1 - \mathbb{P}(\text{arrival during processing period})) \\ &\cdot \mathbb{P}(\text{no arrival in the following repair period}) \cdot \mathbb{P}(\overline{XFS}) \,. \end{split}$$

It follows that:

$$\mathbb{P}(\overline{XFS}) = \frac{\lambda}{\lambda + \xi} + \frac{\xi}{\lambda + \xi} \cdot \overline{D}(\lambda) \cdot \mathbb{P}(\overline{XFS}) = \frac{\lambda}{\lambda + \xi(1 - \overline{D}(\lambda))},$$

and as a result:

$$\mathbb{P}(XFS) = 1 - \mathbb{P}(\overline{XFS}) = \frac{\xi \cdot (1 - \tilde{D}(\lambda))}{\lambda + \xi(1 - \tilde{D}(\lambda))} .$$

Finally, we consider the mean number of served customers during a processing period.

Lemma 3.

$$\mathbb{E}[K] = \frac{1}{1 - \rho_G} \cdot \frac{\lambda(1 + \xi \mathbb{E}[D])}{\lambda + \xi(1 - \tilde{D}(\lambda))} ,$$

where

$$\rho_G = \lambda \cdot \mathbb{E}[X_G] \, .$$

Proof. The term $\mathbb{E}[K]$ follows directly by inserting z = 1 in Eq. (1)

4 Analysis of the multi-queue model

The analysis of the multi-queue model builds on the work of Eisenberg. Eisenberg [14] considers a polling model with a non-patient server and non-preemptive service. For this model, the queue-length distribution is determined at visit beginning, visit completion, service beginning, and service completion instants by studying the imbedded Markov chains defined at these instants. The fundamental relation in the analysis is the relation that counts the number of events with state n that occurred until time t [14, Eq.(4)]. In our work, we extend this relation for the polling model under consideration and we will use this as a building block for obtaining the queue-length distribution at various instants. We will first discuss the stability conditions of the system in Sect. 4.1. Next, in Sect. 4.2, we treat the extended counting relation in more detail. This counting relation is not sufficient to determine the queue-length distribution at all instants. To this end, we derive additional relations between the random variables in Sect. 4.3. However, even with these additional relations we still do not have enough information to solve our model completely. In Sect. 4.4, we will resolve this problem by deriving an explicit expression for the queue-length distribution at visit completion instants. This approach is based on work of Leung [15] for a probabilistically-limited polling model. Finally, we present the steady-state probabilities for our model in Sect. 4.5.

4.1 Stability condition

The polling system is stable if each customer in the system can be served in a finite period of time. Contrary to many other polling models, we must consider stability on a per-queue basis as service capacity cannot be exchanged between the queues. We say that the system is stable if and only if all the queues in the system are stable.

For an individual queue to be stable, we must have that on average the number of customer arrivals per cycle is smaller than the number of customers that can be served at most per cycle. The latter random variable for Q_i will be denoted by S_{\max}^i and is geometrically distributed (due to the exponential visit times), i.e.

$$\mathbb{P}(S_{\max}^{i} = k) = p_{i}(1 - p_{i})^{k}, \ k = 0, 1, 2, \dots,$$

where $p_i = \mathbb{P}(\text{service is preempted} | \text{s.b. at } Q_i) = 1 - \tilde{X}_i(\xi_i)$. Here we use s.b. as short for service beginning. Thus, the stability condition for Q_i then reads:

$$\rho_{G,i} := \frac{\mathbb{E}[\text{arrivals per cycle to } Q_i]}{\mathbb{E}[S_{\max}^i]} = \lambda_i \sum_j (\frac{1}{\xi_j} + c_j) \cdot \frac{1 - X_i(\xi_i)}{\tilde{X}_i(\xi_i)} < 1 \ .$$

4.2 A relation for the queue-length distribution at various instants

We set up a relation for the number of occurrences of specific events. Apart from the events defined in [14], we define a number of additional events. We introduce events related to the start and the completion of an idle period. These events do not appear in Eisenberg's model as in his model the server leaves a queue as soon as it becomes empty. Moreover, we introduce events related to the interruption of a service or idle period due to the end of a server visit. Let us denote by n_i the number of customers at Q_i . Next, we can define the following variables which all refer to the number of the given events with state $\mathbf{n} = (n_1, \ldots, n_M)$ that occur in (0, t)

at Q_i :

 $\omega^{i}(t; \mathbf{n})$, service beginnings; $\pi^{i}(t; \mathbf{n})$, successful service completions; $\pi^{i}_{\star}(t; \mathbf{n})$, interrupted services; $\alpha^{i}(t; \mathbf{n})$, visit beginnings; $\beta^{i}(t; \mathbf{n})$, visit completions; $a^{i}(t; \mathbf{n})$, idle period beginnings; $b^{i}(t; \mathbf{n})$, idle period completions; $b^{i}_{\star}(t; \mathbf{n})$, interrupted idle periods.

We note that **n** refers to the number of customers present in the system (either waiting or in service) immediately after the specific event occurred. These variables are related in the following way for $t \ge 0$:

$$[\pi^{i}(t;\mathbf{n}) + \pi^{i}_{*}(t;\mathbf{n})] + \alpha^{i}(t;\mathbf{n}) + [b^{i}(t;\mathbf{n}) + b^{i}_{*}(t;\mathbf{n})] = \omega^{i}(t;\mathbf{n}) + \beta^{i}(t;\mathbf{n}) + a^{i}(t;\mathbf{n}), \ \forall_{n\in\mathbb{N}^{M}}.$$
 (3)

This counting relation should be read as follows. At each instant that one of the events present at the l.h.s. of (3) with state n occurs, also exactly one event with the same state n at the r.h.s. occurs. We note that the end of a server visit always corresponds to an interruption and vice versa. Therefore, we can isolate these events and break up Eq. (3) into:

$$\pi^i_*(t;\mathbf{n}) + b^i_*(t;\mathbf{n}) = \beta^i(t;\mathbf{n}), \qquad (4)$$

$$\pi^{i}(t;\mathbf{n}) + \alpha^{i}(t;\mathbf{n}) + b^{i}(t;\mathbf{n}) = \omega^{i}(t;\mathbf{n}) + a^{i}(t;\mathbf{n}).$$
(5)

Let us define imbedded Markov chains each corresponding to instants at which one of the counting processes increases. Each state in a Markov chain is uniquely defined by the position i of the server (i = 1, ..., M) and $\mathbf{n} = (n_1, ..., n_M) \in \{0, 1, ...\}^M$, the number of customers present in the system at a certain instant. We define the steady-state probabilities for each event type by dividing the number of events with state \mathbf{n} that occurred until t by the total number of the events until t, and then taking the limit for t to infinity. It can be seen that all these limits indeed do exist by noting first that the quantities in the denominator all go to infinity with probability one. Next, by using an ergodicity theorem [17], it can be shown that all limits exist with probability one. Thus, the probabilities are correctly defined as follows:

where

$$\begin{aligned} \alpha^{i}(t) &= \sum_{\mathbf{n}} \alpha^{i}(t;\mathbf{n}), \qquad \beta^{i}(t) &= \sum_{\mathbf{n}} \beta^{i}(t;\mathbf{n}), \qquad b^{i}(t) &= \sum_{\mathbf{n}} b^{i}(t;\mathbf{n}), \\ b^{i}_{*}(t) &= \sum_{\mathbf{n}} b^{i}_{*}(t;\mathbf{n}), \qquad a^{i}(t) &= \sum_{\mathbf{n}} a^{i}(t;\mathbf{n}), \qquad \omega(t) &= \sum_{i} \sum_{\mathbf{n}} \omega^{i}(t;\mathbf{n}), \\ \pi(t) &= \sum_{i} \sum_{\mathbf{n}} \pi^{i}(t;\mathbf{n}), \quad \pi_{*}(t) &= \sum_{i} \sum_{\mathbf{n}} \pi^{i}_{*}(t;\mathbf{n}). \end{aligned}$$

Notice that (hereby following [14]) we have that all probabilities are conditioned on Q_i except for ω_n^i, π_n^i and $\pi_{n,*}^i$. Along with the steady-state probabilities, let us also define the corresponding p.g.f.'s as follows:

 $\begin{array}{rcl} \alpha^{i}(\mathbf{z}) &=& \sum_{\mathbf{n}} \alpha_{\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, & \beta^{i}(\mathbf{z}) &=& \sum_{\mathbf{n}} \beta_{\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, & b^{i}(\mathbf{z}) &=& \sum_{\mathbf{n}} b_{\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, \\ b_{*}^{i}(\mathbf{z}) &=& \sum_{\mathbf{n}} b_{*,\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, & a^{i}(\mathbf{z}) &=& \sum_{\mathbf{n}} a_{\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, & \omega(\mathbf{z}) &=& \sum_{\mathbf{n}} \omega_{\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, \\ \pi(\mathbf{z}) &=& \sum_{\mathbf{n}} \pi_{\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, & \pi_{*}(\mathbf{z}) &=& \sum_{\mathbf{n}} \pi_{*,\mathbf{n}}^{i} \cdot \mathbf{z}^{\mathbf{n}}, \end{array}$

where $\mathbf{z}^{\mathbf{n}} := z_1^{n_1} \cdots z_M^{n_M}$.

Next, we divide Eqs. (4) and (5) by $\pi(t)$ and take the limit of $t \to \infty$, yielding:

$$\pi_{*,\mathbf{n}}^{i} \lim_{t \to \infty} [\pi_{*}(t)/\pi(t)] + b_{*,\mathbf{n}}^{i} \lim_{t \to \infty} [b_{*}^{i}(t)/\pi(t)] = \beta_{\mathbf{n}}^{i} \lim_{t \to \infty} [\beta^{i}(t)/\pi(t)], \quad (6)$$

$$\pi_{\mathbf{n}}^{i} \lim_{t \to \infty} [\pi(t)/\pi(t)] + \alpha_{\mathbf{n}}^{i} \lim_{t \to \infty} [\alpha^{i}(t)/\pi(t)] + b_{\mathbf{n}}^{i} \lim_{t \to \infty} [b^{i}(t)/\pi(t)] =$$
(7)

$$\omega_{\mathbf{n}}^{i} \lim_{t \to \infty} [\omega(t)/\pi(t)] + a_{\mathbf{n}}^{i} \lim_{t \to \infty} [a^{i}(t)/\pi(t)] .$$

It is readily verified by elementary renewal theory (see, e.g., [18, Prop. 3.3.1]) that under our model assumptions all these limits indeed exist with probability one.

Let us introduce some notation. We denote by $p_{pr,X}$ the probability of an arbitrary service in the system being preempted and by $p_{pr,I}^i$ the probability of an idle period at Q_i being preempted. We define $\kappa_i := \lim_{t\to\infty} [a^i(t)/\pi(t)]$. Further, we denote by $\mathbb{E}[C]$ the mean cycle time of the server. This enables us to present the following theorem.

Theorem 2. The p.g.f.'s of the queue-length distribution at Q_i at various imbedded instants in a polling model with an autonomous server are related as follows:

$$\frac{p_{pr,X}}{1-p_{pr,X}} \cdot \pi^i_*(z) + \kappa_i \cdot p^i_{pr,I} \cdot b^i_*(z) = \gamma \cdot \beta^i(z) ,$$

$$\pi^i(z) + \gamma \cdot \alpha^i(z) + \kappa_i \cdot (1-p^i_{pr,I}) \cdot b^i(z) = \frac{\omega^i(z)}{1-p_{pr,X}} + \kappa_i \cdot a^i(z) ,$$

where

$$p_{pr,X} = 1 - \frac{\sum_{j} \lambda_{j}}{\sum_{j} \lambda_{j} / \tilde{X}_{j}(\xi_{j})},$$

$$p_{pr,I}^{i} = 1 - \tilde{I}_{i}(\xi_{i}), \ i = 1, \dots, M,$$

$$\kappa_{i} = \frac{1}{p_{pr,I}^{i}} \cdot \left(\gamma - \frac{\lambda_{i}}{\sum_{j} \lambda_{j}} \cdot \frac{1 - \tilde{X}_{i}(\xi_{i})}{\tilde{X}_{i}(\xi_{i})}\right), \ i = 1, \dots, M,$$

$$\gamma = \frac{1}{\sum_{j} \lambda_{j} \mathbb{E}[C]}.$$

We will now present several lemmas and defer the proof of the theorem until the end of this section.

Lemma 4.

$$\lim_{t \to \infty} [\alpha^i(t)/\pi(t)] = \lim_{t \to \infty} [\beta^i(t)/\pi(t)] = \frac{1}{\sum_j \lambda_j \mathbb{E}[C]}$$

Proof. Let us first focus on $\lim_{t\to\infty} [\alpha^i(t)/\pi(t)]$, i.e., the limit of the ratio of the number of visit beginnings at Q_i and the total number of service completions. Consider an arbitrary cycle starting and ending with the server arriving to Q_i . The average number of visit beginnings at

 Q_i per cycle is exactly one. The average total number of service completions per cycle is equal to the average total number of arrivals per cycle (assuming a stable system). Hence,

$$\lim_{t \to \infty} [lpha^i(t)/\pi(t)] = rac{1}{\sum_j \lambda_j \cdot \mathbb{E}[C]} \; ,$$

where for $\mathbb{E}[C]$, the mean cycle time, we have:

$$\mathbb{E}[C] = \sum_{j} \left(\frac{1}{\xi_j} + c_j \right) \; .$$

Further, notice that the number of visit completions, $\beta^i(t)$, differs at most one from the number of visit beginnings, $\alpha^i(t)$, for any $t \ge 0$. Therefore, we have that $\lim_{t\to\infty} [\beta^i(t)/\pi(t)] = \lim_{t\to\infty} [\alpha^i(t)/\pi(t)]$.

Lemma 5.

$$\lim_{t \to \infty} [\omega(t)/\pi(t)] = \frac{1}{1 - p_{pr,X}},$$
$$\lim_{t \to \infty} [\pi_*(t)/\pi(t)] = \frac{p_{pr,X}}{1 - p_{pr,X}}.$$

Proof. The $\lim_{t\to\infty} [\omega(t)/\pi(t)]$ is defined as the limit of the ratio of the total number of service beginnings and the total number of (successful) service completions. The numerator and denominator are related via the probability of an arbitrary service being preempted, $p_{pr,X}$. More precisely,

$$rac{\pi(t)}{\omega(t)} = 1 - p_{pr,X}, ext{ for } t o \infty \ .$$

Similar to the relation between $\alpha^i(t)$ and $\beta^i(t)$, we note that $\omega(t)$ and $\pi(t) + \pi_*(t)$ differ at most one for $t \ge 0$. Therefore, we can write:

$$\lim_{t \to \infty} [\pi_*(t)/\pi(t)] = \lim_{t \to \infty} [(\omega(t) - \pi(t))/\pi(t)] = \frac{p_{pr,X}}{1 - p_{pr,X}}.$$

Lemma 6.

$$egin{array}{lll} \lim_{k o \infty} \left[b^i(t)/\pi(t)
ight] &= \kappa_i \cdot \left(1 - p^i_{pr,I}
ight) \,, \ \lim_{k o \infty} \left[b^i_*(t)/\pi(t)
ight] &= \kappa_i \cdot p^i_{pr,I} \,. \end{array}$$

Proof. Recall that we set $\lim_{t\to\infty} [a^i(t)/\pi(t)] =: \kappa_i$, where κ_i is a constant yet to be determined. These limits do not appear to have a simple interpretation, but we can relate them to limits for other events. The number of events $a^i(t)$ and $b^i(t)$ are related as follows:

$$rac{b^i(t)}{a^i(t)}=1-p^i_{pr,I}, ext{ for } t o\infty \ ,$$

where $p_{pr,I}^{i}$, the probability that an idle period at Q_{i} is preempted, depends on *i*, and is given by:

$$p_{pr,I}^i = 1 - \tilde{I}_i(\xi_i) \; .$$

Analogously, $a^{i}(t)$ and $b^{i}_{*}(t)$ are related via:

$$b^i_*(t) = a^i(t) \cdot p^i_{pr,I}, \text{ for } t \to \infty.$$

Proof of Theorem 2. The presented equations follow by first determining the limit expressions in Eqs. (6) and (7). The limit expressions are derived in the Lemmas above. However, these expressions still contain the unknowns $p_{pr,X}$ and κ_i , i = 1, ..., M.

For the service preemption probability $p_{pr,X}$, we obtain:

$$p_{pr,X} = \sum_{j} \mathbb{P}(\text{service is preempted } | \text{ s.b. at } Q_j) \cdot \mathbb{P}(\text{s.b. at } Q_j | \text{ s.b. at some queue})$$

$$= \sum_{j} (1 - \tilde{X}_j(\xi_j)) \cdot \mathbb{P}(\text{s.b. at } Q_j | \text{ s.b. at some queue})$$

$$= \frac{\sum_{j} \lambda_j (1 - \tilde{X}_j(\xi_j)) / \tilde{X}_j(\xi_j)}{\sum_k \lambda_k / \tilde{X}_k(\xi_k)} = 1 - \frac{\sum_{j} \lambda_j}{\sum_j \lambda_j / \tilde{X}_j(\xi_j)}.$$

Here we use s.b. as short for service beginning and also use that:

$$\mathbb{P}(\text{s.b. at } Q_i \mid \text{s.b. at some queue}) = \frac{\lambda_i / (1 - \mathbb{P}(\text{serv. at } Q_i \text{ is preempted } \mid \text{s.b. at } Q_i))}{\sum_j \lambda_j / (1 - \mathbb{P}(\text{serv. at } Q_j \text{ is preempted } \mid \text{s.b. at } Q_j))} \\ = \frac{\lambda_i / \tilde{X}_i(\xi_i)}{\sum_j \lambda_j / \tilde{X}_j(\xi_j)} . \tag{8}$$

Notice that multiple service beginnings may correspond to a single customer.

The unknown κ_i , i = 1, ..., M, can be found from Eq. (7) (or alternatively from Eq. (6)) by inserting all the limit expressions and summing both sides over **n**. After several rearrangements and using that

$$\sum_{\mathbf{n}} \pi_{*,\mathbf{n}}^{i} = \mathbb{P}(\text{s.i. at } Q_{i} \mid \text{s.i. at some queue}) = \frac{\lambda_{i} / \tilde{X}_{i}(\xi_{i}) - \lambda_{i}}{\sum_{j} \left(\lambda_{j} / \tilde{X}_{j}(\xi_{j}) - \lambda_{j}\right)},$$

where we use s.i. as short for service interruption, we eventually obtain:

$$\kappa_i = rac{1}{p_{pr,I}^i} \left(\gamma - rac{\lambda_i}{\sum_j \lambda_j} \cdot rac{1 - ilde{X}_i(\xi_i)}{ ilde{X}_i(\xi_i)}
ight)$$

The final step is to write these equations in terms of p.g.f.'s by multiplication with z^n and summation over **n**.

4.3 Additional relations for the queue-length distributions at different instants

We need additional relations to obtain the queue-length distributions at the different instants defined. Eisenberg [14] presents relations between $\pi^i(\mathbf{z})$ and $\omega^i(\mathbf{z})$ for the non-patient server model with non-preemptive services. We show that with a minor modification this relation can be used to relate both $\pi^i(\mathbf{z})$ and $\omega^i(\mathbf{z})$ and $\pi^i_*(\mathbf{z})$ and $\omega^i(\mathbf{z})$ in our model. Moreover, relations between $a^i(\mathbf{z})$ and $b^i(\mathbf{z})$ and between $a^i(\mathbf{z})$ and $b^i_*(\mathbf{z})$ can be established in a similar fashion. Finally, for completeness we repeat the relation from [14] between $\alpha^i(\mathbf{z})$ and $\beta^{i-1}(\mathbf{z})$.

Relations between service events

Recall that $\omega^i(\mathbf{z})$, $\pi^i_*(\mathbf{z})$ and $\pi^i(\mathbf{z})$ refer to the number of customers at all queues at instants of service beginning, service interruption and successful service completion, respectively. Let us first consider the relation between $\omega^i(\mathbf{z})$ and $\pi^i(\mathbf{z})$. We note that every successful service completion instant has a corresponding service beginning instant, while the correspondence the other way round is not true due to preemption (which is caused by the exogenously determined visit times of the server). Notice that the fact whether a service will get interrupted does not depend on the queue-length distribution at the start of a service.

Unfortunately, we cannot relate $\omega^i(\mathbf{z})$ and $\pi^i(\mathbf{z})$ in the straightforward manner as was done by Eisenberg. In particular, as these p.g.f.'s are not conditioned on the position of the server, we cannot readily describe the number of arriving customers during a completed service. Eisenberg could do so because the conditional and unconditional p.g.f.'s in his model are related identically. This is due to the non-preemption assumption which ensures that the long-term fraction of all service beginnings that occur at Q_i and the long-term fraction of all service completions that occur at Q_i are equal.

Recall first the definitions of $\omega^i(\mathbf{z})$ and $\pi^i(\mathbf{z})$:

$$\begin{split} \omega^{i}(\mathbf{z}) &= \sum_{n_{1}} \cdots \sum_{n_{M}} z_{1}^{n_{1}} \cdots z_{M}^{n_{M}} \mathbb{P}(\mathbf{N} = \mathbf{n} \cap \text{s.b. at } Q_{i}) ,\\ \pi^{i}(\mathbf{z}) &= \sum_{n_{1}} \cdots \sum_{n_{M}} z_{1}^{n_{1}} \cdots z_{M}^{n_{M}} \mathbb{P}(\mathbf{N} = \mathbf{n} \cap \text{s.c. at } Q_{i}) , \end{split}$$

where s.c. is used as short for service completion. Then, to circumvent the use of unconditional p.g.f.'s, we define $\omega_c^i(\mathbf{z})$ and $\pi_c^i(\mathbf{z})$ as follows.

$$\begin{split} \omega^{i}(\mathbf{z}) &= \sum_{n_{1}} \cdots \sum_{n_{M}} z_{1}^{n_{1}} \cdots z_{M}^{n_{M}} \cdot \mathbb{P}(\mathbf{N} = \mathbf{n} \mid \text{s.b. at } Q_{i}) \mathbb{P}(\text{s.b. at } Q_{i} \mid \text{s.b. at some queue}) \\ &=: \omega_{c}^{i}(\mathbf{z}) \cdot \mathbb{P}(\text{s.b. at } Q_{i} \mid \text{s.b. at some queue}) , \\ \pi^{i}(\mathbf{z}) &= \sum_{n_{1}} \cdots \sum_{n_{M}} z_{1}^{n_{1}} \cdots z_{M}^{n_{M}} \cdot \mathbb{P}(\mathbf{N} = \mathbf{n} \mid \text{s.c. at } Q_{i}) \mathbb{P}(\text{s.c. at } Q_{i} \mid \text{s.c. at some queue}) \\ &=: \pi_{c}^{i}(\mathbf{z}) \cdot \mathbb{P}(\text{s.c. at } Q_{i} \mid \text{s.c. at some queue}) , \end{split}$$

where

$$\mathbb{P}(\text{s.c. at } Q_i \mid \text{s.c. at some queue}) = \frac{\lambda_i}{\sum_j \lambda_j} .$$

The latter equation follows immediately by the observation that the number of arriving customers is equal to the number of served customers for a stable system. Further, notice that $\mathbb{P}(s.b. \text{ at } Q_i \mid s.b. \text{ at some queue})$ is given in Eq. (8).

These conditional p.g.f.'s we can relate in the following manner:

$$\pi_c^i(\mathbf{z}) = \frac{X_i'(\mathbf{z})}{z_i} \cdot \omega_c^i(\mathbf{z}) , \qquad (9)$$

where the term $1/z_i$ is due to the fact that the number of customers at Q_i at a service completion instant is exactly one less than at the service beginning instant and $X'_i(\mathbf{z})$ is the p.g.f. of the number of customers that arrive at all queues during a service at Q_i that is indeed completed. The latter is given by:

$$X'_{i}(\mathbf{z}) := \mathbb{E}[\mathbf{z}^{N(X_{i})} \mid X_{i} < Y_{i}] = \frac{\mathbb{E}[\mathbf{z}^{N(X_{i})}\mathbf{1}_{\{X_{i} < Y_{i}\}}]}{\mathbb{P}(X_{i} < Y_{i})} = \frac{\tilde{X}_{i}(\xi_{i} + \sum_{j} \lambda_{j}(1 - z_{j}))}{\tilde{X}_{i}(\xi_{i})}, \quad (10)$$

where we introduced the notation N(T) to denote the number of arrivals during a random period T. The final equation follows from first conditioning on X_i and Y_i and next using that

 $\mathbb{E}[\mathbf{z}^{N(x)}]$ is Poisson distributed with parameter $\sum_{j} \lambda_j \cdot (1 - z_j) \cdot x$ for a given x. Combining the definitions of the conditional p.g.f.'s and Eq. (9), we obtain:

$$\pi_i(\mathbf{z}) = \frac{\mathbb{P}(\text{s.c. at } Q_i \mid \text{s.c. at some queue})}{\mathbb{P}(\text{s.b. at } Q_i \mid \text{s.b. at some queue})} \cdot X_i'(\mathbf{z}) \cdot \frac{\omega_i(\mathbf{z})}{z_i} . \tag{11}$$

The relation between $\pi^i_*(\mathbf{z})$ and $\omega_i(\mathbf{z})$ resembles Eq. (11):

$$\pi^{i}_{*}(\mathbf{z}) = \frac{\mathbb{P}(\text{s.i. at } Q_{i} \mid \text{s.i. at some queue})}{\mathbb{P}(\text{s.b. at } Q_{i} \mid \text{s.b. at some queue})} \cdot X^{*}_{i}(\mathbf{z}) \cdot \omega_{i}(\mathbf{z}) , \qquad (12)$$

where

$$X_i^*(\mathbf{z}) := \mathbb{E}[\mathbf{z}^{N(Y_i)} \mid X_i > Y_i] = \frac{\xi_i}{\xi_i + \sum_j \lambda_j (1 - z_j)} \cdot \frac{1 - X_i(\xi_i + \sum_j \lambda_j (1 - z_j))}{1 - \tilde{X}_i(\xi_i)} .$$

The derivation of $X_i^*(\mathbf{z})$ is done analogously to the derivation of $X_i'(\mathbf{z})$. Notice further that the term $1/z_i$ is absent in Eq. (12), since no customer departs from the queue.

Remark 1. We note that for non-preemptive service the first ratio on the r.h.s. of Eq. (11) equals one as a service beginning corresponds uniquely to a service completion. Further, in this case, we have that the term $X'_i(z)$ equals $\mathbb{E}[\mathbf{z}^{N(X_i)}]$, so that we obtain Eq. (17) of [14].

Relations between idle period events

Recall that $a^i(\mathbf{z})$, $b^i_*(\mathbf{z})$ and $b^i(\mathbf{z})$ refer to the number of customers at instants of idle period beginning, idle period interruption and idle period completion at Q_i , respectively. Let us first consider the relation between $a^i(\mathbf{z})$ and $b^i(\mathbf{z})$. We note that every idle period completion instant has a corresponding idle period beginning instant, while the correspondence the other way round is not true. This is due to the exponential visit time of the server. Whether the idle period gets interrupted only depends on the arrival process and on the distribution of the visit time of the server. In particular, it does not depend on the queue-length distribution at the start of an idle period. Therefore, we may state that the queue-length distribution at idle period beginning instants is independent of whether an idle period completion (due to an arrival to Q_i) will follow or not. Thus, we can relate the generating functions $a^i(\mathbf{z})$ and $b^i(\mathbf{z})$ by the following observations. The p.g.f. of the number of customers that arrive at all queues different from Q_i during an idle period that it is indeed completed is given by:

$$I_i'(\mathbf{z}) := \mathbb{E}[\mathbf{z}^{N(I_i)} \mid I_i < Y_i] = \frac{\tilde{I}_i(\xi_i + \sum_{j \neq i} \lambda_j(1 - z_j))}{\tilde{I}_i(\xi_i)}$$

This expression can be derived in a similar fashion as Eq. (10). Further, we note that exactly one customer arrives at Q_i at the end of the idle period. Together, this yields the following relation between $a^i(\mathbf{z})$ and $b^i(\mathbf{z})$:

$$b^i(\mathbf{z}) = I'_i(\mathbf{z}) \cdot z_i \cdot a^i(\mathbf{z})$$
 .

In the same manner, the relation between $b_*^i(\mathbf{z})$ and $a^i(\mathbf{z})$ can be established:

$$b^i_*(\mathbf{z}) = I'_i(\mathbf{z}) \cdot a^i(\mathbf{z})$$

Note that we use here that $I'_i(\mathbf{z}) = \mathbb{E}[\mathbf{z}^{N(Y_i)} | I_i > Y_i] = \mathbb{E}[\mathbf{z}^{N(I_i)} | I_i < Y_i]$. We are allowed to do so, because both Y_i and I_i are assumed exponentially distributed.

Relation between server visit events

Recall that $\alpha^i(\mathbf{z})$ and $\beta^i(\mathbf{z})$ refer to the number of customers at visit beginning instants and visit completion instants at Q_i , respectively. There exists a well-known relation (see, e.g., [14]) between the number of customers that the server leaves behind in the system at departure from Q_{i-1} and the number of customers in the system that the server finds upon arrival to Q_i . This difference is characterized by the number of arriving customers during a switch-over time from Q_{i-1} to Q_i . We denote by $C^i(\mathbf{z})$ the p.g.f. of this number, which is given by:

$$C_i(\mathbf{z}) = \tilde{C}_i(\sum_j \lambda_j(1-z_j))$$
.

Combining these two observations, we obtain the simple relation:

$$\alpha^i(\mathbf{z}) = C_i(\mathbf{z})\beta^{i-1}(\mathbf{z}) \; .$$

Altogether, we have derived $7 \cdot M$ relations between the $8 \cdot M$ p.g.f's of our interest. For a given value of z, |z| < 1, these relations are all linear and independent. Therefore, to obtain all the desired p.g.f.'s, solving explicitly for M p.g.f.'s is sufficient. This will be done below for $\beta^i(z)$, i = 1, ..., M.

4.4 Queue-length probabilities at visit completion instants via auxiliary variables

We will determine the p.g.f. of the queue-length distribution at visit completion instants, $\beta^i(\mathbf{z})$, explicitly. Notice that for the polling system under consideration, the marginal queue-length distributions can be obtained by analyzing each queue in isolation. However, the joint queue-length distribution cannot be obtained in this way due to the stochastics in the visit times of the server. Our analysis is based on an approach which was introduced by Leung [15] for the study of a probabilistically-limited polling model, and extended in [2] to a time-limited polling model. The analysis builds on the relations of Eisenberg [14] and involves setting up an iterative scheme. A key role in this iterative scheme is played by the (auxiliary) p.g.f.'s $\phi_k(\mathbf{z})$ and $\phi_k^s(\mathbf{z})$, which will be explained below. In the final step of the iteration scheme $\beta^i(\mathbf{z})$ is obtained as a simple function of $\phi_k^s(\mathbf{z})$.

We consider a tagged queue *i* and we will leave out the subscript and superscript *i* whenever it does not lead to ambiguity. We define a *service period* as a period which starts either at a visit beginning or at a service completion instant and ends with either the next service completion instant or an interruption (due to the departure of the server) whichever occurs first. We note that each service period, except for the final service period of a visit, comprises exactly one successfully completed service. Further notice that the first service period always starts at a visit beginning instant and that the final service period always ends at a visit completion instant. Let us denote by $\phi_k(\mathbf{z})$, $k \ge 1$, the p.g.f. of the number of customers at all queues at the end of the *k*th service period and service period *k* is not the final service period (i.e., service period *k* ends with a successful service completion, and service period k + 1 will occur). Similarly, we denote by $\phi_k(\mathbf{z})$, $k \ge 1$, the number of customers at all queues at the end of the *k*th service period and *k* is in fact the final service period (i.e., service period *k* will be interrupted, and service period k + 1 will not occur). Finally, we denote by $\phi_0(\mathbf{z})$ the p.g.f. of the number of customers present at the beginning of a visit. Then, $\phi_k(\mathbf{z})$ and $\phi_k^s(\mathbf{z})$, k = 1, 2, ..., are given by:

$$\begin{aligned} \phi_{k}(\mathbf{z}) &= \phi_{k-1}(\mathbf{z}) \mid_{z_{i}=0} \cdot \left(z_{i} \mathbb{E}[\mathbf{z}^{N(I)} \mathbf{1}_{\{Y>I\}}] \cdot \mathbb{E}[\mathbf{z}^{N(X)} \mathbf{1}_{\{Y>X\}}] \frac{1}{z_{i}} \right) \\ &+ (\phi_{k-1}(\mathbf{z}) - \phi_{k-1}(\mathbf{z}) \mid_{z_{i}=0}) \cdot \mathbb{E}[\mathbf{z}^{N(X)} \mathbf{1}_{\{Y>X\}}] \frac{1}{z_{i}} \\ &= \phi_{k-1}(\mathbf{z}) \mid_{z_{i}=0} \mathbb{E}[\mathbf{z}^{N(X)} \mathbf{1}_{\{Y>X\}}] \left(\mathbb{E}[\mathbf{z}^{N(I)} \mathbf{1}_{\{Y>I\}}] - \frac{1}{z_{i}} \right) + \phi_{k-1}(\mathbf{z}) \mathbb{E}[\mathbf{z}^{N(X)} \mathbf{1}_{\{Y>X\}}] \frac{1}{z_{i}} ,\end{aligned}$$

$$(13)$$

and

$$\begin{aligned} \phi_{k}^{s}(\mathbf{z}) &= \phi_{k-1}(\mathbf{z}) \mid_{z_{i}=0} \cdot \left(\mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < I\}}] + z_{i} \mathbb{E}[\mathbf{z}^{N(I)} \mathbf{1}_{\{Y > I\}}] \cdot \mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < X\}}] \right) & (14) \\ &+ (\phi_{k-1}(\mathbf{z}) - \phi_{k-1}(\mathbf{z}) \mid_{z_{i}=0}) \cdot \mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < X\}}] \\ &= \phi_{k-1}(\mathbf{z}) \mid_{z_{i}=0} \left(\mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < I\}}] + \mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < X\}}] \cdot \left(z_{i} \mathbb{E}[\mathbf{z}^{N(I)} \mathbf{1}_{\{Y > I\}}] - 1 \right) \right) \\ &+ \phi_{k-1}(\mathbf{z}) \mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < X\}}], \end{aligned}$$

where

$$\begin{split} \phi_{0}(\mathbf{z}) &= \alpha(\mathbf{z}) \\ \mathbb{E}[\mathbf{z}^{N(I)} \mathbf{1}_{\{Y > I\}}] &= \tilde{I}_{i}(\xi_{i} + \sum_{j \neq i} \lambda_{j}(1 - z_{j})) \\ \mathbb{E}[\mathbf{z}^{N(X)} \mathbf{1}_{\{Y > X\}}] &= \tilde{X}_{i}(\xi_{i} + \sum_{j} \lambda_{j}(1 - z_{j})) \\ \mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < I\}}] &= \frac{\xi_{i}}{\xi_{i} + \sum_{j \neq i} \lambda_{j}(1 - z_{j})} \cdot (1 - \tilde{I}_{i}(\xi_{i} + \sum_{j \neq i} \lambda_{j}(1 - z_{j}))) \\ \mathbb{E}[\mathbf{z}^{N(Y)} \mathbf{1}_{\{Y < X\}}] &= \frac{\xi_{i}}{\xi_{i} + \sum_{j} \lambda_{j}(1 - z_{j})} \cdot (1 - \tilde{X}_{i}(\xi_{i} + \sum_{j} \lambda_{j}(1 - z_{j}))) . \end{split}$$

Here N(T) denotes the number of arrivals during a random period T while $1_{\{A\}}$ denotes the indicator function for event A. Equations (13) and (14) can be explained by the following observations. The number of customers at all queues at the end of a service period is equal to the number present at the end of the previous service period plus the ones that arrived during the present service period. The length of the service period depends on whether a customer was present at the end of the previous service period, which explains why each equation consists of two parts. Also, the length of a service period, and thus the number of arriving customers, depends on whether a service period is interrupted or not. Finally, we note that $\phi_0(1) = 1$, while $\phi_k(1) \leq 1$, for all k = 1, 2, ..., since the *k*th service completion may not occur at all during a visit to Q_i . This explains the differences between Eqs. (13) and (14).

Notice that there is a one-to-one relationship between a visit completion and the end of a final service period. Therefore, we can write

$$\beta^i(\mathbf{z}) = \sum_{k=1}^{\infty} \phi^s_k(\mathbf{z}) \; .$$

We set up an iterative scheme to obtain $\beta^i(\mathbf{z})$ numerically. The scheme is constructed in terms of Discrete Fourier Transforms (DFTs) as these appear more convenient for computational purposes. To this end, we replace z_i , \forall_i , in the expressions above by $\omega_i^{k_i}$, where $\omega_i = exp(-2\pi I/N_i)$, so that all expressions become functions of $\mathbf{k} = (k_1, \ldots, k_M)$. Here *I* is the imaginary unit and N_i refers to the number of discrete points used for Q_i to determine the joint probabilities. These probabilities that will eventually follow are exact for $N_i \to \infty$, \forall_i . However, the strength of the approach is that in general the probabilities are already close to the exact probabilities for small values of N_i . The pseudo-code of the iterative scheme is presented in Algorithm 1. The standard values for the convergence parameters that have been used are $\epsilon = 10^{-6}$ and $\delta = 10^{-9}$. Finally, via the Inverse Fourier Transform, the steady-state probabilities β_n^i are found.

Remark 2. The p.g.f. $\pi^i(\mathbf{z})$, which refers to the queue-length at service completion instants, can now be obtained using the derived relations (see Sect. 4.2-4.3) and the explicit computation of $\beta^i(\mathbf{z})$. However, $\pi^i(\mathbf{z})$ can also be expressed in terms of the introduced auxiliary p.g.f. $\phi_k(\mathbf{z})$ as follows:

$$\pi^{i}(\mathbf{z}) = \frac{\mathbb{P}(s.c. \ at \ Q_{i} \mid s.c.)}{\mathbb{E}[\# \ s.c. \ per \ visit \ to \ Q_{i}]} \cdot \sum_{k=1}^{\infty} \phi_{k}(\mathbf{z}) \ .$$

$\beta^{i_0}(\mathbf{k}) = 1, \ \forall_{i_0}, \ \forall_{\mathbf{k}}$; (start with an empty system)
FOR $i_1 = 1, \ldots, M$
set $i_2 := i_1$;
REPEAT
$\hat{eta}^{i_2}(\mathbf{k})=eta^{i_2}(\mathbf{k}),\ orall_{\mathbf{k}};$
set $j := 0$;
set $\phi_0(\mathbf{k}) = \beta^{i_2-1}(\mathbf{k}) \cdot C_{i_2}(\mathbf{k});$
REPEAT
set $j := j + 1$;
compute $\phi_j(\mathbf{k}), \forall_{\mathbf{k}}$, using Eq. (13);
compute $\phi_j^s(\mathbf{k}), \forall_{\mathbf{k}}$, using Eq. (14);
compute $\beta^{i_2}(\mathbf{k}) = \sum_{l=1}^{j} \phi_l^s(\mathbf{k}), \ \forall_{\mathbf{k}};$
UNTIL $1 - \operatorname{Re}(\beta^{i_2}(0)) < \delta$
set $i_2 := \operatorname{MOD}(i_2, M) + 1;$
$\boxed{\text{UNTIL} \text{Re}(\beta^{i_1}(\mathbf{k})) - \text{Re}(\hat{\beta}^{i_1}(\mathbf{k})) < \epsilon, \ \forall_{\mathbf{k}}}$
END (FOR)

Algorithm 1. Pseudo-code of iterative scheme for determining $\beta^{i}(\mathbf{k}), \forall_{i}$.

Remark 3. In our model, interruptions can occur during both services and idle periods, while in Leung's time-limited model (see [2]) only services can be interrupted. The latter is due to the fact that in Leung's model the server moves to the next queue if there are no customers present anymore. Due to the additional event of idle period interruption in our model, the probability $\psi_i(j) \ge 1$ (one or more customers present at Q_i after j services) of Eq.(9) of [15] which is conditioned on the event that no interruption occurs during the jth service is no longer equal to the unconditional probability. Nevertheless, we strongly believe that for our model the approach of [15] could still be followed to find $\beta^i(\mathbf{z})$. However, the expressions will become quite involved, so that we proposed here an unconditional approach.

4.5 Steady-state queue-length probabilities

The exponential visit times allow us to obtain the steady-state queue-length probabilities. More specifically, we have that a departing server observes the system in steady-state conditioned on the position of the server. Thus, we can write for the steady-state probabilities



Figure 1: The coefficient of correlation as function of Λ for $\mu = 1.00$ and $\xi = 1.00$ (exponential service times).



Figure 2: The coefficient of correlation as function of ξ for $\Lambda = 0.15$ and $\mu = 1.00$ (exponential service times).

 p_n :

$$p_{\mathbf{n}} = \sum_{i=1}^{M} \mathbb{P}(\mathbf{n} | \text{server at } Q_i) \cdot \mathbb{P}(\text{server at } Q_i) = \sum_{i=1}^{M} \beta_{\mathbf{n}}^i \cdot \frac{1/\xi_i}{\sum_j 1/\xi_j}$$

This contrasts with most previous work on polling models for which no steady-state queuelength probabilities could be derived.

5 Approximations

We have performed experiments for a wide range of parameter settings for the polling model. As an example, we present results for a symmetric system with three queues, exponential service times and zero switch-over times. For ease of presentation, we define $\Lambda = \sum_{j} \lambda_{j}$, $\mu = \mu_{i}$, and $\xi = \xi_{i}$, for i = 1, ..., M. Specifically, we plot the coefficient of correlation, $\rho_{1,2|Q_{j}}$, j = 1, 2, 3, for the conditional queue length at Q_{1} and Q_{2} as function of Λ and ξ , where $\rho_{1,2|Q_{j}}$ is defined as follows:

$$\rho_{1,2|Q_j} := \frac{\operatorname{Cov}(N_1, N_2 | \operatorname{server} \operatorname{at} Q_j)}{\sqrt{\operatorname{Var}(N_1 | \operatorname{server} \operatorname{at} Q_j)\operatorname{Var}(N_2 | \operatorname{server} \operatorname{at} Q_j)}}$$
$$= \frac{\mathbb{E}[N_1, N_2 | \operatorname{server} \operatorname{at} Q_j] - \mathbb{E}[N_1 | \operatorname{server} \operatorname{at} Q_j]\mathbb{E}[N_2 | \operatorname{server} \operatorname{at} Q_j]}{\sqrt{\operatorname{Var}(N_1 | \operatorname{server} \operatorname{at} Q_j)\operatorname{Var}(N_2 | \operatorname{server} \operatorname{at} Q_j)}}$$

We will only consider the conditional queue-lengths here. This is because the system state generally depends on the position of the server, so that it is more meaningful to compare conditional probabilities. Moreover, if we would take a snapshot of the system state at a random instant in time, then we do not expect it to be in line with the unconditional time-equilibrium probabilities.

In Fig. 1, we plot $\rho_{1,2|Q_j}$ as function of the arrival rate Λ for the situation $\mu = 1.00$ and $\xi = 1.00$. It is shown that the correlation between the queues is quite small (for all server's positions), although it increases (in absolute sense) slightly in Λ . Figure 2 shows the impact of increasing ξ (i.e., decreasing the mean visit time to a queue) on $\rho_{1,2|Q_j}$ for the situation $\Lambda = 0.15$ and $\xi = 1.00$. The plot shows that the coefficient of correlation decreases rapidly in ξ . This is in accordance with the fact that for $\xi \to \infty$ the queue lengths indeed become independent

yielding a coefficient of correlation equal to zero. We have also generated results for many other parameter settings for the symmetric three-queue system. These results demonstrate that for a wide range of settings the coefficient of correlation is quite small which indicates little dependence between the queue lengths at the different queues.

A natural next step is then to study approximations for the joint queue-length distribution of the polling model based on the assumption of independence of the queues. Such approximations could be of great value since our experiments have shown that the computation time for the joint queue-length probabilities in the polling model may grow quite large. Moreover, the convergence steps in the iterative scheme may become quite small which further contributes to large computation times.

The approximation for the joint queue-length distribution is thus based on the marginal distributions. These marginal distributions can be computed directly via the unreliable server model (see Sect. 3). In this way, the single-queue results can be obtained very fast which is often a necessity for real applications. Specifically, the approximation reads as follows:

$$\mathbb{P}(N_1 = n_1, \dots, N_M = n_M | \text{server at } Q_j) \approx \prod_{i=1}^M \mathbb{P}(N_i = n_i | \text{server at } Q_j) .$$
(15)

To assess the quality of this approximation, we compute the terms on the r.h.s. of the Eq. (15) via the USM. As we have not analyzed these terms yet, this will be done next.

Let us consider the unreliable server model with arrival rate λ , service rate μ , exponentially distributed availability periods with parameter ξ and $\operatorname{Erlang}_{M-1}(\xi)$ distributed repair periods. We let the p.g.f. $\hat{N}_{1j}(z) = \mathbb{E}[z^{N(Q_1)}|$ server at $Q_j]$, $j = 1, \ldots, M$, refer to the number of customers in the queue given that the server is either at the queue (j = 1) or at "stage" j - 1 of the repair period. Notice that (due to exponentially distributed availability periods) $\hat{N}_{11}(z)$ in fact refers to the p.g.f. of the number of customers present at an arbitrary instant of the availability period. Denote further by $\hat{N}_{1D}(z)$ the p.g.f. of the number of customers present at an arbitrary instant of the repair period. These quantities are related to $P_{L_d}(z)$ as follows:

$$P_{L_d}(z) = p_a \hat{N}_{11}(z) + p_{\tau} \hat{N}_{1D}(z)$$

where p_a and p_r are the long-term fractions that the server is available and being repaired, respectively. Observe that $\hat{N}_{11}(z)$ and $\hat{N}_{1D}(z)$ are also related via:

$$\hat{N}_{1D}(z) = \hat{N}_{11}(z) \cdot \hat{D}_A(z) ,$$

where $\hat{D}_A(z)$ is the p.g.f. of the number of arrivals from the start of the repair period until an arbitrary instant of that period, and satisfies, using simple regenerative processes theory (see, e.g., [19]):

$$\hat{D}_A(z) = rac{1-D(z)}{\hat{D}'(1)(1-z)} \; ,$$

where $\hat{D}(z)$ (= $\tilde{D}(\lambda(1-z))$) is the p.g.f. of the number of arrivals during the repair period. Hence, it follows that:

$$\hat{N}_{11}(z) = rac{P_{L_d}(z)}{p_a + p_r \hat{D}_A(z)} \; .$$

We note $\hat{N}_{1j}(z)$, $j \neq 1$, can be decomposed in three independent parts. The first part refers to the number of customers present at the end of an availability period. The second part accounts for the arrivals during the already completed repair stages. Finally, the last part represents the

number of arrivals from the beginning of repair stage j - 1 until a random instant during this stage. In terms of p.g.f.'s, this leads to:

$$\hat{N}_{1j}(z) = \hat{N}_{11}(z) \cdot \prod_{k=1}^{j-2} \hat{D}^k(z) \cdot \hat{D}^j_A(z) , \ j = 2, \dots, M ,$$

where $\hat{D}^k(z)$ refers to the arrivals during the (completed) kth stage of the repair period and is given by

$$\hat{D}^{k}(z) = \tilde{D}^{k}(\lambda(1-z)), \ k = 1, \dots, M-2,$$

and $\hat{D}_{A}^{j}(z)$ (cf. $\hat{D}_{A}(z)$) is given by

$$\hat{D}_A^j(z) = \frac{1 - \hat{D}^j(z)}{\hat{D}'^j(1)(1 - z)}, \ j = 2, \dots, M$$
.

Finally, the probabilities $\mathbb{P}(N_i = n_i | \text{server at } Q_j)$ are obtained from $N_{1j}(z)$ using DFT techniques. Notice that for a comparison with an asymmetric polling system all steps above have to be performed for each queue separately.

The proposed approximation is anticipated to work well in situations where the individual queues behave independently. In our polling model, it seems that due to our imposed visit-time distribution the dependencies between the different queues are small. For instance, the number of arrivals during the absence of the server and the time that a queue is served are known (in distribution) and independent of what occurs at the other queues in the system.

5.1 Performance measure

We have now all the tools at hand to investigate the independencies between the queues in the polling system. Let us emphasize that our objective here is not to perform an exhaustive numerical study for all system parameters and service time distributions. The underlying idea of the approximation is that if the queues in the system would turn out to be "almost" independent, then the results of a much simpler single-queue model can be used as a good approximation for a complex multi-queue polling model. Therefore, our purpose is mainly to gain preliminary insight in the parameter ranges for which the approximation works well. The performance measure that we use to assess the quality of the approximation is as follows. We use the measure of total variation distance [20] for the queue-length distribution conditional on the position of the server, denoted by $\theta_{cond,i}^p$:

$$\theta_{cond,j}^p := \sum_{\mathbf{n}} \left| \mathbb{P}(N_1 = n_1, \dots, N_M = n_M \mid \text{server at } Q_j) - \prod_{i=1}^M \mathbb{P}(N_i = n_i \mid \text{server at } Q_j) \right|$$

5.2 Numerical results

We present here results from experiments for a symmetric three-queue polling model for both exponentially and deterministically distributed service times. For ease of presentation, we define $\theta_{cond}^p = \theta_{cond,j}^p$, for j = 1, ..., M.

The results for the total variation distance in the exponential case are presented in Figs. 3 and 4. First, consider Fig. 3 in which θ_{cond}^p is plotted as function of Λ for various values of ξ . The slopes observed in this figure clearly show that θ_{cond}^p is not insensitive to Λ , but increases



Figure 3: The total variation distance as function of Λ (exponential service times).



Figure 5: The total variation distance as function of Λ (deterministic service times).



Figure 4: The total variation distance as function of ξ (exponential service times).



Figure 6: The total variation distance as function of ξ (deterministic service times).

linearly in the arrival rate. Moreover, it can be seen that θ_{cond}^p decreases in ξ . To better understand the rate of decrease in ξ , we plot in Fig. 4 the impact of ξ on θ_{cond}^p for various values of Λ . It is shown that θ_{cond}^p decreases rapidly in ξ toward zero for all values of Λ .

The results for the deterministic service times are presented in Figs. 5 and 6. Figure 5 shows θ_{cond}^p as function of Λ for various values of ξ . Again as for the exponential case, θ_{cond}^p increases linearly in Λ . The impact of ξ on θ_{cond}^p appears small. This is confirmed by the plot of Fig. 6 which shows the total variation distance as function of ξ for various values of Λ . An important difference with respect to the exponential case is that the θ_{cond}^p goes to some asymptotic value strictly larger than zero. The latter is due to the fact that the load for the deterministic case increases in ξ , so that the queue lengths will not approach independence under the stable regime (i.e., $\rho < 1$).

Let us wrap up the main observations that we have done in our experiments for the threequeue symmetric system: (i) θ_{cond}^p is positively correlated to the arrival rate Λ ; (ii) θ_{cond}^p decreases rapidly toward zero in the visit time parameter ξ for exponential service times, while for deterministic service θ_{cond}^p decreases to an asymptotic value.

We have seen that there exists a wide range of parameter settings for which the approximation works quite well. However, the approximation appears not applicable to heavily loaded systems. For such situations, it might be worthwhile to consider heavy-traffic approximations. This will be part of future work.

6 Conclusions

Polling models with an autonomous server may arise as a performance model in the context of mobile wireless technologies. We have analyzed this polling model in great detail by determining the queue-length distribution at various instants. Our analytical approach appears mainly applicable to systems with a light to moderate load. We have performed several experiments to study the independence between queues, so that we identify system parameter settings for which a simple single-queue model can successfully be applied to approximate performance measures. These experiments show that the quality of the approximation is not very sensitive to the total arrival rate, but mainly depends on the mean visit time. The shorter the visit times, the better will be the approximation for the polling model measures.

In future work, we will study other network structures such as a (multihop) chain model or a multi-path model. We strongly believe that similar techniques as described above may prove useful to analyze such models. Later, we want to combine analytical results for these simple network structures to analyze more complex network structures. For instance, more complex mobility patterns and even models with multiple servers will be considered. Also incorporating communication between mobile nodes is a valuable model extension.

References

- H. Takagi, "Queueing analysis of polling systems: An update," Chapter of Stochastic Analysis of Computer and Communication Systems, North-Holland Publ. Cy., Amsterdam, pp. 267–318, 1990.
- [2] K. K. Leung, "Cyclic-service systems with non-preemptive time-limited service," IEEE Tr. on Communications, vol. 42(8), pp. 2521–2524, 1994.
- [3] I. Frigui and A.-S. Alfa, "Analysis of a time-limited polling system," Computer Communications, vol. 21(6), pp. 558–571, 1998.
- [4] E. de Souza e Silva, H. Gail, and R. Muntz, "Polling systems with server timeouts and their application to token passing networks," *IEEE Tr. on Networking*, vol. 3(5), pp. 560– 575, 1995.
- [5] M. Tangemann and K. Sauer, "Performance analysis of the timed token protocol of FDDI and FDDI-II," IEEE Journal on Selected Areas in Communications, vol. 9(2), pp. 271–278, Feb. 1991.
- [6] M. Eisenberg, "Two queues with changeover times," Operations Research, vol. 19, pp. 386–401, 1971.
- [7] S. Borst, "A polling system with a dormant server," Report BS-R9313, Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, 1993.
- [8] O. Boxma, S. Schlegel, and U. Yechiali, "A note on an M/G/1 queue with a waiting server, timer and vacations," *American Mathematical Society Translations*, vol. 2(207), pp. 25–35, 2002.
- [9] —, "Two-queue polling models with a patient server," Annals of Operations Research, vol. 112, pp. 101–121, 2002.

- [10] J. Xie, M. J. Fischer, and C. M. Harris, "Workload and waiting time in a fixed-time loop system," Computers Operations Research, vol. 24(8), pp. 789–803, 1997.
- [11] D. Gaver, "A waiting line with interrupted service, including priorities," *Journal of Royal Statistical Society*, vol. 24(1), pp. 73–90, 1962.
- [12] G. Fayolle and R. Iasnogorodski, "Two coupled processors: The reduction to a Riemann-Hilbert problem," Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, vol. 47, pp. 325–351, 1979.
- [13] E. Coffman, G. Fayolle, and I. Mitrani, "Two queues with alternating service periods," in Performance '87: Proceedings of the 12th IFIP WG 7.3 International Symposium on Computer Performance Modelling, Measurement and Evaluation, 1988, pp. 227–239.
- [14] M. Eisenberg, "Queues with periodic service and changeover times," Operations Research, vol. 20(2), pp. 440–451, 1972.
- [15] K. K. Leung, "Cyclic-service systems with probabilistically-limited service," IEEE Journal on Selected Areas in Communications, vol. 9(2), pp. 185–193, Feb. 1991.
- [16] J. van Ommeren, "The discrete-time single-server queueing model," Queueing Systems, vol. 8, pp. 279–294, 1991.
- [17] F. Baccelli and P. Brémaud, Elements of Queueing Theory. Springer-Verlag, 1994.
- [18] S. M. Ross, Stochastic Processes. John Wiley & Sons, 1996.
- [19] S. Fuhrmann and R. B. Cooper, "Stochastic decompositions in the M/G/1 queue with generalized vacations," Operations Research, vol. 33(5), pp. 1117–1129, 1985.
- [20] W. Feller, An Introduction to Probability Theory and Its Applications, Volume II. John Wiley & Sons, 1966.

Regularly Varying Tail of the Waiting Time Distribution in an M/G/1 Retrial Queue

Jerim Kim and Bara Kim

Department of Mathematics and Telecommunication Mathematics Research Center Korea University, 1, Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea e-mail: b1155@hanmail.net, bara@korea.ac.kr

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

Abstract

We consider an M/G/1 retrial queue, where the service time distribution has a regularly varying tail with index $-\beta$, $1 < \beta < 2$. The waiting time distribution is shown to have a regularly varying tail with index $1 - \beta$, and the pre-factor is determined explicitly. The result is obtained by comparing the waiting time in the M/G/1 retrial queue with the waiting time in the ordinary M/G/1 queue with random order service policy.

Keywords: M/G/1 retrial queue, regular variation, waiting time distribution, random order service.

1 Introduction

Retrial phenomena arise in many practical situations such as in call center systems and many other telecommunication systems. Retrial queues, which deals with the stochastic models for the retrial phenomena, have been investigated for several decades. Detailed overviews for retrial queues can be found in Falin and Templeton [7], Artalejo [1] and Choi and Chang [6].

In this paper, we consider an M/G/1 retrial queueing system, where customers arrive according to a Poisson process with intensity λ , service times *B* for customers are independent and identically distributed with distribution function F_B , and there is a single server. If the server is idle at the time of a customer arrival, the arriving customer begins to be served immediately and leaves the system after service completion. Otherwise, i.e., if the server is busy, the arriving customer joins a retrial group, called an orbit. While in orbit, each customer spends an exponential time with mean ν^{-1} before visiting the server again. If an incoming repeated customer from the orbit finds the server idle, it is served and leaves the system after service completion. Otherwise, i.e., if the repeated customer finds the server busy, the customer comes back to the orbit immediately, and tries her or his luck after an exponential time with mean ν^{-1} again. The traffic load ρ is defined as $\rho = \lambda \mathbb{E}B$. It is assumed that $\rho < 1$ for the stability of the system.

The interest of this paper is the heavy-tailed asymptotics for the waiting time distribution in the M/G/1 retrial queue. There are fluent references for the heavy-tailed asymptotics in usual queues. See, for examples, [2, 3, 9, 12] and references therein. However, for the heavytailed asymptotics in retrial queues it seems that Shang, Liu and Li [11] is the only known result in open literature. Shang, Liu and Li [11] showed that the stationary distribution of the queue length in the M/G/1 retrial queue is subexponential if the stationary distribution of the queue length in the corresponding ordinary M/G/1 queue is subexponential. As a corollary of this property, they proved that the the stationary distribution of the queue length in the M/G/1 retrial queue has a regularly varying tail if the service time distribution has a regularly varying tail.

The main contribution of this paper is to show that if the service time distribution has a regularly varying tail of index $-\beta$, $1 < \beta < 2$, in the M/G/1 retrial queue, then the waiting time distribution has a regularly varying tail of index $1 - \beta$. More precisely, we prove that if the distribution function F_B of service times satisfies

$$1 - F_B(x) \sim x^{-\beta} L(x) \text{ as } x \to \infty$$

with a slowly varying function L, then the distribution function F_W for the waiting time of an arbitrary customer satisfies

$$1 - F_W(x) \sim c x^{1-\beta} L(x) \text{ as } x \to \infty$$
 (1)

with a constant c > 0 that is given explicitly. Here and subsequently $f(x) \sim g(x)$ denotes $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$.

Boxma et al. [5] obtained the same result as (1) for the waiting time distribution in the ordinary M/G/1 queue with random order service (ROS) policy. The main result (1) of this paper is obtained by comparing the waiting time in the M/G/1 retrial queue with the waiting time in the ordinary M/G/1 queue with ROS policy.

The remainder of this paper is organized as follows: In Section 2, we show that for the M/G/1 retrial queue, if the service time distribution has a regularly varying tail of index $-\beta$, $\beta > 1$, then tails of several first passage time distributions are bounded by a function that is of regular variation with index $-\beta$. In Section 3, when the service time distribution has a regularly varying tail of index $-\beta$, $1 < \beta < 2$, the main result (1) is derived with the explicit expression for *c* by comparing the waiting time in the M/G/1 retrial queue with the waiting time in the ordinary M/G/1 queue with ROS policy.

2 First passage time distributions

Let

N(t) = the number of customers in the orbit at time t; $C(t) = \begin{cases} 1, & \text{if the server is busy at } t, \\ 0, & \text{if the server is idle at } t; \end{cases}$ $X(t) = \begin{cases} \text{the elapsed service time of the customer who is in service at } t, & \text{if } C(t) = 1, \\ 0, & \text{if } C(t) = 0. \end{cases}$

Then $\{(N(t), C(t), X(t)) : t \ge 0\}$ is a Markov process. Let

$$\tau_n = \inf\{t > 0 : N(t) = n, C(t) = 1, X(t) = 0\}, \qquad n = 0, 1, 2, \dots;$$
(2)

$$\sigma_n = \inf\{t > 0 : N(t) = n, C(t) = 0\}, \qquad n = 0, 1, 2, \dots;$$
(3)

$$G_n(x) = \mathbb{P}(\sigma_n \le x \mid N(0) = n, C(0) = 1, X(0) = 0), \qquad n = 0, 1, 2, \dots;$$
(4)

$$H_n(x) = \mathbb{P}(\tau_{n-1} \le x \mid N(0) = n, C(0) = 1, X(0) = 0), \qquad n = 1, 2, 3, \dots.$$
(5)

For a distribution function F, the complementary distribution function is denoted by \overline{F} , i.e, $\overline{F}(x) = 1 - F(x), x \in \mathbb{R}$. Clearly

$$\overline{G}_n(x) \le \overline{H}_n(x), \qquad n = 1, 2, 3, \dots$$
(3)

Further it can be shown, by stochastic comparison, that

$$\overline{G}_n(x) \ge \overline{G}_m(x), \quad x \in \mathbb{R}, \quad \text{if } 0 \le n \le m;$$

$$\overline{H}_n(x) \ge \overline{H}_m(x), \quad x \in \mathbb{R}, \quad \text{if } 0 \le n \le m.$$

In this section, we assume that the service time distribution has a regularly varying tail with index $-\beta$, $\beta > 1$, i.e.,

$$\overline{F}_B(x) \sim x^{-\beta} L(x)$$
 as $x \to \infty$

with a slowly varying function *L*. The following proposition asserts that, for all *n*, $\overline{G}_n(x)$ and $\overline{H}_n(x)$ are bounded by a function that is of regular variation with index $-\beta$. This result will be used in Section 3 to prove that the tail of the waiting time distribution in the retrial queue has a regular variation with index $1 - \beta$.

Proposition 1. We have

$$\overline{G}_n(x) \lesssim x^{-\beta}L(x), \quad n=0,1,2,\ldots,$$

and

 $\overline{H}_n(x) \lesssim x^{-\beta}L(x), \quad n=1,2,3,\ldots,$

where $f(x) \lesssim g(x)$ denotes $\lim_{x\to\infty} \frac{f(x)}{g(x)} < \infty$.

The proof is deferred to the end of this section. For the proof, we need a series of lemmas. **Lemma 1.** For $n \ge 1$,

 $\overline{G}_n(x) \lesssim \overline{F}_B(x)$ if and only if $\overline{H}_n(x) \lesssim \overline{F}_B(x)$.

Proof. By (3), $\overline{H}_n(x) \lesssim \overline{F}_B(x)$ implies $\overline{G}_n(x) \lesssim \overline{F}_B(x)$. Now we show the converse:

 $\overline{G}_n(x) \lesssim \overline{F}_B(x)$ implies $\overline{H}_n(x) \lesssim \overline{F}_B(x)$.

Suppose that $\overline{G}_n(x) \lesssim \overline{F}_B(x)$. Let

$$J_n(x) = \mathbb{P}(\tau_{n-1} \le x \mid N(0) = n, C(0) = 0).$$

Then

$$H_n(x) = G_n * J_n(x). \tag{4}$$

We observe that

$$J_n(x) = \frac{n\nu}{n\nu+\lambda} E_{n\nu+\lambda}(x) + \frac{\lambda}{n\nu+\lambda} E_{n\nu+\lambda} * H_n(x),$$
(5)

where E_{α} denotes the exponential distribution function with mean α^{-1} . Substituting (5) into (4) leads to

$$H_n(x) = \frac{n\nu}{n\nu + \lambda} G_n * E_{n\nu + \lambda}(x) + \frac{\lambda}{n\nu + \lambda} G_n * E_{n\nu + \lambda} * H_n(x)$$

This implies

$$H_n(x) = \sum_{k=1}^{\infty} \frac{n\nu}{n\nu + \lambda} \left(\frac{\lambda}{n\nu + \lambda}\right)^{k-1} (G_n * E_{n\nu + \lambda})^{*k}(x),$$

where the superscript *k on the right hand side denotes the *k*-fold convolution. Since $\overline{G}_n(x) \leq \overline{F}_B(x)$, we have $\overline{G}_n * \overline{E}_{n\nu+\lambda}(x) \leq \overline{F}_B(x)$. By Proposition 2.9 in [12], we obtain $\overline{H}_n(x) \leq \overline{F}_B(x)$.

Now we define

Note that q is the probability that at least one exogenous arrival occurs during a service time.

Lemma 2. (1) For n = 0, 1, 2, ...,

$$G_n(x) = (1 - q)G_{n,0}(x) + q(K_n * G_n)(x),$$
(6)

where

$$\begin{array}{lll} G_{n,0}(x) & = & \mathbb{P}(\sigma_n \leq x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) = 0), \\ K_n(x) & = & \mathbb{P}(\tau_n \leq x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) \geq 1). \end{array}$$

(2) For $n = 0, 1, 2, \ldots$,

$$\overline{G}_{n,0}(x) \leq \frac{e^{-\lambda x}}{1-q}\overline{F}_B(x),$$

$$\overline{K}_n(x) \leq \frac{1}{q}\overline{H}_{n+1}(x).$$

Proof. (1) We decompose $G_n(x)$ as

$$G_{n}(x) = \mathbb{P}(\sigma_{n} \leq x \mid N(0) = n, C(0) = 1, X(0) = 0)$$

$$= \mathbb{P}(\sigma_{n} \leq x, A(\theta) = 0 \mid N(0) = n, C(0) = 1, X(0) = 0)$$

$$+ \mathbb{P}(\sigma_{n} \leq x, A(\theta) \geq 1 \mid N(0) = n, C(0) = 1, X(0) = 0)$$

$$= (1 - q)\mathbb{P}(\sigma_{n} \leq x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) = 0)$$

$$+ q \mathbb{P}(\sigma_{n} \leq x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) \geq 1)$$

$$= (1 - q)G_{n,0}(x) + q\mathbb{P}(\sigma_{n} \leq x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) \geq 1). \quad (7)$$

Given $\{N(0) = n, C(0) = 1, X(0) = 0, A(\theta) \ge 1\}$, we have $\tau_n < \sigma_n$, i.e,

$$\sigma_n = \tau_n + (\sigma_n - \tau_n) \text{ with } \sigma_n - \tau_n > 0.$$

Further, given $\{N(0) = n, C(0) = 1, X(0) = 0, A(\theta) \ge 1\}$, we have the following:

- τ_n has the distribution function K_n ;
- τ_n and $\sigma_n \tau_n$ are independent;

• $\sigma_n - \tau_n$ has the distribution function G_n .

Therefore

$$\mathbb{P}(\sigma_n \le x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) \ge 1) = K_n * G_n(x).$$
(8)

Substituting (8) into (7) leads to (6).

(2) We write

$$\begin{aligned} \overline{G}_{n,0}(x) &= \mathbb{P}(\sigma_n > x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) = 0) \\ &= \frac{1}{1-q} \mathbb{P}(\sigma_n > x, A(\theta) = 0 \mid N(0) = n, C(0) = 1, X(0) = 0) \\ &= \frac{1}{1-q} \mathbb{P}(\theta > x, A(\theta) = 0 \mid N(0) = n, C(0) = 1, X(0) = 0). \end{aligned}$$

Since $\{\theta > x, A(\theta) = 0\} \subset \{\theta > x, A(x) = 0\}$, we have

$$\overline{G}_{n,0}(x) \leq \frac{1}{1-q} \mathbb{P}(\theta > x, A(x) = 0 \mid N(0) = n, C(0) = 1, X(0) = 0)$$
$$= \frac{1}{1-q} \overline{F}_B(x) e^{-\lambda x}.$$

We write

$$\overline{K}_{n}(x) = \mathbb{P}(\tau_{n} > x \mid N(0) = n, C(0) = 1, X(0) = 0, A(\theta) \ge 1)$$

= $\frac{1}{q} \mathbb{P}(\tau_{n} > x, A(\theta) \ge 1 \mid N(0) = n, C(0) = 1, X(0) = 0).$ (9)

Since

$$\begin{split} \mathbb{P}(\tau_n > x, A(\theta) = k \mid N(0) = n, C(0) = 1, X(0) = 0) \\ \leq \quad \mathbb{P}(\tau_n > x, A(\theta) = k \mid N(0) = n + 1, C(0) = 1, X(0) = 0), \quad k = 1, 2, 3, \dots, \end{split}$$

we have

$$\mathbb{P}(\tau_n > x, A(\theta) \ge 1 \mid N(0) = n, C(0) = 1, X(0) = 0) \\
= \sum_{k=1}^{\infty} \mathbb{P}(\tau_n > x, A(\theta) = k \mid N(0) = n, C(0) = 1, X(0) = 0) \\
\le \sum_{k=1}^{\infty} \mathbb{P}(\tau_n > x, A(\theta) = k \mid N(0) = n + 1, C(0) = 1, X(0) = 0) \\
\le \mathbb{P}(\tau_n > x \mid N(0) = n + 1, C(0) = 1, X(0) = 0) \\
= \overline{H}_{n+1}(x).$$
(10)

By (9) and (10), we obtain

$$\overline{K}_n(x) \le \frac{1}{q} \overline{H}_{n+1}(x).$$

For $n \ge 1$, we consider an ordinary M/G/1 queue where arrival rate is λ and service times have a distribution function $F_{B(n)}$:

$$F_{B^{(n)}}(x) = \sum_{k=1}^{\infty} \frac{n\nu}{n\nu + \lambda} \left(\frac{\lambda}{n\nu + \lambda}\right)^{k-1} (F_B * E_{n\nu + \lambda})^{*k}(x).$$
(11)

We remark that $F_{B^{(n)}}$ is the distribution function of $B^{(n)}$ defined as

$$B^{(n)} = \sum_{k=1}^{\mathcal{I}} (B_k + \mathcal{E}_k),$$

where $B_k, \mathcal{E}_k, k = 1, 2, 3, ...$, and \mathcal{I} are independent random variables whose distributions are given by

$$\mathbb{P}(B_k \le x) = F_B(x), \qquad x \in \mathbb{R},$$
(12)

$$\mathbb{P}(\mathcal{E}_k \le x) = E_{n\nu+\lambda}(x), \qquad x \in \mathbb{R},$$
(13)

$$\mathcal{P}(\mathcal{I}=k) = \frac{n\nu}{n\nu+\lambda} \left(\frac{\lambda}{n\nu+\lambda}\right)^{k-1}, \quad k=1,2,3,\dots$$
 (14)

The mean of $B^{(n)}$ is given by

P

$$\mathbb{E}B^{(n)} = (\mathbb{E}B_1 + \mathbb{E}\mathcal{E}_1)\mathbb{E}\mathcal{I} = \left(\mathbb{E}B + \frac{1}{n\nu + \lambda}\right)\left(1 + \frac{\lambda}{n\nu}\right).$$

Let

$$\rho^{(n)} \equiv \lambda \mathbb{E}B^{(n)} = \left(\rho + \frac{\lambda}{n\nu + \lambda}\right) \left(1 + \frac{\lambda}{n\nu}\right)$$
(15)

denote the offered load in the M/G/1 queue with the distribution function (11) for service times.

Lemma 3. Suppose that $\rho^{(n)} < 1$. The distribution function $G^{(n)}$ of a busy period in the M/G/1 queue satisfies

$$\overline{G^{(n)}}(x) \sim (1-
ho^{(n)})^{-eta-1}(1+rac{\lambda}{n
u})x^{-eta}L(x) \quad \text{as } x \to \infty.$$

Proof. By Proposition 2.9 in [12],

$$\overline{F}_{B^{(n)}}(x) \sim (1 + \frac{\lambda}{n\nu}) x^{-\beta} L(x) \quad \text{as } x \to \infty.$$
(16)

Combining (16) with the result in [8] completes the proof.

We now prove Proposition 1.

Proof of Proposition 1. Choose *n* such that $\rho^{(n)} < 1$. Lemma 3 yields

$$\overline{G^{(n)}}(x) \lesssim x^{-\beta} L(x). \tag{17}$$

By stochastic comparison of the M/G/1 retrial queue and the ordinary M/G/1 queue where service times have the distribution function $F_{B(n)}$, it can be easily shown that

$$\overline{G}_k(x) \le \overline{G^{(n)}}(x), \quad k \ge n.$$
(18)

We have, by (17) and (18),

$$\overline{G}_k(x) \lesssim x^{-\beta} L(x), \quad k \ge n,$$
(19)

and by (6),

$$G_k(x) = G_{k,0} * \sum_{i=0}^{\infty} (1-q)q^i K_k^{*i}(x), \qquad k = 0, 1, 2, \dots$$

By Lemma 2 (2) and Proposition 2.9 in [12],

$$\overline{G}_k(x) \lesssim x^{-\beta}L(x)$$
 if $\overline{H}_{k+1}(x) \lesssim x^{-\beta}L(x), \quad k = 0, 1, 2, \dots$ (20)

The proof is complete by Lemma 1, (19) and (20).

3 Regularly varying tail of the waiting time distribution

In this section, we prove that if the service time distribution has a tail of a regular variation with index $-\beta$, $1 < \beta < 2$, then the waiting time distribution in the M/G/1 retrial queue has a tail of regular variation with index $1 - \beta$.

The result is proved by comparing the waiting time distribution in the M/G/1 retrial queue with the waiting time distribution in the ordinary M/G/1 queue with ROS policy.

We consider the corresponding ordinary M/G/1 queue with ROS, where the arrival rate is λ and service times have distribution function F_B . In ROS policy, the server randomly takes one of the waiting customers into service at the completion of a service. Let W_{ROS} denote a generic random variable for the waiting time of an arbitrary customer.

First we provide a result of Boxma et al. [5] on the regularly varying tail of the waiting time distribution in the M/G/1 queue with ROS. We assume, as $x \to \infty$,

$$\overline{F}_B(x) \sim x^{-\beta} L(x), \quad 1 < \beta < 2.$$
(21)

Lemma 4. [5] If (21) holds, then

$$\overline{F}_{W_{ROS}}(x) \sim c x^{1-\beta}L(x) \text{ as } x \to \infty,$$

where

$$c = \frac{\rho}{1-\rho}h(\rho,\beta)\frac{1}{\beta-1}\frac{1}{\mathbb{E}B},$$
(22)

with

$$\begin{split} h(\rho,\beta) &= \int_0^1 f(u,\rho,\beta) du, \\ f(u,\rho,\beta) &= \frac{\rho}{1-\rho} (\frac{\rho u}{1-\rho})^{\beta-1} (1-u)^{\frac{1}{1-\rho}} + (1+\frac{\rho u}{1-\rho})^{\beta} (1-u)^{\frac{1}{1-\rho}-1}. \end{split}$$

We now present the main result of this paper:

Theorem 1. Let W be the waiting time of an arbitrary customer in the M/G/1 retrial queue. If (21) holds, then the distribution function F_W of W satisfies

$$\overline{F}_W(x) \sim \overline{F}_{W_{ROS}}(x) \quad as \ x \to \infty,$$

i.e.,

$$\overline{F}_W(x) \sim c x^{1-\beta} L(x) \quad as \ x \to \infty,$$

where c is given by (22).

Since $W \ge W_{ROS}$ in distribution, we have

$$\overline{F}_W(x) \ge \overline{F}_{W_{ROS}}(x) \quad \text{for all } x \in \mathbb{R}.$$

This and Lemma 4 yield

$$\liminf_{x\to\infty}\frac{\overline{F}_W(x)}{x^{1-\beta}L(x)}\geq c.$$

Thus the theorem is proved if we show

$$\limsup_{x \to \infty} \frac{\overline{F}_W(x)}{x^{1-\beta}L(x)} \le c.$$
(23)

The remainder of this section is devoted to the proof of (23).

We start with introducing notation. Consider the M/G/1 retrial queue. When $N(0) \ge 1$, choose an arbitrary customer in the orbit and call her or him a tagged customer. Let ϕ be the epoch of the beginning of service for the tagged customer. Let

$$\Phi_k(t) = \begin{cases} \mathbb{P}(\phi \le t \mid N(0) = k, C(0) = 1, X(0) = 0), & k = 1, 2, \dots, \\ U(t), & k = 0, \end{cases}$$
(24)

where

$$U(t) = \begin{cases} 1, & t \ge 0, \\ 0, & t < 0. \end{cases}$$

Next, for each n with $\rho^{(n)} < 1$, where $\rho^{(n)}$ is defined as in (15), we consider an ordinary M/G/1 queue with ROS where arrival rate is λ and service times have the distribution function $F_{B^{(n)}}$ defined as in (11). Let $N^{(n)}(t)$ be the number of customers in the queue at t, excluding the one in service if any. When there is a customer in service at t, let $X^{(n)}(t)$ be the elapsed service time of the customer in service. If the service is idle at t, $X^{(n)}(t)$ is set to be zero. When $N^{(n)}(0) \geq 1$, choose an arbitrary customer who is waiting in the queue and call her or him a tagged customer. Let $\phi^{(n)}$ be the epoch of the beginning of service for the tagged customer. Let

$$\Phi_k^{(n)}(t) = \begin{cases} \mathbb{P}(\phi^{(n)} \le t \mid N^{(n)}(0) = k, X^{(n)}(0) = 0), & k > n, \\ U(t), & k \le n. \end{cases}$$

Lemma 5. (a) For $k \ge 0$,

$$\overline{\Phi}_k(t) \le \overline{\Phi}_k^{(n)} * H_1^{*n}(t).$$
(25)

(b) For $k \ge 0$ and $n \ge 0$,

$$\overline{\Phi}_{k+n}(t) \leq \overline{H_1^{*n} * \left(\frac{n}{k+n}U + \frac{k}{k+n}\Phi_k\right)}(t),$$
(26)

$$\overline{\Phi}_{k+n}(t) \leq \frac{H_1^{*2n} * \left(\frac{n}{k+n}U + \frac{k}{k+n}\Phi_k^{(n)}\right)(t)}{(27)}$$

$$\overline{\Phi}_{k+n}(t) \leq H_1^{*2n} * \Phi_k^{(n)}(t).$$
(28)

Proof. (a) Letting

$$\Phi_{k,n}(t) = \begin{cases} \mathbb{P}(\min\{\phi, \tau_n\} \le t \mid N(0) = k, C(0) = 1, X(0) = 0), & k > n, \\ U(t), & k \le n, \end{cases}$$

where τ_n is given by (2), we have

$$\overline{\Phi}_{k,n}(t) \leq \overline{\Phi_k^{(n)}}(t).$$
(29)

Clearly,

$$\overline{\Phi}_k(t) \leq \overline{\Phi_{k,n} * H_n * H_{n-1} * \cdots * H_1}(t).$$
(30)



Figure 1: Embedded points for the Markov process $\{M_k : k = 1, 2, 3, ...\}$ in the M/G/1 retrial queue





Figure 2: A service time for the M/G/1 queue

Substituting (29) and $\overline{H}_m(t) \leq \overline{H}_1(t)$, $m = 1, 2, \dots, n$, into (30) yields (25).

(b) We prove (26) by induction on n. If n = 0, then (26) is trivial. Suppose that (26) holds for $n = m \ge 0$. Then

$$\overline{\Phi}_{k+m+1}(t) \leq H_{k+m+1} * \left(\frac{1}{k+m+1}U + \frac{k+m}{k+m+1}\Phi_{k+m}\right)(t) \\
\leq \overline{H_{k+m+1}} * \left(\frac{1}{k+m+1}U + \frac{k+m}{k+m+1}H_1^{*m} * \left(\frac{m}{k+m}U + \frac{k}{k+m}\Phi_k\right)(t) \\
\leq \overline{H_1^{*(m+1)}} * \left(\frac{m+1}{k+m+1}U + \frac{k}{k+m+1}\Phi_k\right)(t).$$

Thus (26) holds for n = m + 1, which completes the proof of (26). Substitution of (25) into (26) yields (27). The assertion (28) is immediate from (27).

Now we consider Markov chains embedded in the M/G/1 retrial queue and in the ordinary M/G/1 queue with the distribution function $F_{B^{(n)}}$ in (11) for service times.

First, we describe the embedded Markov chain in the M/G/1 retrial queue. For k = 1, 2, 3, ..., let M_k be the number of customers in the orbit immediately after the beginning of the *k*th service for the M/G/1 retrial queue. We observe that $\{M_k : k = 1, 2, 3, ...\}$ is a Markov chain. For an illustration, embedded points are marked with dots in Figure 1.

Next we describe the embedded Markov chain in the ordinary M/G/1 queue with the distribution $F_{B^{(n)}}$ in (11) for service times. Recall that generic service time $B^{(n)}$ is written as

$$B^{(n)} = \sum_{i=1}^{\mathcal{I}} (B_i + \mathcal{E}_i),$$

where $B_i, \mathcal{E}_i, i = 1, 2, 3, ..., \text{ and } \mathcal{I}$ are independent with distributions given by (12)-(14). We call each $B_i + \mathcal{E}_i$ a subservice. Thus a service in the M/G/1 queue consists of a geometric number of subservices. Further a subservice consists of two periods, namely *B*-period and \mathcal{E} -period. The lengths of *B*-period and \mathcal{E} -period have distribution functions F_B and $E_{n\nu+\lambda}$, respectively. Figure 2 illustrates the structure of a service.

For k = 1, 2, 3..., let $M_k^{(n)}$ be the number of customers waiting in the queue, excluding the one that starts a subservice, immediately after the beginning of the *k*th subservice for the M/G/1 queue with the distribution function $F_{B^{(n)}}$ for service times. We observe that $\{M_k^{(n)}:$



Figure 3: Embedded points for the Markov process $\{M_k^{(n)} : k = 1, 2, 3, ...\}$ in the M/G/1 queue with the distribution function $F_{B^{(n)}}$ for service times.

k = 1, 2, 3, ... is a Markov chain. For an illustration, embedded points are marked with dots in Figure 3.

The following lemma provides a relation on the stationary distributions of $\{M_k : k = 1, 2, 3, ...\}$ and $\{M_k^{(n)} : k = 1, 2, 3, ...\}$.

Lemma 6. Let M and $M^{(n)}$ denote random variables having stationary distributions of $\{M_k : k = 1, 2, 3, ...\}$ and $\{M_k^{(n)} : k = 1, 2, 3, ...\}$, respectively. Then

$$(M-n)^+ \leq M^{(n)}$$
 in distribution,

where $(a)^+ = \max\{a, 0\}$.

Bara Kim

Proof. Suppose that $M_1 = 0$ and $M_1^{(n)} = 0$. Then induction on k shows that, for k = 1, 2, 3, ...,

$$(M_k - n)^+ \leq M_k^{(n)}$$
 in distribution.

Letting $k \to \infty$ completes the proof.

Now we prove (23) through 3 steps.

Step 1. Let Ψ be a distribution function defined as

$$\Psi(x) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M=k) a_{l}(y) \Big(\frac{\nu}{(k+l+1)\nu+\lambda} U + \frac{(k+l)\nu}{(k+l+1)\nu+\lambda} \Phi_{k+l} + \frac{\lambda}{(k+l+1)\nu+\lambda} \Phi_{k+l+1} \Big) (x-y) dy$$
(31)

where

$$a_l(y) = \frac{1}{\mathbb{E}B} \int_y^\infty e^{-\lambda t} \frac{(\lambda t)^l}{l!} dF_B(t)$$
(32)

and Φ_k is given by (24). Then

$$\overline{F}_W(x) \le \rho \overline{\Psi * E_{\nu+\lambda}}(x), \ x > 0.$$
(33)

Proof. We choose an arbitrary customer who arrives at the retrial queue and call her or him a tagged customer. Let

$$I = \begin{cases} 1, & \text{if the tagged customer arrives while the server is busy,} \\ 0, & \text{otherwise.} \end{cases}$$



Figure 4: Arrival and beginning of the service for a tagged customer in the M/G/1 retrial queue

By the 'Poisson arrivals see time averages' (PASTA) property,

$$\mathbb{P}(I=1) = \rho. \tag{34}$$

When I = 1, let us define the following related epochs; see Figure 4:

- t_* = the arrival epoch of the tagged customer;
- t_1 = the beginning epoch of the service during which the tagged customer arrives;
- t_2 = the ending epoch of the service during which the tagged customer arrives;
- t_3 = the beginning epoch of the next service after t_2 ;
- t_4 = the beginning epoch of the service for the tagged customer.

When I = 1, let A denote the number of exogenous arrivals during (t_1, t_2) excluding the tagged customer. Given I = 1, $N(t_1)$ and $(A, t_2 - t_*)$ are independent. Further $N(t_1)$ has the same distribution as M. Therefore

$$\mathbb{P}(N(t_1) = k, \ \mathcal{A} = l, \ t_2 - t_* \le y \mid I = 1) = \mathbb{P}(M = k)\mathbb{P}(\mathcal{A} = l, t_2 - t_* \le y \mid I = 1).$$
(35)

Given I = 1, the joint distribution of A and $t_2 - t_*$ is given by

$$\frac{d}{dy}\mathbb{P}(\mathcal{A}=l, t_2-t_* \le y \mid I=1) = a_l(y), \quad l=0,1,2,\dots, y \ge 0,$$
(36)

where $a_l(y)$ is defined as (32). By (34), (35) and (36), we have

$$\frac{d}{dy}\mathbb{P}(I=1, N(t_1)=k, \ \mathcal{A}=l, \ t_2-t_* \le y) = \rho\mathbb{P}(M=k)a_l(y), \ l=0,1,2,\ldots, \ y \ge 0.$$
(37)

If I = 1, $N(t_1) = k$, A = l and $t_2 - t_* \le y$, then $N(t_2) = k + l + 1$; the k + l + 1 customers in the orbit at t_2 consist of the tagged customer and the other k + l customers. Hence, given $\{I = 1, N(t_1) = k, A = l, t_2 - t_* \le y\}$, $t_3 - t_2$ and $t_4 - t_3$ have distribution functions

$$E_{(k+l+1)\nu+\lambda} \text{ and } \frac{\nu}{(k+l+1)\nu+\lambda}U + \frac{(k+l)\nu}{(k+l+1)\nu+\lambda}\Phi_{k+l} + \frac{\lambda}{(k+l+1)\nu+\lambda}\Phi_{k+l+1}$$

Further, given $\{I = 1, N(t_1) = k, A = l, t_2 - t_* = y\}$, $t_3 - t_2$ and $t_4 - t_3$ are independent. Therefore,

$$\mathbb{P}(t_{4} - t_{2} > x \mid I = 1, N(t_{1}) = k, \mathcal{A} = l, t_{2} - t_{*} = y) \\
= \frac{E_{(k+l+1)\nu+\lambda} * \left(\frac{\nu}{(k+l+1)\nu+\lambda}U + \frac{(k+l)\nu}{(k+l+1)\nu+\lambda}\Phi_{k+l} + \frac{\lambda}{(k+l+1)\nu+\lambda}\Phi_{k+l+1}\right)}{E_{\nu+\lambda} * \left(\frac{\nu}{(k+l+1)\nu+\lambda}U + \frac{(k+l)\nu}{(k+l+1)\nu+\lambda}\Phi_{k+l} + \frac{\lambda}{(k+l+1)\nu+\lambda}\Phi_{k+l+1}\right)}(x).$$
(38)

By (37) and (38), the complementary distribution function \overline{F}_W of a waiting time in the retrial queue satisfies the following: For x > 0,

$$\overline{F}_{W}(x) = \mathbb{P}(I = 1, t_{4} - t_{*} > x) \\
= \rho \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M = k)a_{l}(y)\mathbb{P}(t_{4} - t_{*} > x \mid I = 1, N(t_{1}) = k, \mathcal{A} = l, t_{2} - t_{*} = y)dy \\
= \rho \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M = k)a_{l}(y)\mathbb{P}(t_{4} - t_{2} > x - y \mid I = 1, N(t_{1}) = k, \mathcal{A} = l, t_{2} - t_{*} = y)dy \\
\leq \rho \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M = k)a_{l}(y) \\
\times \overline{E_{\nu+\lambda} * (\frac{\nu}{(k+l+1)\nu+\lambda}U + \frac{(k+l)\nu}{(k+l+1)\nu+\lambda}\Phi_{k+l} + \frac{\lambda}{(k+l+1)\nu+\lambda}\Phi_{k+l+1})}(x - y)dy, \quad (39)$$

which is written as (33) with Ψ in (31).

Step 2. Let $W^{(n)}$ be the waiting time of an arbitrary customer in the ordinary M/G/1 queue with ROS and the service time distribution function $F_{B^{(n)}}$. Then

$$\overline{F}_{W^{(n)}}(x) \geq \rho \overline{\Psi^{(n)}}(x), \tag{40}$$

where $\Psi^{(n)}$ is a distribution function defined as

$$\Psi^{(n)}(x) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M^{(n)} = k) a_{l}(y) \Big(\frac{\lambda}{n\nu+\lambda} \Phi^{(n)}_{k+l+1} + \frac{n\nu}{n\nu+\lambda} \frac{1}{k+l+1} U + \frac{n\nu}{n\nu+\lambda} \frac{k+l}{k+l+1} \Phi^{(n)}_{k+l} \Big) (x-y) dy$$
(41)

with $\Phi_k^{(n)}$ and $a_l(y)$ in (29) and (32), respectively.

Proof. We consider the ordinary M/G/1 queue with ROS and distribution function $F_{B^{(n)}}$ for service times. We choose an arbitrary customer who arrives at the queue with ROS and call her or him a tagged customer. Recall the structure of a service in Figure 2. Let

$$I^{(n)} = \begin{cases} 1, & \text{if the tagged customer arrives in a } B\text{-period}, \\ 2, & \text{if the tagged customer arrives in a } \mathcal{E}\text{-period}, \\ 0, & \text{otherwise.} \end{cases}$$

By the PASTA property, we know that

$$\mathbb{P}(I^{(n)} = 1) = \rho; \quad \mathbb{P}(I^{(n)} = 2) = \rho^{(n)} - \rho; \quad \mathbb{P}(I^{(n)} = 0) = 1 - \rho^{(n)}. \tag{42}$$

When $I^{(n)} = 1$, let us define the following related epochs; see Figure 5:

 $t_*^{(n)}$ = the arrival epoch of the tagged customer;

 $t_1^{(n)}$ = the beginning epoch of the *B*-period during which the tagged customer arrives;

 $t_2^{(n)}$ = the ending epoch of the *B*-period during which the tagged customer arrives;

 $t_3^{(n)} =$ the beginning epoch of the next subservice after $t_2^{(n)}$;

 $t_4^{(n)} =$ the beginning epoch of the service for the tagged customer.


Figure 5: Arrival and beginning of the service for a tagged customer in the ordinary M/G/1 with random order service and service time distribution $F_{B^{(n)}}$.

When $I^{(n)} = 1$, let $\mathcal{A}^{(n)}$ denote the number of exogenous arrivals during $(t_1^{(n)}, t_2^{(n)})$ excluding the tagged customer. Given $I^{(n)} = 1$, $N^{(n)}(t_1^{(n)})$ and $(\mathcal{A}^{(n)}, t_2^{(n)} - t_1^{(n)})$ are independent. Further $N^{(n)}(t_1^{(n)})$ has the same distribution as $M^{(n)}$. Therefore

$$\mathbb{P}(N^{(n)}(t_1^{(n)}) = k, \ \mathcal{A}^{(n)} = l, \ t_2^{(n)} - t_*^{(n)} \le y \mid I^{(n)} = 1) = \mathbb{P}(M^{(n)} = k)\mathbb{P}(\mathcal{A}^{(n)} = l, \ t_2^{(n)} - t_*^{(n)} \le y \mid I^{(n)} = 1).$$
(43)

Given $I^{(n)} = 1$, the joint distribution of $\mathcal{A}^{(n)}$ and $t_2^{(n)} - t_*^{(n)}$ is given by

$$\frac{d}{dy}\mathbb{P}(\mathcal{A}^{(n)}=l, t_2^{(n)}-t_*^{(n)} \le y \mid I^{(n)}=1) = a_l(y), \quad l=0,1,2,\ldots, \ y \ge 0,$$
(44)

where $a_l(y)$ is given by (32). By (42), (43) and (44), we have

$$\frac{d}{dy}\mathbb{P}(I^{(n)} = 1, \ N^{(n)}(t_1^{(n)}) = k, \ \mathcal{A}^{(n)} = l, \ t_2^{(n)} - t_*^{(n)} \le y \mid I^{(n)} = 1) = \rho\mathbb{P}(M^{(n)} = k)a_l(y)$$
(45)

On the other hand,

$$\mathbb{P}(t_4^{(n)} - t_*^{(n)} > x \mid I^{(n)} = 1, \ N^{(n)}(t_1^{(n)}) = k, \ \mathcal{A}^{(n)} = l, \ t_2^{(n)} - t_*^{(n)} = y) \\
= \ \mathbb{P}(t_4^{(n)} - t_2^{(n)} > x - y \mid I^{(n)} = 1, \ N^{(n)}(t_1^{(n)}) = k, \ \mathcal{A}^{(n)} = l, \ t_2^{(n)} - t_*^{(n)} = y) \\
\ge \ \mathbb{P}(t_4^{(n)} - t_3^{(n)} > x - y \mid I^{(n)} = 1, \ N^{(n)}(t_1^{(n)}) = k, \ \mathcal{A}^{(n)} = l, \ t_2^{(n)} - t_*^{(n)} = y) \\
= \ \mathbb{P}(t_4^{(n)} - t_3^{(n)} > x - y \mid I^{(n)} = 1, \ N^{(n)}(t_2^{(n)}) = k + l + 1) \\
\ge \ \mathbb{P}(t_4^{(n)} - t_3^{(n)} > x - y \mid I^{(n)} = 1, \ N^{(n)}(t_3^{(n)} -) = k + l + 1),$$
(46)

where the first inequality follows from $t_4^{(n)} - t_2^{(n)} \ge t_4^{(n)} - t_3^{(n)}$ and the last inequality follows from $N^{(n)}(t_2^{(n)}) \le N^{(n)}(t_3^{(n)}-)$. When $I^{(n)} = 1$ and $N^{(n)}(t_3^{(n)}-) = k+l+1$, the tagged customer and the other k+l customers are waiting for services immediately before $t_3^{(n)}$. Therefore when $I^{(n)} = 1$ and $N^{(n)}(t_3^{(n)}-) = k+l+1$, we have the following at time $t_3^{(n)}$:

- a *B*-period begins without service completion with probability $\frac{\lambda}{n\nu+\lambda}$;
- a service is completed and the tagged customer starts a service with probability $\frac{n\nu}{n\nu+\lambda}\frac{1}{k+l+1}$;
- a service is completed and a customer among the other k + l customers starts a service with probability $\frac{n\nu}{n\nu+\lambda}\frac{k+l}{k+l+1}$.

Bara Kim

Thus

$$\mathbb{P}(t_4^{(n)} - t_3^{(n)} > x - y \mid I^{(n)} = 1, N^{(n)}(t_3^{(n)} -) = k + l + 1) \\
= \frac{\lambda}{n\nu + \lambda} \Phi_{k+l+1}^{(n)} + \frac{n\nu}{n\nu + \lambda} \frac{1}{k+l+1} U + \frac{n\nu}{n\nu + \lambda} \frac{k+l}{k+l+1} \Phi_{k+l}^{(n)}(x - y).$$
(47)

Substituting (47) into (46) leads to

$$\mathbb{P}(t_4^{(n)} - t_*^{(n)} > x \mid I^{(n)} = 1, \ N^{(n)}(t_1^{(n)}) = k, \ \mathcal{A}^{(n)} = l, \ t_2^{(n)} - t_*^{(n)} = y) \\
\geq \frac{\lambda}{n\nu + \lambda} \Phi_{k+l+1}^{(n)} + \frac{n\nu}{n\nu + \lambda} \frac{1}{k+l+1} U + \frac{n\nu}{n\nu + \lambda} \frac{k+l}{k+l+1} \Phi_{k+l}^{(n)}(x-y).$$
(48)

By (45) and (48),

$$\overline{F}_{W^{(n)}}(x) \geq \mathbb{P}(I^{(n)} = 1, t_{4}^{(n)} - t_{*}^{(n)} > x) \\
= \rho \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M^{(n)} = k) a_{l}(y) \\
\times \mathbb{P}(t_{4}^{(n)} - t_{*}^{(n)} > x \mid I^{(n)} = 1, N^{(n)}(t_{1}^{(n)}) = k, \mathcal{A}^{(n)} = l, t_{2}^{(n)} - t_{*}^{(n)} = y) dy \\
\geq \rho \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M^{(n)} = k) a_{l}(y) \\
\times \overline{\frac{\lambda}{n\nu+\lambda} \Phi_{k+l+1}^{(n)} + \frac{n\nu}{n\nu+\lambda} \frac{1}{k+l+1}U + \frac{n\nu}{n\nu+\lambda} \frac{k+l}{k+l+1} \Phi_{k+l}^{(n)}}(x-y) dy, \quad (49)$$

which is written as (40) with $\Psi^{(n)}$ in (41).

Step 3. The assertion (23) holds.

Proof. For $x \ge 0$

$$\overline{\Psi}(x) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M=k)a_{l}(y)$$

$$\times \overline{\frac{\nu}{(k+l+1)\nu+\lambda}U + \frac{(k+l)\nu}{(k+l+1)\nu+\lambda}\Phi_{k+l} + \frac{\lambda}{(k+l+1)\nu+\lambda}\Phi_{k+l+1}(x-y)dy}$$

$$\leq \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}((M-n)^{+} = k)a_{l}(y)$$

$$\times \overline{\frac{\nu}{(k+n+l+1)\nu+\lambda}U + \frac{(k+n+l)\nu}{(k+n+l+1)\nu+\lambda}\Phi_{k+l} + \frac{\lambda}{(k+n+l+1)\nu+\lambda}\Phi_{k+l+1}(x-y)dy}}$$

$$\leq \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}((M-n)^{+} = k)a_{l}(y)$$

$$\times \overline{\frac{n\nu}{n\nu+\lambda}\frac{1}{k+n+l+1}U + \frac{n\nu}{n\nu+\lambda}\frac{k+n+l}{k+n+l+1}\Phi_{k+n+l} + \frac{\lambda}{n\nu+\lambda}\Phi_{k+n+l+1}(x-y)dy}, \quad (50)$$

where the last inequality follows from

$$\overline{U}(x) \leq \overline{\Phi_{k+l}^{(n)}}(x) \leq \overline{\Phi_{k+l+1}^{(n)}}(x);$$

$$\frac{\nu}{(k+n+l+1)\nu+\lambda} \geq \frac{n\nu}{n\nu+\lambda} \frac{1}{k+n+l+1};$$

$$\frac{\nu}{(k+n+l+1)\nu+\lambda} \leq \frac{\lambda}{n\nu+\lambda}.$$

We have, by (27),

$$\overline{\Phi}_{k+n+l}(x) \leq \overline{\left(\frac{n}{k+n+l}U + \frac{k+l}{k+n+l}\Phi_{k+l}^{(n)}\right) * H^{*2n}}(x),$$
(51)

and by (28),

$$\overline{\Phi}_{k+n+l+1}(x) \leq \overline{\Phi_{k+l+1}^{(n)} * H^{*2n}(x)}.$$
 (52)

Substituting (51) and (52) into (50) yields

1

$$\overline{\Psi}(x) \leq \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}((M-n)^{+} = k)a_{l}(y) \\
\times \overline{\frac{n\nu}{n\nu+\lambda} \frac{1}{k+n+l+1}U + \frac{n\nu}{n\nu+\lambda} \frac{k+n+l}{k+n+l+1} \left(\frac{n}{k+n+l}U + \frac{k+l}{k+n+l}\Phi_{k+l}^{(n)}\right) * H_{1}^{*2n} + \frac{\lambda}{n\nu+\lambda}\Phi_{k+l+1}^{(n)} * H_{1}^{*2n}}(x-1)} \\
\leq \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}((M-n)^{+} = k)a_{l}(y) \\
\times \overline{\left(\frac{n\nu}{n\nu+\lambda} \frac{n+1}{k+n+l+1}U + \frac{n\nu}{n\nu+\lambda} \frac{k+l}{k+n+l+1}\Phi_{k+l}^{(n)} + \frac{\lambda}{n\nu+\lambda}\Phi_{k+l+1}^{(n)}\right) * H_{1}^{*2n}}(x-y)dy.$$

Lemma 6 and (53) yield

$$\overline{\Psi}(x) \leq \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M^{(n)} = k) a_{l}(y) \\
\times \overline{\left(\frac{n\nu}{n\nu+\lambda} \frac{n+1}{k+n+l+1}U + \frac{n\nu}{n\nu+\lambda} \frac{k+l}{k+n+l+1} \Phi_{k+l}^{(n)} + \frac{\lambda}{n\nu+\lambda} \Phi_{k+l+1}^{(n)}\right) * H_{1}^{*2n}}(x - y) dy \\
\leq \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \int_{0}^{\infty} \mathbb{P}(M^{(n)} = k) a_{l}(y) \\
\times \overline{\left(\frac{n\nu}{n\nu+\lambda} \frac{1}{k+l+1}U + \frac{n\nu}{n\nu+\lambda} \frac{k+l}{k+l+1} \Phi_{k+l}^{(n)} + \frac{\lambda}{n\nu+\lambda} \Phi_{k+l+1}^{(n)}\right) * H_{1}^{*2n}}(x - y) dy \\
= \overline{\Psi^{(n)} * H_{1}^{*2n}}(x).$$
(54)

By (33), (54) and (40),

$$\overline{F}_W(x) \le \overline{F_{W^{(n)}} * E_{\nu+\lambda} * H_1^{*2n}}(x).$$
(55)

Lemma 4 with (16) yields

$$\overline{F}_{W^{(n)}}(x) \sim c_n x^{1-\beta} L(x) \text{ as } x \to \infty,$$
 (56)

where

$$c_n = \frac{\rho_n}{1 - \rho_n} h(\rho_n, \beta) \frac{1}{\beta - 1} \frac{1}{\mathbb{E}B^{(n)}}.$$

By (56) and Proposition 1, we have

$$\overline{F_{W^{(n)}} * E_{\nu+\lambda} * H_1^{*2n}}(x) \sim c_n x^{1-\beta} L(x) \quad \text{as } x \to \infty,$$

which together with (55) leads to

$$\limsup_{x \to \infty} \frac{\overline{F}_W(x)}{x^{1-\beta}L(x)} \leq c_n.$$
(57)

Finally, we obtain (23) by letting $n \rightarrow \infty$ in (57).

References

- [1] J.R. Artalejo, A classified bibliography of research on retrial queues: progress in 1990-1999, *Top*, Vol. 7, Issue 2, 187-211, 1999.
- [2] S. Asmussen, Applied Probability and Queues, Second Edition, Springer, 2003.
- [3] S. Asmussen, C. Klüppelberg and K. Sigman, Sampling at subexponential times, with queueing applications, Stochastic Processes and their Applications, 79, 265-286, 1999.
- [4] N.H. Bingham, C.M. Goldie and J.L. Teugels, *Regular variation*, Cambridge University Press, Cambridge, 1987.
- [5] O.J. Boxma, S. G. Foss, J.-M. Lasgouttes and R. Núñez-Queija, Waiting time asymptotics in the single server queue with service in random order, *Queueing Systems*, 46, Issue 1-2, 35-73, 2004.
- [6] B.D. Choi and Y. Chang, Single server retrial queues with priority calls, *Math. Comput. Model.*, 30, 7-32, 1999.
- [7] G.I. Falin and J.G.C. Templeton, Retrial Queues, Chapman & Hall, London, 1997.
- [8] A. De Meyer and J.L. Teugels, On the asymptotic behaviour of the distribution of the busy period and service time in M/G/1 queue, *J. Appl. Prob.* 17, 802-813, 1980.
- [9] B. Kim, Jisu Lee and In-Suk Wee, Tail asymptotics for the fundamental period in the MAP/G/1 Queue, *Queueing Systems*, in revision.
- [10] B. Kim and K. Sohraby, Tail behavior of the queue size and waiting time in a queue with discrete autoregressive arrivals, *Adv. Appl. Prob.*, 38, 1116-1131, 2006.
- [11] W. Shang, L. Liu and Q. Li, Tail asymptotics for the queue length in an M/G/1 retrial queue, *Queueing Systems*, 52, 193-198, 2006.
- [12] K. Sigman, Appendix: A primer on heavy-tailed distributions, *Queueing Systems*, 33, 261-275, 1999.

A Retrial Queueing System with a Variable Number of Active Servers

Rein Nobel

Department of Econometrics, Vrije Universiteit, Amsterdam

Abstract

A retrial queueing model is considered with Poisson input and an unlimited number of servers. At any epoch only a finite number of the servers are active, the others are called dormant. An active server is always in one of two possible states, idle or busy. When upon arrival of a customer at least one of the active servers is idle, the newly arrived customer goes into service immediately, making the idle server busy. When at an arrival epoch all active servers are busy, the decision must be made to send the newly arrived customer into orbit, or to activate a dormant server for immediate service of the arrived customer. Customers in orbit try to reenter the system after an exponentially distributed retrial time. At service completion epochs the decision must be made to keep the newly become idle server active, or to make this server dormant. The service times of the customers are independent and have a Coxian-2 distribution. Given specific costs for activating servers, keeping servers active and a holding cost for customers staying in orbit, the problem is when to activate and shut down servers in order to minimize the long-run average cost per unit time. Using Markov decision theory an efficient algorithm is discussed for calculating an optimal policy.

Keywords: retrial model, semi-Markov decision model, fictitious decision epochs.

1 Introduction

In recent years we have seen a considerable increase in the number of call centers. Both private companies and governmental institutions use these centers for answering questions from their customers. As a consequence, a lot of research has been undertaken to study the random behavior of these call centers. Not surprisingly, queueing theory plays a dominant role in this research. Starting with Erlang's Loss model, many papers have been written in which besides lost calls, also retrials and abandonments have been incorporated. For a tutorial overview we refer to [3] and [5] and the references therein. A nice introductory paper on abandonments is [7] in which the so-called Palm/Erlang-A model is discussed. For the impact of retrials on call center performance we refer to [1]. The main topic in call center research is to find a balance between service quality, expressed e.g. in waiting-time characteristics, and the cost of operation, expressed, e.g. in the number of active servers. Hence, the so-called 'staffing problem' is a central topic in most of the call center literature. Formulated in the terminology of queueing theory, this problem can be described as follows. Given all the relevant parameters for some multi-server queueing model in which customers have the option to abandon the system after not having been served within some random time, and/or to retry entrance to the system some random time later after an unsuccessful arrival, the question is how many active servers (agents) must be available to guarantee a required balance between service quality and operational cost. The given parameters of the queueing model include the arrival rate, the service rate, the abandonment rate and the retrial rate. The design parameter is the number of agents. Taking a model with *all* parameter values fixed as a starting point, most papers give a descriptive analysis for the steady-state behavior of the system. Due to their complexity these models often do not allow for a practically feasible exact solution. This is *a fortiori* the case for the transient behavior of the systems and/or when parameters are time-dependent. To cope with this intractability, all kinds of approximations are considered, such as fluid and diffusion approximations, e.g. see [6]. We will not give an extensive overview of all the research on call centers done so far, but the point we want to make is that most of this research is descriptive in character: the steady-state or transient behavior of the models is studied for a set of *given* parameters. Much less attention has been dedicated to finding dynamic operational policies as a solution for the staffing problem.

So, instead of giving a descriptive analysis of some queueing model with a fixed number of servers, we propose to study the staffing problem as a dynamic optimization problem: let the number of active servers depend on the current congestion of the system, expressed in the number of busy servers and the number of waiting customers, and increase or decrease the number of active servers depending on instantaneous changes of the congestion as a consequence of an arrival, a departure or an abandonment. To pursue this idea of dynamic manpower planning, we propose to study a retrial multi-server queueing model with an adaptable number of servers. To limit the calculational burden, we do not consider abandonments in this paper, but we want to underline that abandonments can be easily incorporated in the model, if one wishes to do so. For this retrial model, to be described in detail below, we will use Markov decision theory to calculate an operating policy, for which a subtle balance between the costs of congestion and the operational costs is minimized.

In a standard retrial model (see [2] for a monograph on retrial queues) customers who find all servers busy try to enter the system some time later. We say that the customer goes into orbit. Nowadays it is very common that the system knows at any time how many customers are in orbit (we can simply register unsuccessful calls). So, this information can be used in the determination of the number of active servers. In our model the number of servers is unlimited but at any moment only a finite number is active (the others are called dormant), and this number is under control of the management of the system. Hence, we consider a multiserver retrial queueing system with a controllable number of active servers, who can be idle or busy. When upon arrival of a customer no idle server is available, a choice must be made to activate a dormant server, or send the newly arrived customer into orbit. Upon a service completion it must be decided to shut down an active idle server (i.e. make him dormant), or keep him active for possible new arrivals. Of course, these decisions must be guided by some optimization criterion, i.e. a cost structure must be introduced in the model. Given this cost structure, the problem is to find the strategy for activating and shutting down servers, for which the long-run average cost per unit time is minimal. This strategy is a so-called stationary dynamic strategy, i.e. the decisions prescribed by the strategy take into account all the relevant information available at the decision epochs, not more and not less, in other words, the decisions are based on the complete state description of the system. By choosing a specific stochastic structure with respect to the probability distributions involved, we can describe our problem in terms of a semi-Markov decision model. A straightforward application of a standard algorithm from Markov decison theory is not feasible here, due to the large state space. By introducing so-called fictitious decision epochs we will show how to overcome this obstacle.

In Section 2 the queueing model is described in detail. Section 3 describes a semi-Markov decision model and the value-iteration algorithm to calculate the optimal policy for which the long-run average cost per unit time is minimal. In Section 4 some numerical results are given.

2 Description of the model

We consider a queueing model with retrials and a controllable number of active servers. The number of servers is unlimited, but at any epoch only a finite number of servers is active, either idle or busy. The non-active servers are called dormant. For idle servers linear operating costs α per server per unit time are incurred, whereas for busy servers these costs are γ per server per unit time ($\gamma > \alpha$). Customers arrive at the system according to a Poisson process with rate λ . Each customer requires a service time denoted by the generic variable S, and the service times of different customers are independent. We give the service times S a Coxian-2 distribution with parameters b, μ_1 and μ_2 with 0 < b < 1 and $\mu_1 < \mu_2$. We recall here that this says that S is, with probability b, distributed as a sum of two independent exponential phases, say S_1 and S_2 with mean $1/\mu_1$ and $1/\mu_2$, respectively, and with probability 1 - b, S is distributed as one exponential phase S_1 with mean $1/\mu_1$. As we will see in the next section, the Coxian-2 distribution is a very convenient choice for the service times, due to the memoryless property of the exponential phases. Also it is easy to fit a Coxian-2 distribution when only the first two moments are given (see [10] for further details). When upon arrival of a customer at least one idle server is available, the customer immediately starts its service, reducing the number of idle servers by one. When no idle server is present, either the newly arrived customer goes into orbit, or a dormant server is activated to serve the customer immediately. Activating a dormant server requires a set-up cost K. Customers in orbit try to enter the system again after an exponentially distributed retrial time R with mean $1/\nu$. The different retrial times are independent. For customers in orbit linear holding costs h per customer per unit time are incurred. At service completion epochs the choice must be made between keeping the server active (in the idle state) or making him dormant. So, the question is when to activate a dormant server upon an arrival and when to shut down an active server upon a departure, in order to minimize the long-run average cost per unit time.

To calculate this optimal policy, in the next section we will formulate the model as a *semi-Markov decision model*. To avoid the problem of an infinite state space we will give our analysis for a *truncated* model, i.e. we limit the number of available servers to a finite number C and the maximum number of customers allowed to be in orbit will be taken M. A customer in orbit is considered to stay in orbit until he is accepted for service. So, to complete the description of the truncated model, we must specify precisely what to do upon arrivals and departures in the boundary situations:

- New arrivals who find *M* customers in orbit and less than *C* servers busy will always be accepted,
- An arrival from orbit will always be accepted when *M* customers are in orbit and less than *C* servers are busy,
- New arrivals who find *M* customers in orbit and *C* servers busy are rejected,
- An arrival from orbit will stay in orbit when upon arrival C servers are busy,
- New arrivals who find *C* servers busy and less than *M* customers in orbit are always sent to orbit.

By taking *C* and *M* sufficiently large the fraction of customers which will be rejected is negligible, so our numerical results will be valid for the untruncated model as well.

3 The semi-Markov decision model

We assume that the reader is acquainted with the concepts of Markov decision theory (see [9] and [10] for thorough introductions to this subject), so we will not give an extensive description of the building blocks of a semi-Markov decision model. We just recall that we have to specify a state space S, action sets A(s) for each state $s \in S$, a matrix of transition probabilities p[s'|s, a]for $s', s \in S$ and $a \in A(s)$, expected one-step costs $\eta[s, a]$ for $s \in S$ and $a \in A(s)$, and the expected sojourn times $\tau[s, a]$ for each state $s \in S$ and $a \in A(s)$. All these building blocks will now be specified for the controllable queueing model described in the previous section. To describe a semi-Markov decision model with a sparse matrix of transition probabilities, it is very convenient to introduce so-called fictitious decision epochs (see [8] and [10]). According to the model description the only decision epochs are the arrival epochs at which no idle server is present and the epochs of service completion. To guarantee a sparse matrix of transition probabilities, we include all arrival epochs, and phase completion epochs of service times as decision epochs as well. At these latter epochs no real decision is made, we just leave the system as it is. We denote this 'no action' decision by 0. At service completion epochs two decisions are possible, shut down the server who has just become idle (denoted by -1), or leave the system as it is (again denoted by 0). At arrival epochs we leave the system as it is, when an idle server is available. A real decision has to be made only when all active servers are busy. Then we can send the newly arrived job into orbit (denoted by 0), or we can activate a dormant server (denoted by 1). At this point it will be clear why our choice for Coxian service times is so convenient: due to the exponential phases of the service time it is sufficient to know whether a service is in its first phase or in its second phase, and this enables a simple description of the state of the system, as we will see next. In fact we can introduce the following state description at the decision epochs,

 $(i, j_1, j_2, k, e), \quad i = 0, 1, \dots; \quad j_1, j_2 = 0, 1, \dots; \quad k = 0, 1, \dots; \quad e = 0, 1, -1, -2,$

with the following interpretation

- *i* is the number of idle servers,
- j_1 is the number of busy servers in the first service phase,
- j_2 is the number of busy servers in the second service phase,
- *k* is the number of jobs in orbit,
- *e* describes the type of event that occurred: e = 0 denotes a new arrival (from the Poisson stream), e = 1 denotes an arrival from the orbit, e = -1 stands for a phase completion of an ongoing service time, which leads to the next phase of this service time, and e = -2 stands for the completion of a service time.

We emphasize that the numbers i, j_1, j_2, k always refer to the numbers just *after* the 'event' *e* has occurred, but *before* the decision is made. Specifically, a customer in orbit is considered to stay in orbit until he is accepted for service. Notice that not all combinations (i, j_1, j_2, k, e) refer to real states, e.g. the states $(0, j_1, j_2, k, -2)$ do not exist because upon a departure (e = -2) at least one server must be idle.

Next, we will specify the elements of the semi-Markov decision model

$$(\mathcal{S}, \{\mathcal{A}(s), s \in \mathcal{S}\}, \{\tau[s, a], s \in \mathcal{S}, a \in \mathcal{A}(s)\}, \{\eta[s, a], s \in \mathcal{S}, a \in \mathcal{A}(s)\}, \{p[t|s, a], s, t \in \mathcal{S}, a \in \mathcal{A}(s)\})$$

which describes the retrial queueing model with a variable number of active servers. As already indicated above, the state space S is taken as

$$\mathcal{S} = \{(i, j_1, j_2, k, e) \mid i = 0, 1, \dots; j_1, j_2 = 0, 1, \dots; i + j_1 + j_2 \le C; k = 0, 1, \dots, M; e = 0, 1, -1, -2\}$$

The action sets $\mathcal{A}(s)$ are very simple. At each service completion epoch the decision must be made to shut down an idle server or to leave the system as it is. When at an arrival epoch no idle servers are available the decision must be made to switch on a dormant server or send the newly arrived job into orbit. Because we do not allow more than M customers in orbit, we always accept an arriving customer when M customers are in orbit, unless the number of active servers is C. In the latter case we reject primary customers from the Poisson stream and leave the arriving customers from orbit in orbit until the number of busy servers has dropped below C. These remarks lead to the following action sets.

For the one-step transition probabilities p[s'|s, a], denoting the conditional probability that, given action a is taken in state s, at the next decision epoch the state is s', we first consider the real decision epochs, i.e. the service completion epochs, and the arrival epochs when no idle server is available. First we give the one-step transition probabilities given that a service completion has occurred. So the decision a is either 0 (keep all idle servers active) or -1 (switch off an idle server).

$$\begin{split} p[(i+a,j_1,j_2,k,0) \mid (i,j_1,j_2,k,-2),a] &= \frac{\lambda}{\lambda+j_1\mu_1+j_2\mu_2+k\nu}, \\ p[(i+a,j_1,j_2,k,1) \mid (i,j_1,j_2,k,-2),a] &= \frac{k\nu}{\lambda+j_1\mu_1+j_2\mu_2+k\nu}, \\ p[(i+a,j_1-1,j_2+1,k,-1) \mid (i,j_1,j_2,k,-2),a] &= \frac{bj_1\mu_1}{\lambda+j_1\mu_1+j_2\mu_2+k\nu}, \\ p[(i+a+1,j_1-1,j_2,k,-2) \mid (i,j_1,j_2,k,-2),a] &= \frac{(1-b)j_1\mu_1}{\lambda+j_1\mu_1+j_2\mu_2+k\nu}, \\ p[(i+a+1,j_1,j_2-1,k,-2) \mid (i,j_1,j_2,k,-2),a] &= \frac{j_2\mu_2}{\lambda+j_1\mu_1+j_2\mu_2+k\nu}. \end{split}$$

Of course, the third and fourth transition is only possible when $j_1 > 0$, and the last transition requires that $j_2 > 0$.

Next, we write down these probabilities given that an arrival has taken place, a primary arrival (e = 0) or an arrival from orbit (e = 1), and no idle servers are present. Now, the decision a = 0 stands for 'send (keep) he arrived customer (in)to orbit' and a = 1 denotes

'switch on a dormant server'.

$$p[(0, j_1 + a, j_2, k + 1 - a - e, 0) | (0, j_1, j_2, k, e), a] = \frac{\lambda}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(0, j_1 + a, j_2, k + 1 - a - e, 1) | (0, j_1, j_2, k, e), a] = \frac{(k + 1 - a - e)\nu}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(0, j_1 + a - 1, j_2 + 1, k + 1 - a - e, -1) | (0, j_1, j_2, k, e), a] = \frac{b(j_1 + a)\mu_1}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a - 1, j_2, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{(1 - b)(j_1 + a)\mu_1}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1, j_2, k, e), a] = \frac{j_2\mu_2}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \\ p[(1, j_1 + a, j_2 - 1, k + 1 - a - e, -2) | (0, j_1 + j_2\mu_2 + (k - a - e))\mu]$$

Similarly, we can treat the fictitious decision epochs, the arrival epochs with i > 0 idle servers available, and the phase completion epochs. The only decision is now 0 (leave the system as it is).

$$\begin{split} p[(i-1,j_1+1,j_2,k-e,0) \mid (i,j_1,j_2,k,e),0] &= \frac{\lambda}{\lambda + (j_1+1)\mu_1 + j_2\mu_2 + (k-e)\nu},\\ p[(i-1,j_1+1,j_2,k-e,1) \mid (i,j_1,j_2,k,e),0] &= \frac{(k-e)\nu}{\lambda + (j_1+1)\mu_1 + j_2\mu_2 + (k-e)\nu},\\ p[(i-1,j_1,j_2+1,k-e,-1) \mid (i,j_1,j_2,k,e),0] &= \frac{b(j_1+1)\mu_1}{\lambda + (j_1+1)\mu_1 + j_2\mu_2 + (k-e)\nu},\\ p[(i,j_1,j_2,k-e,-2) \mid (i,j_1,j_2,k,e),0] &= \frac{(1-b)(j_1+1)\mu_1}{\lambda + (j_1+1)\mu_1 + j_2\mu_2 + (k-e)\nu},\\ p[(i,j_1+1,j_2-1,k-e,-2) \mid (i,j_1,j_2,k,e),0] &= \frac{j_2\mu_2}{\lambda + (j_1+1)\mu_1 + j_2\mu_2 + (k-e)\nu},\\ p[(i,j_1,j_2,k,0) \mid (i,j_1,j_2,k,-1),0] &= \frac{\lambda}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu},\\ p[(i,j_1-1,j_2+1,k,-1) \mid (i,j_1,j_2,k,-1),0] &= \frac{bj_1\mu_1}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu},\\ p[(i+1,j_1-1,j_2,k,-2) \mid (i,j_1,j_2,k,-1),0] &= \frac{(1-b)j_1\mu_1}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu},\\ p[(i+1,j_1,j_2-1,k,-2) \mid (i,j_1,j_2,k,-1),0] &= \frac{j_2\mu_2}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu}. \end{split}$$

Next, we consider the boundary cases, i.e. the number of customers in orbit is M and/or the number of busy servers is C. First, we look at arrivals finding M customers in orbit and less than C servers busy $(j_1 + j_2 < C)$. As stated above in this case we always accept a new

customer.

$$\begin{array}{lll} p[(0,j_{1}+1,j_{2},M-e,0)\mid(0,j_{1},j_{2},M,e),1] &=& \displaystyle\frac{\lambda}{\lambda+(j_{1}+1)\mu_{1}+j_{2}\mu_{2}+(M-e)\nu},\\ p[(0,j_{1}+1,j_{2},M-e,1)\mid(0,j_{1},j_{2},M,e),1] &=& \displaystyle\frac{(M-e)\nu}{\lambda+(j_{1}+1)\mu_{1}+j_{2}\mu_{2}+(M-e)\nu},\\ p[(0,j_{1},j_{2}+1,M-e,-1)\mid(0,j_{1},j_{2},M,e),1] &=& \displaystyle\frac{b(j_{1}+1)\mu_{1}}{\lambda+(j_{1}+1)\mu_{1}+j_{2}\mu_{2}+(M-e)\nu},\\ p[(1,j_{1},j_{2},M-e,-2)\mid(0,j_{1},j_{2},M,e),1] &=& \displaystyle\frac{(1-b)(j_{1}+1)\mu_{1}}{\lambda+(j_{1}+1)\mu_{1}+j_{2}\mu_{2}+(M-e)\nu},\\ p[(1,j_{1},j_{2}-1,M-e,-2)\mid(0,j_{1},j_{2},M,e),1] &=& \displaystyle\frac{j_{2}\mu_{2}}{\lambda+(j_{1}+1)\mu_{1}+j_{2}\mu_{2}+(M-e)\nu}. \end{array}$$

Now we consider arrivals who find *C* servers busy (so $j_1 + j_2 = C$) and less than *M* customers in orbit. They are always sent into orbit.

$$\begin{split} p[(0,j_1,j_2,k+1-e,0) \mid (0,j_1,j_2,k,e),0] &= \frac{\lambda}{\lambda+j_1\mu_1+j_2\mu_2+(k+1-e)\nu},\\ p[(0,j_1+1,j_2,k+1-e,1) \mid (0,j_1,j_2,k,e),0] &= \frac{(k+1-e)\nu}{\lambda+j_1\mu_1+j_2\mu_2+(k+1-e)\nu},\\ p[(0,j_1-1,j_2+1,k+1-e,-1) \mid (0,j_1,j_2,k,e),0] &= \frac{bj_1\mu_1}{\lambda+j_1\mu_1+j_2\mu_2+(k+1-e)\nu},\\ p[(1,j_1-1,j_2,k+1-e,-2) \mid (0,j_1,j_2,k,e),0] &= \frac{(1-b)j_1\mu_1}{\lambda+j_1\mu_1+j_2\mu_2+(k+1-e)\nu},\\ p[(1,j_1,j_2-1,k+1-e,-2) \mid (0,j_1,j_2,k,e),0] &= \frac{j_2\mu_2}{\lambda+j_1\mu_1+j_2\mu_2+(k+1-e)\nu}.\end{split}$$

Finally, we look at arrival epochs when C servers are busy and M customers are in orbit. Then new arrival are rejected and arrivals from orbit stay in orbit. So we get, $(j_1 + j_2 = C, e = 0, 1)$

$$\begin{split} p[(0,j_1,j_2,M,0) \mid (0,j_1,j_2,M,e),0] &= \frac{\lambda}{\lambda+j_1\mu_1+j_2\mu_2+M\nu}, \\ p[(0,j_1,j_2,M,1) \mid (0,j_1,j_2,M,e),0] &= \frac{M\nu}{\lambda+j_1\mu_1+j_2\mu_2+M\nu}, \\ p[(0,j_1-1,j_2+1,M,-1) \mid (0,j_1,j_2,M,e),0] &= \frac{bj_1\mu_1}{\lambda+j_1\mu_1+j_2\mu_2+M\nu}, \\ p[(1,j_1-1,j_2,M,-2) \mid (0,j_1,j_2,M,e),0] &= \frac{(1-b)j_1\mu_1}{\lambda+j_1\mu_1+j_2\mu_2+M\nu}, \\ p[(1,j_1,j_2-1,M,-2) \mid (0,j_1,j_2,M,e),0] &= \frac{j_2\mu_2}{\lambda+j_1\mu_1+j_2\mu_2+M\nu}. \end{split}$$

Let us next consider the $\tau[s, a]$, i.e. the expected time until the next decision epoch given that in state *s* action *a* is chosen.

$$\begin{aligned} \tau[(i, j_1, j_2, k, e), 0] &= \frac{1}{\lambda + (j_1 + 1)\mu_1 + j_2\mu_2 + (k - e)\nu}, \quad i = 1, 2, \dots, \\ \tau[(0, j_1, j_2, k, e), a] &= \frac{1}{\lambda + (j_1 + a)\mu_1 + j_2\mu_2 + (k + 1 - a - e)\nu}, \quad a = 0, 1, \\ \tau[(i, j_1, j_2, k, -1), 0] &= \frac{1}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu}, \quad i = 0, 1, 2, \dots, \\ \tau[(i, j_1, j_2, k, -2), a] &= \frac{1}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu}, \quad i = 1, 2, \dots, \quad a = 0, -1. \end{aligned}$$

To complete the formulation of the Markov-decision model we must specify the costs $\eta[s, a]$, i.e., the total expected costs incurred until the next decision epoch when in state *s* action *a* is taken. We give a few examples.

$$\begin{split} \eta[(i,j_1,j_2,k,e),0] &= \frac{1}{\lambda + (j_1+1)\mu_1 + j_2\mu_2 + (k-e)\nu}(i\alpha + (j_1+1+j_2)\gamma + (k-e)h), \quad i = 1,2,\ldots, \\ \eta[(0,j_1,j_2,k,e),a] &= aK + \frac{1}{\lambda + (j_1+a)\mu_1 + j_2\mu_2 + (k+1-a-e)\nu}((j_1+a+j_2)\gamma + (k+1-a-e)h), \\ \eta[(i,j_1,j_2,k,-1),0] &= \frac{1}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu}(i\alpha + (j_1+j_2)\gamma + kh), \quad i = 0,1,2,\ldots, \\ \eta[(i,j_1,j_2,k,-2),a] &= \frac{1}{\lambda + j_1\mu_1 + j_2\mu_2 + k\nu}((i-a)\alpha + (j_1+j_2)\gamma + kh), \quad i = 1,2,\ldots; \quad a = 0,-1. \end{split}$$

Once all the elements of the Markov-decision model are known, we can use the value-iteration algorithm to calculate the optimal switching strategy. We give the formulation of the algorithm in general terms (see also [10]). First choose a positive number τ with $\tau \leq \min_{s,a} \tau[s, a]$ and a tolerance number ϵ , e.g., $\epsilon = 10^{-6}$.

INIT For all $s \in S$, choose nonnegative numbers $W_0(s)$ with $W_0(s) \leq \min_a \{\eta[s, a] / \tau[s, a]\}$. Let n := 1.

LOOP For all $s \in S$, calculate

$$W_n(s) = \min_{a \in \mathcal{A}(s)} \left[\frac{\eta[s,a]}{\tau[s,a]} + \frac{\tau}{\tau[s,a]} \sum_{t \in \mathcal{S}} p[t \mid s,a] W_{n-1}(t) + \left\{ 1 - \frac{\tau}{\tau[s,a]} \right\} W_{n-1}(s) \right],$$

and let $D_n(s) \in \mathcal{A}(s)$ be the action that minimizes the right-hand side.

EVAL Compute the bounds,

$$m_n = \min_{s \in S} \{ W_n(s) - W_{n-1}(s) \}, \quad M_n = \max_{s \in S} \{ W_n(s) - W_{n-1}(s) \}.$$

TEST If $M_n - m_n \le \epsilon m_n$ then STOP with the resulting policy D_n , else n := n + 1 and go to LOOP.

This algorithm returns after say n iterations a stationary policy D_n^* that minimizes the longrun average costs per unit time. The (approximate) minimal average costs is calculated as $g^* = (m_n + M_n)/2$.

4 Numerical Results

In this section we will present some numerical results. Because there are many parameters which can be varied we must make a selection. To start, we have chosen to keep the arrival rate and the mean service time constant and we vary only the retrial rate, and the squared coefficient of variation of the service time (we used Gamma normalisation for fitting the parameters of the Coxian-2 distribution; see [10] for the details how to choose the parameters b, μ_1 and μ_2 to guarantee a given mean and squared coefficient of variation). Because the mathematical state-description is more detailed than any reasonable physical state-description, we present a natural heuristic policy, with the corresponding cost, besides the optimal solution. To explain the heuristics, notice that the mathematical state-description in our model contains the phases of the ongoing services which cannot be observed physically. Because in practice we only observe the number of idle servers i, the number of busy servers j, and, by registration, the number in orbit k (in other words on occurrence of the event e the *physical* state is (i, j, k, e), we must base our decisions on this information for all different mathematical states (i, j_1, j_2, k, e) with $j_1 + j_2 = j$. So, we are forced to select one decision in all these latter states, whereas the (mathematically) optimal policy may prescribe different decisions for these states. We have chosen a kind of democratic heuristic rule, defined as follows. When in the majority of the states (i, j_1, j_2, k, e) decision a is the optimal decision, then we prescribe this decision in all corresponding physical states (i, j, k, e) with $j = j_1 + j_2$. In Table 1–2 we present the minimal cost and the corresponding heuristic cost for the following parameter values,

$$\lambda = 3, \quad E[S] = 2, \quad h = 10, \quad \alpha = 20, \quad \gamma = 25, \quad K = 500.$$

In Table 1 the holding cost for staying in the orbit h = 10 and in Table 2 we have chosen h = 1. We vary the retrial rate ν and the squared coefficient of variation of the service time c_S^2 . Notice that the difference between the optimal cost and the (democratic) heuristic cost is negligible for $c_s^2 \leq 1$. This difference turns out to be significant only for high holding costs and very irregular service times.

To give an idea of the 'form' of the strategies for turning on and off servers we present both the primary arrival strategy and one departure strategy for a specific choice of the parameters. In Table 3 and Table 4 we present these optimal (heuristic) strategies for exponential service and in Table 5 and Table 6 for very irregular service times ($c_S^2 = 8$). In these tables the number of customers in orbit is presented horizontally and the number of active servers vertically. Notice that in the tables for the arrival strategies the active servers are all busy (otherwise there is nothing to decide), but for the departure strategies the decisions are not based on the number of active servers alone; we also need to know how many servers are busy. So, for each specific number of active servers, say *i*, to be complete we should present *i* rows, i.e. one row for each possible number of idle servers. To avoid such an overwhelming amount of information in one table, in Table 5 and Table 6 we have made the choice to show only the decisions for the situation that half of the active servers is idle. Finally, in these tables a '0' stands for 'leave the system as it is', so for the arrival strategies: send the newly arrived customer into orbit, and for the departure strategies 'keep the server just becoming idle active', and a '1' for 'turning on a dormant server' (arrival) and 'turning off an idle server' (departure). We see from these tables that the policies are rather insensitive for the squared coefficient of variation of the service time, whereas the associated costs are quite different (see Table 2, for $c_S^2 = 1$ the minimal costs are 178.42, and for $c_s^2 = 8$, ceteris paribus, 193.98). This fact, that optimal policies are rather robust for the variability of the service time, is a well-known phenomenon in the literature on controlled queueing systems.

				N 4		
$\nu \setminus c_S^2$	0.6	0.8	1	2	4	8
0.25	228.53	229.03	229.53	231.76	234.61	234.55
	228.53	229.03	229.53	231.81	235.15	239.44
0.5	216.13	217.18	218.14	221.87	225.43	226.36
	216.16	217.19	218.14	222.00	226.73	231.20
1	207.94	209.05	210.08	214.34	218.90	220.38
	207.96	209.05	210.08	214.44	220.06	226.17
2	203.63	204.71	205.72	210.02	214.94	216.32
	203.66	204.71	205.72	210.12	215.84	222.98
4	200.72	202.35	203.38	207.62	212.65	213.72
	200.74	202.36	203.38	207.69	214.24	220.36

Table 1: Minimal and heuristic cost for $\lambda = 3$, E[S] = 2, h = 10, $\alpha = 20$, $\gamma = 25$, K = 500.

Table 2: Minimal and heuristic cost for $\lambda = 3$, E[S] = 2, h = 1, $\alpha = 20$, $\gamma = 25$, K = 500.

$\nu \setminus c_S^2$	0.6	0.8	1	2	4	8
0.25	192.15	193.24	194.22	197.97	201.89	203.98
	192.19	193.26	194.22	198.20	202.72	205.43
0.5	184.09	185.37	186.55	191.14	196.16	199.01
	184.13	185.40	186.55	191.40	196.80	200.73
1	179.61	180.87	182.05	186.94	192.56	195.57
	179.64	180.87	182.05	187.11	193.08	197.30
2	177.36	178.53	179.67	184.56	190.43	193.36
	177.38	178.54	179.67	184.69	190.90	195.31
4	176.05	177.33	178.42	183.25	189.15	192.02
	176.12	177.33	178.42	183.37	189.82	193.98

References

- [1] S. Aguir, F. Karaesmen, Z. Aksin and F. Chauvet. (2004). The impact of retrials on call center performance. *OR Spectrum* 26: 353–376.
- [2] G. Falin and J. Templeton (1997). Retrial Queues. Chapman & Hall, London.
- [3] N. Gans, G. Koole, A. Mandelbaum (2003). Telephone call centers: Tutorial, review and research prospects. *Manufacturing and Service Operations Management* 5:79-141.
- [4] O. Garnett, A. Mandelbaum and M. Reiman (2002). Designing a call center with impatient customers. *Manufacturing & Service Operations Management* 4: 208–227.
- [5] G. Koole and A. Mandelbaum (2002). Queueing models of call centers, an introduction. *Annals of Operation Research* **113**, 41–59.
- [6] A. Mandelbaum, W. Massey, M. Reiman, A. Stolyar and B. Rider (2002). Queue lengths and waiting times for multiserver queues with abandonment and retrials. *Telecommunication Systems* 21: 149-171.
- [7] A. Mandelbaum and S. Zeltyn (2007). Service engineering in action: The Palm/Erlang-A queue, with applications to call centers. *Advances in Services Innovations*: 17–45, Springer, Berlin.

											D							•			
act \ orb	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3: PRIMARY ARRIVALS STRATEGY	$r_{s}^{2} = 1, \nu$	$\gamma = 4, h = 1, \alpha =$	$= 20, \gamma = 25, K = 500.$
------------------------------------	----------------------	-------------------------------	-------------------------------

- [8] R.D. Nobel and R. Bekker (2001). Optimal control for two queues with one switching server. Proceedings of the National Conference on Mathematical and Computational Models, December 27–28, 2001. PSG College of Technology, Coimbatore, India, 21–33.
- [9] M.L. Puterman (1994). Markov Decision Processes. John Wiley and Sons, New York.
- [10] H.C. Tijms (1994). Stochastic Models: An Algorithmic Approach. John Wiley and Sons, New York.

act \setminus orb	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 4. DEPARTURES STRATEGY $c_{\alpha}^{2} = 1$ $\nu = 4$ h	$=1 \alpha = 20, \gamma = 25 K = 500$
1000 4.001000000000000000000000000000000	= 1, u = 20, f = 20, R = 000

Table 5: PRIMARY ARRIVALS STRATEGY $c_S^2 = 8$, $\nu = 4$, h = 1, $\alpha = 20$, $\gamma = 25$, K = 500.

act \ orb	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6: DEPARTURES STRATEGY $c_S^2 = 8$, $\nu = 4$, h = 1, $\alpha = 20$, $\gamma = 25$, K = 500.

act \ orb	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Cross-Layer Design with Adaptive Modulation and Coding Scheme for QoS Support and Its Theoretical Analysis

Gang Uk Hwang^a and Fumio Ishizaki^b

^aDepartment of Mathematical Sciences and Telecommunication Engineering Program, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea. ^bDepartment of Information and Telecommunication Engineering, Nanzan University, Seto, Japan. Email: guhwang@amath.kaist.ac.kr, fumio@ieee.org

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

Abstract

In this paper, we consider a cross-layer design problem of a wireless network with AMC (Adaptive Modulation and Coding). In our cross-layer design, the average packet transmission error rate at the PHY layer and the packet overflow probability at the MAC layer are simultaneously investigated. In addition, we assume that packet retransmission is allowed for the packets received in error and that a certain level of packet overflow probability at the MAC layer is given as the QoS (Quality of Service) requirement. To capture the joint effect of the packet service process at the MAC layer. For our cross-layer design, we provide a theoretical analysis on the behavior of the effective bandwidth function when we change the AMC scheme at the PHY layer. Based on our theoretical results, we propose a new framework for our cross-layer design, with which we can meet the required packet overflow probability at the MAC layer as low as possible. A numerical example is provided to see the validity of the proposed framework and to investigate its characteristics.

1 Introduction

While available radio spectrum is extremely scarce, the demand for multimedia wireless services requiring QoS (Quality of Service) has been tremendously increasing. Accordingly, enhancing the spectral efficiency in wireless communications is one of the key issues and AMC (Adaptive Modulation and Coding) schemes have been studied extensively as one of promising techniques to solve this problem.

However, most of existing studies on AMC schemes rely on the assumption that data are continuously available at the transmitter. That is, the performance of the Physical (PHY) layer is considered independently of the performance of the Medium Access Control (MAC) layer. However, the performance at the MAC layer is significantly affected by the AMC scheme employed at the PHY layer. Thus, in order to get benefits by employing the AMC at the PHY layer and to provide better network performance, we should consider the interaction between the queue at the MAC layer and the AMC at the PHY layer. This motivates the cross-layer design between the PHY and MAC layers, and there are many papers on the cross-layer design in the open literature, e.g., [1, 2, 3, 4].

In this paper, we consider downlink transmission from a Base Station (BS) to a Mobile Station (MS) over the Nakagami-*m* fading channel. There is a queue at the MAC layer and the AMC at the PHY layer in the BS. We assume that packet retransmission is allowed for the packets received in error. Then, all packets are eventually transmitted once stored in the queue and packet losses can occur due to buffer overflow at the MAC layer. So, the packet overflow probability at the queue can be used to measure the performance of the MAC layer. On the other hand, the performance of the PHY layer can be estimated by the average packet error rate (PER) during transmission due to the AMC. Assuming that the MS requires a certain level of packet overflow probability as the QoS requirement, our cross-layer design objective is to meet the required packet overflow probability while maintaining the average PER at the PHY layer as low as possible.

In this paper, to capture the joint effect of the queueing at the MAC layer and the AMC at the PHY layer we use the effective bandwidth function (EBF) of the packet service process at the queue. For the details of the EBF, refer to [5, 6]. We investigate the behavior of the EBF theoretically when we change the AMC scheme at the PHY layer, and based on our theoretical results we propose a cross-layer design framework, with which we can achieve our objective described above. Note that our previous work [7] considers the same problem, but any theoretical analysis is not provided and only numerical studies are given to see the validity and characteristics of the framework proposed in [7]. Our work in this paper provides the detailed theoretical analysis and the proposed framework is based on the theoretical analysis, which is the main contribution of this paper.

The remainder of this paper is organized as follows: In section 2, we describe our system considered in this paper. In section 3, we analyze the queue at the MAC layer based on the EBF of the packet service process. In section 4, we analyze the behavior of the EBF theoretically when we change the AMC scheme at the PHY layer and provide our cross-layer design framework based on our analysis. In section 5, we give a numerical example to validate our framework and to investigate the characteristics of our framework. In section 6, we give our conclusions.

2 System Modelling

We consider downlink transmission from a Base Station (BS) to a Mobile Station (MS) over a slowly varying fading channel. At the PHY layer, transmissions are performed PHY frame-by-frame, where each PHY frame duration is fixed with length T_f (sec). The PHY frame duration T_f is considered to be unit time in our model, and accordingly we assume that time axis is divided into unit times and time is indexed by t (t = 0, 1, ...). We also assume that the channel condition is slowly varying and remains invariant per PHY frame.

2.1 Wireless channel Model

We assume that the slowly varying fading channel is modelled by the Nakagami-m model where the received SNR (signal-to-noise ratio) γ per frame is a random variable with Gamma probability density function:

$$p_{\gamma}(\gamma) = \frac{m^m \gamma^{m-1}}{\overline{\gamma}^m \Gamma(m)} \exp\left(-\frac{m\gamma}{\overline{\gamma}}\right),\tag{1}$$

where $\overline{\gamma} = E[\gamma]$ is the average received SNR, $\Gamma(m) = \int_0^\infty t^{m-1} \exp(-t) dt$ is the Gamma function, and *m* is the Nakagami fading parameter $(m \ge 1/2)$.

We partition the entire SNR range into M+1 ranges with boundaries $\{\gamma_k\}_{k=0}^{M+1}$ where $\gamma_0 = 0$ and $\gamma_{M+1} = \infty$. We assume that all boundary values $\{\gamma_k\}_{k=0}^{M+1}$ are *fixed*. The range $[\gamma_k, \gamma_{k+1})$ is called *Range* k and denoted by $R_k, 0 \le k \le M$.

To describe the dynamics of the fading channel, we use a Finite State Markov Chain (FSMC) $\{m(t)|t = 0, 1, \dots\}$ with state space $\{0, 1, \dots, M\}$ and when the estimated SNR is in R_k at time t, the Markov chain is defined to be in state k, i.e., m(t) = k.

Let $P = (p_{i,j})$ be the transition probability matrix of the FSMC $\{m(t)\}$ where $p_{i,j}$ denotes the conditional probability that the FSMC $\{m(t)\}$ is in state j at time t + 1, given that it is in state i at time t. To save space, we omit the detailed derivation of the matrix P (for the detailed derivation, see [2]), but note that we allow state transitions from a given state to its two adjacent states only, if any, in the FSMC considered in this paper.

For later use, let π_k ($k \in \{0, 1, \dots, M\}$) denote the stationary probability that the FSMC is in state k, i.e., $\pi_k > 0$, $\sum_{i=0}^{M} \pi_i p_{i,j} = \pi_j$, $0 \le j \le M$ and $\sum_{k=0}^{M} \pi_k = 1$. Then by definition

$$\pi_k = \int_{\gamma_k}^{\gamma_{k+1}} p_{\gamma}(\gamma) d\gamma, \qquad k = 0, \dots, M,$$

where $p_{\gamma}(\gamma)$ is given by (1), and it can be easily shown that, for $0 \le k \le M$

$$\pi_k = \frac{\Gamma(m, m\gamma_k/\overline{\gamma}) - \Gamma(m, m\gamma_{k+1}/\overline{\gamma})}{\Gamma(m)},\tag{2}$$

where $\Gamma(m, x) = \int_x^\infty t^{m-1} \exp(-t) dt$ is the complementary incomplete Gamma function.

2.2 Adaptive Modulation and Coding

The BS employs an Adaptive Modulation and Coding (AMC) scheme with N transmission modes as given in Table 1 where we have N = 5.

We assume that the MS estimates its own SNR every unit time and feeds back the channel state to the BS with no delay through an error-free path. The BS adapts the transmission mode in the AMC every unit time based on the feedback channel state as follows: One of AMC transmission modes in Table 1 is assigned for each range and the AMC transmission mode assigned for Range R_k is denoted by n(k). When the estimated SNR, γ , is in R_k , the BS selects the AMC transmission mode n(k). Later, we will discuss how to select the AMC transmission mode n(k) for Range R_k to achieve our cross layer design objective.

2.3 MAC Layer Model

There is a queue at the MAC layer of the BS, and the service discipline of the queue is first-infirst-out (FIFO). For the service process for packets in the queue at the MAC layer, we assume the following: If a packet is received incorrectly at the MS after error detection, this information is immediately fed back to the BS and the BS retransmits the packet in the next PHY frame. On the other hand, if a packet is received correctly at the MS after error detection, this information is immediately fed back to the BS and the BS retransmits the packet from the next PHY frame.

When transmission mode l is used, we assume that d_l packets in the queue of the MAC layer are mapped into a PHY frame and transmitted simultaneously in the corresponding PHY frame. We assume that $d_1 < d_2 < \cdots < d_N$ and a good example set of $\{d_l\}_{l=0}^N$ can be found in [1]. To avoid deep channel fades, we further assume that no data are sent when the channel

Mode <i>n</i>	1	2	3	4	5
Modulation	BPSK	QPSK	QPSK	16QAM	64QAM
Coding Rate R _e	1/2	1/2	3/4	3/4	3/4
R_n	0.5	1.0	1.5	3.0	4.5

Table 1: The AMC scheme with 5 modes

Table 2: The values of a_n , g_n , and $\tilde{\gamma}_n$ for a packet size of 1080 bits [1]

Mode n	1	2	3	4	5
a_n	274.7229	90.2514	67.6181	53.3987	35.3508
g_n	7.9932	3.4998	1.6883	0.3756	0.0900
$\tilde{\gamma}_n$ (dB)	-1.5331	1.0942	3.9722	10.2488	15.9784

state is in Range R_0 (the worst channel condition), and in this case we assume that a new transmission mode, called transmission mode 0, is used, i.e., n(0) = 0 and $d_0 = 0$ for Range R_0 and transmission mode 0, respectively.

To model the packet service process at the MAC layer, we first consider the packet error process at the PHY layer in our model. The packet error rate (PER) at the PHY layer is expressed as a function of the transmission mode selected by the BS. Let $PER_l(\gamma)$ denote the PER at the PHY layer when transmission mode *l* is used and the received SNR is equal to γ . For transmission modes in Table 1, when the packet length is 1080 bits, Liu *et al.* [1] showed that $PER_l(\gamma)$ can be approximated as

$$\operatorname{PER}_{l}(\gamma) \approx \begin{cases} 1 & (0 < \gamma < \tilde{\gamma}_{l}), \\ a_{l} \exp(-g_{l}\gamma) & (\gamma \ge \tilde{\gamma}_{l}), \end{cases}$$
(3)

where a_l , g_l , and $\tilde{\gamma}_l$ are the mode-dependent parameters and are given in Table 2.

Then, when the received SNR is in Range R_k with transmission mode n(k), the corresponding PER, $r_{k,n(k)}$, is computed as, for $n(k) \ge 1$,

$$r_{k,n(k)} = \frac{1}{\pi_k} \int_{\gamma_k}^{\gamma_{k+1}} a_{n(k)} \exp(-g_{n(k)}\gamma) p_{\gamma}(\gamma) d\gamma$$
(4)

where π_k is the steady state probability that the Markov chain is in state k (or the channel condition is in R_k) and $p_{\gamma}(\gamma)$ is the p.d.f. (probability density function) of the received SNR. Note that π_k and $p_{\gamma}(\gamma)$ are given in (2) and (1), respectively. In practice, we have $0 < r_{k,n(k)} < 1$ for $n(k) \ge 1$. For simplicity, if n(k) = 0, we use $r_{k,n(k)} = r_{k,0} = 1$.

The PER averaged over all transmission modes, called the average PER and denoted by P_{PER} , is then given by

$$P_{PER} = \frac{\sum_{l=0}^{M} \pi_l d_{n(l)} r_{l,n(l)}}{\sum_{l=0}^{M} \pi_l d_{n(l)}}.$$
(5)

Note that, if we use higher transmission mode for R_k , the corresponding PER is higher, i.e., $0 < r_{k,1} < r_{k,2} < \cdots < r_{k,N}$, because the function $a_l \exp(-g_l \gamma)$ is increasing in transmission mode l, i.e., $a_1 \exp(-g_1 \gamma) < a_2 \exp(-g_2 \gamma) < \cdots < a_N \exp(-g_N \gamma)$ and all the others in the right hand side of (4) are fixed.

We now consider the packet service process of the queue at the MAC layer. We assume that packet errors occur independently on a packet-by-packet basis with probability $r_{k,n(k)}$ when m(t) = k at time t.

Let $c_l(t)$ (l = 0, ..., N; t = 0, 1, ...) denote a random variable representing the number of packets successfully transmitted during [t, t + 1) when transmission mode l is selected at time t. Note that, when transmission mode $l(1 \le l \le N)$ is selected at a PHY frame, the number of packets to be transmitted from the queue in the MAC layer is d_l . Then when the estimated SNR γ at time t is in R_k and transmission mode n(k) is selected at time t, $c_{n(k)}(t)$ is according to a Binomial distribution with parameters $d_{n(k)}$ and $1 - r_{k,n(k)}$. The corresponding moment generating function (MGF), denoted by $\phi_{k,n(k)}$, is

$$\phi_{k,n(k)}(\theta) = \left[(1 - r_{k,n(k)})e^{\theta} + r_{k,n(k)} \right]^{d_{n(k)}}.$$
(6)

Then the packet service process is given by $\{c_{n(m(t))}(t)\}$. That is, when the wireless channel state at time *t* is *k*, i.e., m(t) = k, the number of successfully transmitted packets is $c_{n(k)}(t)$. Since $\{m(t)\}$ is a Markov chain, the packet service process $\{c_{n(m(t))}(t)\}$ is a Markov modulated process.

3 Queueing Analysis

In this section, we focus on the queueing process at the MAC layer. Let q(t) (t = 0, 1, ...) denote a random variable representing the queue length (i.e., the number of packets in the queue) at time t. Let a(t) (t = 0, 1, ...) denote a random variable representing the number of packets newly arriving just after time t. Since the packet service process { $c_{n(m(t))}(t)$ } denotes the number of successfully transmitted packets during [t, t + 1), the queueing process {q(t)} evolves according to the following recursion:

$$q(t+1) = \max\{0, q(t) + a(t) - c_{n(m(t))}(t)\}.$$
(7)

To analyze the queueing process, we use the theory of the effective bandwidth. Let C(t) (t = 0, 1, ...) denote a random variable representing the cumulative service process during the interval [0, t), i.e., $C(t) = \sum_{s=0}^{t-1} c_{n(m(s))}(s)$. Let $\Lambda_C(\theta)$ denote the Gärtner-Ellis limit of the cumulative service process C(t), i.e., $\Lambda_C(\theta) = \lim_{t\to\infty} t^{-1} \log \operatorname{Eexp}(\theta C(t))$, provided that the limit exists. Then the Effective Bandwidth Function (EBF) of the packet service process is defined by [6, 8]

$$\xi_C(\theta) = -\frac{\Lambda_C(-\theta)}{\theta}.$$

To compute the EBF of the packet service process, let $\phi(\theta)$ be the diagonal matrix with diagonal elements { $\phi_{0,n(0)}(\theta), \phi_{1,n(1)}(\theta), \dots, \phi_{M,n(M)}(\theta)$ } where $\phi_{k,n(k)}(\theta)$ are given in (6). Since the packet service process { $c_{n(m(t))}(t)$ } is a Markov modulated process, it can be shown that the EBF of the packet service process is given by

$$\xi_C(\theta) = -\frac{\log \delta_C(-\theta)}{\theta},\tag{8}$$

where $\delta_C(\theta)$ is the Perron-Frobenius (PF) eigenvalue of the matrix $C(\theta) = \phi(\theta)P$. For the proof, refer to [5, 6].

Similarly, let A(t) (t = 0, 1, ...) denote a random variable representing the cumulative arrival process during the interval [0, t), i.e., $A(t) = \sum_{n=0}^{t-1} a(n)$. We define the EBF of the arrival process, $\xi_A(\theta)$, by [6]

$$\xi_A(\theta) = \frac{\Lambda_A(\theta)}{\theta},$$

where

$$\Lambda_A(\theta) = \lim_{t \to \infty} t^{-1} \log \operatorname{Eexp}(\theta A(t)).$$

Now we are ready to investigate the queueing performance with the help of the EBFs of the packet service and arrival processes. Let $q(\infty)$ denote a random variable representing the queue length evolved by (7) in steady state. It is known that under some conditions, the packet overflow probability $\mathbb{P}(q(\infty) > x)$ in steady state is approximately given by [6, 8, 9]

$$\mathbb{P}(q(\infty) > x) \approx \mathbb{P}(q(\infty) > 0) \exp(-\theta^* x), \tag{9}$$

where θ^* is the unique real solution of the equation

$$\Lambda_A(\theta) + \Lambda_C(-\theta) = 0,$$

(or equivalently) $\xi_A(\theta) - \xi_C(\theta) = 0.$ (10)

In addition, we have

$$\mathbb{P}(q(\infty) > 0) = \frac{\xi_A(0)}{\xi_C(0)}.$$
(11)

It has been known that the approximation (9) provides a good prediction on the packet overflow probability for a wide range of queueing systems. So, we develop our cross layer design based on the approximation (9) in the next section.

4 Cross-Layer Design Framework and Analysis

In this section, we provide our cross-layer design framework and theoretical analysis of our framework. In our cross-layer design, we consider the performances of the PHY layer and the MAC layer simultaneously. The performance of the PHY layer is estimated by the average PER given in (5), and the performance of the MAC layer is estimated by the packet overflow probability given in (9). Note that there is a tradeoff between the average PER and the packet overflow probability. For lower values of the average PER, lower transmission modes in the AMC are used for ranges, which results in higher packet overflow probability. On the other hand, for higher values of the average PER, higher transmission modes in the AMC can be used, which results in lower packet overflow probability. In [7], this tradeoff is shown through numerical studies.

Now assume that a user requires a certain level of packet overflow probability, i.e., for the reference buffer size *B*, the required packet overflow probability is given by $P_0 = P\{q(\infty) > B\}$. Then, our cross-layer design objective is to guarantee the required packet overflow probability while maintaining the average PER as low as possible.

4.1 The EBF of the packet service process

Even though we can not control the arrival process, the packet service process can be controlled by changing transmission modes n(k) for ranges R_k ($k = 0, 1, \dots, M$). So, our cross-layer design problem is equivalent to assigning suitable transmission modes n(k) for all ranges in such a way that our objective is satisfied. For doing this, we first observe the behavior of the EBF of the packet service process. Note that, if we increase the EBF of the packet service process, then the solution θ^* in (10) increases and the probability $\mathbb{P}(q(\infty) > 0)$ in (11) decreases, and consequently from (9) we see that the packet overflow probability decreases. We will observe this behavior in Fig. 1 and Fig. 2 in section 5. Based on this observation, we consider the following (briefly described) framework in our cross-layer design: An initial set of transmission modes for ranges is given and this initial set of transmission modes does not obviously satisfy our cross-layer design objective. We try to change transmission modes for some ranges in such a way that the EBF of the packet service process increases with the cost of increasing the average PER until the required packet overflow probability is satisfied.

In what follows, we discuss the necessary and sufficient conditions on the transmission mode change under which the resulting EBF of the packet service process increases.

Theorem 4.1. Consider an arbitrary range, say, R_k with transmission mode n(k), and assume that we change the transmission mode of R_k from n(k) to n(k) + 1. The condition that the average number of successfully transmitted packets increases by the transmission mode change, i.e., $(1 - r_{k,n(k)})d_{n(k)} < (1 - r_{k,n(k)+1})d_{n(k)+1}$ is a necessary condition for the increase in the EBF of the packet service process by the transmission mode change.

To prove the theorem, we need the following lemmas and the proof of Theorem 4.1 is given in Appendix. In Lemma 4.2, we consider $\xi_C(\theta)$ in (8) and the relevant matrix $C(\theta) = \phi(\theta)P$.

Lemma 4.2. Consider the transmission mode change of Range R_k from n(k) to n(k)+1. If $\phi_{k,n(k)}(-\theta) > \phi_{k,n(k)+1}(-\theta)$ for $\theta > 0$, the EBF of the packet service process increases for each $\theta > 0$ by the transmission mode change. On the other hand, if $\phi_{k,n(k)}(-\theta) < \phi_{k,n(k)+1}(-\theta)$ for $\theta > 0$, the EBF of the packet service process decreases for each $\theta > 0$ by the transmission mode change.

Proof: See Appendix.

Lemma 4.3. If $(1 - r_{k,n(k)})d_{n(k)} \ge (1 - r_{k,n(k)+1})d_{n(k)+1}$ for $\theta > 0$, we have

$$\phi_{k,n(k)}(-\theta) = [(1 - r_{k,n(k)})e^{-\theta} + r_{k,n(k)}]^{d_{n(k)}}$$

< $\phi_{k,n(k)+1}(-\theta) = [(1 - r_{k,n(k)+1})e^{-\theta} + r_{k,n(k)+1}]^{d_{n(k)+1}}.$

Proof: See Appendix.

Theorem 4.4. Consider an arbitrary range, say, R_k with transmission mode $n(k) \geq 1$, and assume that we change the transmission mode of R_k from n(k) to n(k) + 1. Assume further that $(1 - r_{k,n(k)})d_{n(k)} < (1 - r_{k,n(k)+1})d_{n(k)+1}$. For $\theta > 0$, let

$$f_{k,n(k)}(\theta) = d_{n(k)} \log[(1 - r_{k,n(k)})e^{-\theta} + r_{k,n(k)}] - d_{n(k)+1} \log[(1 - r_{k,n(k)+1})e^{-\theta} + r_{k,n(k)+1}],$$

and

$$A_{k,n(k)} = \frac{d_{n(k)+1}r_{k,n(k)}}{d_{n(k)}(1-r_{k,n(k)})} - \frac{r_{k,n(k)+1}}{1-r_{k,n(k)+1}}.$$

- 1. If $A_{k,n(k)} \ge 0$, then $f_{k,n(k)}(\theta) > 0$ for all $\theta > 0$. Accordingly, the EBF of the packet service process increases by the transmission mode change.
- 2. If $A_{k,n(k)} < 0$ then there exists $\tilde{\theta}$ such that $f_{k,n(k)}(\theta) > 0$ for $0 < \theta < \tilde{\theta}$ and $f_{k,n(k)}(\theta) < 0$ for $\theta > \tilde{\theta}$. Accordingly, the EBF of the packet service process increases for $0 < \theta < \tilde{\theta}$ and the EBF of the packet service process decreases for $\theta > \tilde{\theta}$ by the transmission mode change.

Proof: See Appendix.

Remark 1. For Range R_k with transmission mode 0, i.e., n(k) = 0, when we change the transmission mode from 0 to 1, the necessary condition in Theorem 4.1 is always satisfied because $r_{k,n(k)} = r_{k,0} = 1$ and $d_0 = 0$. Further, we always have $\phi_{k,0}(-\theta) = 1 > \phi_{k,1}(-\theta)$, which implies that the EBF of the packet service process always increases by the transmission mode change from 0 to 1. However, the transmission mode change from 0 to 1 is not always possible. For instance, when Range R_k with $[\gamma_k, \gamma_{k+1})$ is in a very low SNR region such as $\gamma_k < \tilde{\gamma}_1$, the computation of $r_{k,n(k)+1} = r_{k,1}$ from (4) is no longer valid. Therefore, we should check if Range R_k is above $\tilde{\gamma}_1$ in this case. Note that the same situation can occur when we change higher transmission modes, but in practice this does not seem to happen. We discuss it later in Remark 5.

Remark 2. From the proof of Theorem 4.4 in Appendix, we see that there exists θ_0 defined by

$$\theta_0 = \log\left[\frac{d_{n(k)+1}/d_{n(k)} - 1}{-A_{k,n(k)}}\right],$$
(12)

such that $f_{k,n(k)}(\theta) > 0$ for $0 < \theta < \theta_0 < \overline{\theta}$. If the value of θ_0 is large enough and beyond the region of interest, i.e., the decay rate θ^* of the queueing system that satisfies the cross-layer design objective, is obviously less than θ_0 , we can conclude that the EBF of the packet service process always increases in the region of interest by the transmission mode change from n(k) to n(k) + 1.

4.2 The Cross-Layer Design Procedure

We now describe our cross-layer design procedure based on Theorems 4.1 and 4.4. In our cross-layer design, we start with an initial (sufficiently small) average PER value, denoted by r_v . With the value r_v , we determine the initial set of transmission modes $\{n(k)\}$ in such a way that the resulting average PER is below r_v . For details, see below. Then we repeatedly change the transmission modes for ranges in such a way that the EBF of the packet service process increases until either there is no available transmission mode change for ranges (called condition 1) or the initial average PER value is achieved (called condition 2). In the case where condition 2 occurs first, we compute the packet overflow probability, P_t , by (9). If the resulting packet overflow probability is below the required packet overflow probability, i.e., $P_t < P_o$, then we stop the procedure and the resulting set of transmission modes is our solution for the crosslayer design. Otherwise, we increase the initial average PER value r_v by a predefined value $\Delta(> 0)$, i.e., $r'_v = r_v + \Delta$ and repeatedly change the transmission modes for ranges with the new updated value r'_v until either one of two conditions 1 and 2 occurs first. We continue the above procedure until the resulting packet overflow probability is below the required packet overflow probability, i.e., $P_t < P_o$, for the first time. In the case where condition 1 occurs first before the resulting packet overflow probability is below the required packet overflow probability, we conclude that our cross-layer design is an infeasible problem and our system can not guarantee the required packet overflow probability. Note that the infeasibility can arise when the channel condition is severely bad and the required packet overflow probability is very low.

In what follows, we provide the detailed procedure of our cross-layer design.

- 1. We set an initial average PER value r_v which is a sufficiently small value.
- 2. We consider the set of reference boundary values $\{\gamma^{(n)}\}$ defined by

$$\gamma^{(0)} = 0, \qquad \gamma^{(N+1)} = \infty,$$

$$\gamma^{(n)} = \frac{1}{g_n} \log \frac{a_n}{r_v} \quad (n = 1, \dots, N).$$

- 3. For Range R_k with lower boundary γ_k , if $\gamma^{(m)} \leq \gamma_k < \gamma^{(m+1)}$, then transmission mode m is selected for Range R_k i.e., n(k) = m.
- 4. For the set of transmission modes $\{n(k)\}$ compute the indices of boundary ranges, $I(l)(l = 0, 1, \dots, N-1)$ as follows: I(l) = k if n(k) = l and n(k+1) = l+1.
- 5. Among boundary ranges $\{R_{I(l)}|l = 0, 1, 2, \dots, N-1\}$, select boundary ranges for which the transmission mode change is possible in such a way that the EBF of the packet service process increases by the change. To this we check the following conditions:
 - i) $(1 r_{I(l),n(I(l))})d_{n(I(l))} < (1 r_{I(l),n(I(l))+1})d_{n(I(l))+1}$. (refer to Theorem 1.)
 - ii) $\theta_0 = \log \left[\frac{d_{n(I(l))+1}/d_{n(I(l))}-1}{-A_{I(l),n(I(l))}} \right]$ is large enough. (refer to (12).)
 - iii) I(l) > I(l-1) + 1 for l > 0 and $I(0) \ge 1$. (For the reason of this condition, see Remark 5.)
 - iv) $\gamma_{I(l)} > \tilde{\gamma}_{n(I(l))+1}$ (For the reason of this condition, see Remarks 1 and 5.)
 - If there is no boundary range satisfying the above conditions, we set *i* = 0 and go to step 8.
 - Otherwise, let L_1, \dots, L_K denote the indices of boundary ranges satisfying the above conditions. Here, *K* denotes the number of boundary ranges for which transmission mode changes are possible. Let i = 1. Go to step 6.
- 6. We change the transmission mode for R_{L_i} as follows:

$$I(n(L_i)) = I(n(L_i)) - 1, n(L_i) = n(L_i) + 1.$$

- 7. For the resulting set of transmission modes $\{n(k)\}$, compute the average PER, r_t , from (5).
 - If $r_t \leq r_v$ and $1 \leq i < K$, we set i = i + 1 and go to step 6.
 - If $r_t \leq r_v$ and i = K, go to step 5.
 - Otherwise, i.e., if $r_t > r_v$, we set $r_v = r_t$ and go to step 8.
- 8. Compute the packet overflow probability, P_t , from (9) and (11). For the required packet overflow probability P_o ,
 - if $P_t \ge P_o$, we increase the value of r_v by a predefined value $\Delta(>0)$, i.e., $r_v = r_v + \Delta$. For $1 \le i < K$, we set i = i + 1 and go to step 6. For i = K, go to step 5. For i = 0, we stop the procedure and declare the infeasibility of our problem.
 - Otherwise, i.e., if $P_t < P_o$, we stop the procedure.

Remark 3. In step 1, the initial average PER value should be sufficiently small enough to guarantee that the resulting packet overflow probability with the initial set of transmission modes is greater than the required packet overflow probability. Otherwise, step 8 will not work.

Remark 4. The reference boundary values $\{\gamma^{(n)}\}$ in step 2 are the same as in [1]. That is, if transmission mode *n* is used for Range R_k , $\gamma^{(n)}$ is the minimum SNR value such that the corresponding PER, $r_{k,n(k)}$, is less than the initial PER value r_v . Since we use transmission mode n(k) = m for Range R_k with lower boundary value γ_k satisfying $\gamma^{(m)} \leq \gamma_k < \gamma^{(m+1)}$ in step 3, the resulting average PER with the initial set of transmission modes $\{n(k)\}$ is always less than the initial PER value r_v . In addition, it is obvious that n(0) = 0 for Range R_0 .

Remark 5. In step 5, condition ii) is not an exact but a coarse condition for the increase in the EBF of the packet service process by the transmission mode change. Recall that the region of θ for which the EBF of the packet service process increases by the transmission mode change, is $0 < \theta(< \theta_0) < \tilde{\theta}$. The reason why we use condition ii) instead of the exact condition that $\tilde{\theta}$ is large enough, is that we do not have any closed-form formula of $\tilde{\theta}$ and accordingly more numerical computations are needed to get the value of $\tilde{\theta}$. However, we see that condition ii) is satisfied in most numerical studies and considering condition ii) does not limit the applicability of our procedure severely.

Condition iv) is considered because of the validity of the new transmission mode after the change. That is, as seen in (3) transmission modes have their own minimum boundaries $\{\tilde{\gamma}_l\}$, and if the SNR is below the boundary of transmission mode l, $\tilde{\gamma}_l$, the PER is 1 when transmission mode l is used. If a range contains a boundary $\tilde{\gamma}_l$ of transmission mode l, we should not use transmission mode l for the range due to the severe PER.

Condition iii) is considered to guarantee that there exists at least one range for each transmission mode. So, this condition may be removed, but in that case there should be a modification on the update procedure of determining the boundary ranges.

Remark 6. As we increase the value of r_v , our procedure naturally results in the increase of the EBF of the packet service process and accordingly the decrease of the packet overflow probability. Hence, if it is not declared to be infeasible, the final resulting set of transmission modes obtained in step 8 satisfies the objective of our cross-layer design problem. That is, it guarantees the required packet overflow probability while maintaining the average PER as low as possible.

Remark 7. In our cross-layer design procedure, we compute the packet overflow probability when the resulting PER r_t is greater than the updated value of r_v for the first time. However, if we use a sufficiently small value of Δ in the update procedure, we compute the packet overflow probability and the PER whenever we change the transmission mode, i.e., whenever we visit step 6. Note that there is a trade-off between the computational complexity and the value of Δ . That is, the smaller the value of Δ is, the more computational complexity we have in the procedure.

Remark 8. In our cross-layer design procedure, we change the transmission modes of ranges subsequently, i.e., we change $R_{L_1}, R_{L_2}, \dots, R_{L_K}$ in turn. However, there are a number of different orders of changing transmission modes which can increase the EBF of the packet service process. Note that, the decay rate θ^* in (10) of the queueing system depends on the arrival process as well as the packet service process. So, the effect of the increase in the EBF on the packet overflow probability also depends on the arrival process. Accordingly, it is difficult to determine the best order of changing transmission modes which results in the optimal performance over all possible arrival processes in practice. Instead, we consider the above simple but efficient procedure for our cross-layer design in this paper. Even though our framework does not consider the global optimality over all possible cases, our study can provide a benchmark for future advanced cross-layer design frameworks.

5 Numerical Example

In this section, we provide a numerical example where our cross-layer design procedure is applied. For simplicity, we assume that the wireless channel is modelled by a Rayleigh fading model, that is, the Nakagami parameter m is equal to 1. Then, the probability density function $p_{\gamma}(\gamma)$ is given by

$$p_{\gamma}(\gamma) = rac{1}{\overline{\gamma}} \exp\left(-rac{\gamma}{\overline{\gamma}}
ight).$$

We start with how to divide the whole SNR range to construct a FSMC which can well describe the wireless channel. Regarding this issue, there are a number of works in the open



Figure 1: The behavior of the effective bandwidth function

literature, e.g. [10, 11, 12, 13, 14], but in this section we use the method proposed by [12] where the boundaries $\{\gamma_k\}_{k=1}^M$ of ranges satisfy the following equations:

$$c = \frac{\exp\left(-\frac{\gamma_k}{\overline{\gamma}}\right) - \exp\left(-\frac{\gamma_{k+1}}{\overline{\gamma}}\right)}{\sqrt{\frac{2\pi\gamma_k}{\overline{\gamma}}}\exp\left(-\frac{\gamma_k}{\overline{\gamma}}\right) + \sqrt{\frac{2\pi\gamma_{k+1}}{\overline{\gamma}}}\exp\left(-\frac{\gamma_{k+1}}{\overline{\gamma}}\right)}}$$

for $k = 0, 1, 2, \dots, M - 1$. In this study, we use M = 35 and c is approximated equal to 3.003 which is in the region that [12] recommends. Our cross-layer design framework does not depend on the selection of the FSMC which describes the wireless channel, but note that the better the FSMC describes the wireless channel, the more meaningful our result is in practice. The construction of a suitable FSMC for the wireless channel is beyond the scope of this study.

For the other parameters of the Rayleigh fading channel, we use the following parameters in the numerical example.

- the frame length $T_f = 2 \text{ ms}$
- the sequence $\{d_n\}$ of service rates for AMC modes : $d_n = 2n, 0 \le n \le N$
- the average SNR $\bar{\gamma} = 10 \text{ dB}$
- the Doppler frequency $f_d = 10$ Hz

For the arrival process, we consider an ON and OFF process with the following parameters: The transition probability from the ON state (resp. OFF state) to the OFF state (resp. ON state) is 0.35 (resp. 0.55), and the number of packets arriving in unit time with ON state (resp. OFF state) is 5 (resp. 0). We assume that the reference buffer size is 1000 and the required packet overflow probability is 10^{-3} .

Fig. 1 shows how the EBF of the packet service process is updated in our cross-layer design procedure. In the figure, we see that the EBF of the packet service process increases and that the EBF of the packet service process for the final assignment is greater than that for the initial assignment.

Table 3 shows the initial and final assignments of transmission modes for ranges. During the procedure we see the following observations:

Mode	Initial	Final
	Assignment	Assignment
0	Regions 0 to 2	Regions 0 to 1
1	Region 3	Region 2
2	Region 4	Region 3
3	Regions 5 to 10	Regions 4 to 6
4	Regions 11 to 20	Regions 7 to 18
5	Regions 21 to 35	Regions 19 to 35

Table 3: The transmission mode assignment

- As seen in Fig. 1 our system with the AMC scheme based on the initial set of transmission modes is an unstable system because there is no solution of (10) in this case. However, for the other cases in Fig. 1 our system is stable.
- As we change the transmission modes of ranges, the effect on the increase in the EBF of the packet service process becomes less significant. For instance, there is a big difference between the first two EBFs (of the initial assignment and case 1) in Fig. 1, but for the other EBFs there is a relatively small difference between two consecutive EBFs. However, if we consider the region of *θ* near 0, we still have meaningful differences among EBFs.
- When we change the transmission mode at least greater than or equal to 3 to higher, the increase in the EBF of the packet service process is significant. On the other hand, the change of transmission mode less than 3 results in a small change in the EBF of the packet service process.
- The violation of the conditions i) and ii) in step 5 does not happen in our numerical example, which means that the EBF increases by the transmission mode change in the regions of interest.
- We can not change the initial transmission mode of Range R_1 because the condition of iv) in step 5 is not satisfied. That is, since the SNR in Range R_1 is very low, we can not apply transmission mode 1 for Range R_1 .

Obviously, the above observations are based on our numerical example and should not be generalized to other examples, but we think most of them are still true for other examples and accordingly should bear in mind.

Next, to see the performance improvement with respect to the packet overflow probability we give Fig. 2 which shows the corresponding packet overflow probabilities (9) to the EBFs (of case 1, case 2 and the final assignment) shown in Fig. 1. Note that we can not get the packet overflow probability for our system with the AMC scheme based on the initial assignment. As seen in Fig. 2, as the EBF is increasing, the packet overflow probability is decreasing. In addition, we see that the AMC with our final assignment satisfies the required packet overflow probability 10^{-3} (or $\log(10^{-3}) = -6.908$). The corresponding packet overflow probabilities are changed from 0.00614 to 0.000657 and the corresponding average PERs are changed from 0.00062 to 0.0222. From Fig. 1 and Fig. 2, we can check the validity of our cross-layer design procedure and the characteristics of our cross-layer design framework.



Figure 2: The packet overflow probabilities

6 Conclusions

In this paper, we provide a cross-layer design framework to support QoS for wireless communication services. In our framework, we consider the joint effect of the queueing at the MAC layer and the AMC at the PHY layer and provide a procedure of selecting a suitable AMC scheme, with which we can achieve our cross-layer design objective. The effective bandwidth theory plays an important role in our study. The main contribution of this paper is that we provide a theoretical analysis on the behavior of the effective bandwidth function of the packet service process at the MAC layer and the proposed framework is based on our theoretical analysis. A numerical example is provided to see the validity and characteristics of the proposed framework.

Appendix

A.1. The proof of Lemma 4.2

Proof: When we change the transmission mode from n(k) to n(k) + 1, the element $\phi_{k,n(k)}(-\theta)$ is replaced by $\phi_{k,n(k)+1}(-\theta)$ in the matrix $C(-\theta)$. When $\phi_{k,n(k)}(-\theta) > \phi_{k,n(k)+1}(-\theta)$ for $\theta > 0$, then the resulting Perron-Frobenius eigenvalue $\delta_C(-\theta)$ of the matrix $C(-\theta)$ decreases for each $\theta > 0$ because $C(-\theta)$ is a nonnegative and irreducible matrix [15, 16]. Since $-\log x$ is a decreasing function in x, the resulting EBF of the packet service process, i.e., $-\log \delta_c(-\theta)/\theta$ increases for each $\theta > 0$. Similarly, we can also prove the remaining part of our lemma.

A.2. The proof of Lemma 4.3

Proof: For simplicity, we use r_n , d_n , r_{n+1} and d_{n+1} instead of $r_{k,n(k)}$, $d_{n(k)}$, $r_{k,n(k)+1}$ and $d_{n(k)+1}$, respectively. Let

$$f_1(\theta) = [(1 - r_n)e^{-\theta} + r_n]^{d_n}, f_2(\theta) = [(1 - r_{n+1})e^{-\theta} + r_{n+1}]^{d_{n+1}}.$$

It suffices to show that $f_1(\theta) < f_2(\theta)$ for $\theta > 0$. Observe that

$$\log f_1(\theta) - \log f_2(\theta) = d_n \log[(1 - r_n)e^{-\theta} + r_n] - d_{n+1} \log[(1 - r_{n+1})e^{-\theta} + r_{n+1}].$$

From the fact that $(1 - r_n)d_n \ge (1 - r_{n+1})d_{n+1}$, we get

$$d_n \geq \frac{1-r_{n+1}}{1-r_n}d_{n+1}$$

and

$$d_n \log[(1 - r_n)e^{-\theta} + r_n] \le \frac{1 - r_{n+1}}{1 - r_n} d_{n+1} \log[(1 - r_n)e^{-\theta} + r_n]$$

for $\log[(1 - r_n)e^{-\theta} + r_n] < 0$. Note here that n(= n(k)) is not equal to 0 due to the condition $d_n(1 - r_n) \ge (1 - r_{n+1})d_{n+1}$. Hence,

$$\log f_1(\theta) - \log f_2(\theta) \le \frac{1 - r_{n+1}}{1 - r_n} d_{n+1} \log[(1 - r_n)e^{-\theta} + r_n] - d_{n+1} \log[(1 - r_{n+1})e^{-\theta} + r_{n+1}].$$

Multiplying $\frac{1-r_n}{d_{n+1}}(>0)$ on both sides, we get

$$\frac{1-r_n}{d_{n+1}} [\log f_1(\theta) - \log f_2(\theta)] \\\leq (1-r_{n+1}) \log[(1-r_n)e^{-\theta} + r_n] - (1-r_n) \log[(1-r_{n+1})e^{-\theta} + r_{n+1}].$$

If we can show that the right hand side of the above inequality is negative for $\theta > 0$, the proof is completed. To do this, let $x = 1 - r_n$ and $y = 1 - r_{n+1}$. Then the right hand side is

$$y\log[xe^{-\theta}+1-x]-x\log[ye^{-\theta}+1-y]$$

and 0 < y < x < 1. We will show that

$$\frac{1}{x}\log[xe^{-\theta}+1-x] < \frac{1}{y}\log[ye^{-\theta}+1-y].$$

For 0 < t < 1 and $\theta > 0$, define $g(t, \theta)$ by

$$g(t,\theta) = \frac{1}{t} \log[te^{-\theta} + 1 - t].$$

Observe that

$$\begin{aligned} \frac{\partial}{\partial t}g(t,\theta) &= \frac{1}{t^2} \left\{ \frac{e^{-\theta} - 1}{te^{-\theta} + 1 - t} t - \log[te^{-\theta} + 1 - t] \right\} \\ &= \frac{1}{t^2} \left\{ 1 - \frac{1}{te^{-\theta} + 1 - t} - \log[te^{-\theta} + 1 - t] \right\} \end{aligned}$$

Using the fact that $h(a) = 1 - 1/a - \log(a) < 0$ for 0 < a < 1, we get $\frac{\partial}{\partial t}g(t,\theta) < 0$ for 0 < t < 1. This yields $g(t,\theta)$ is strictly decreasing in *t* for each $\theta > 0$. Since 0 < y < x < 1, we get $g(x,\theta) < g(y,\theta)$ and we complete the proof.

A.3. The proof of Theorem 4.1

Proof: If $(1 - r_{k,n(k)})d_{n(k)} \ge (1 - r_{k,n(k)+1})d_{n(k)+1}$ for $\theta > 0$, we have $\phi_{k,n(k)}(-\theta) < \phi_{k,n(k)+1}(-\theta)$ by Lemma 4.3. Then, by Lemma 4.2 the EBF of the packet service process decreases by the transmission mode change. This completes the proof.

A.4. The proof of Theorem 4.4

Proof: For simplicity, we use r_n , d_n , r_{n+1} , d_{n+1} and $f_n(\theta)$ instead of $r_{k,n(k)}$, $d_{n(k)}$, $r_{k,n(k)+1}$, $d_{n(k)+1}$ and $f_{k,n(k)}(\theta)$, respectively, in the proof. First observe that $f_n(0) = 0$ and

$$f'_{n}(\theta) = \frac{-d_{n}(1-r_{n})e^{-\theta}}{(1-r_{n})e^{-\theta}+r_{n}} + \frac{d_{n+1}(1-r_{n+1})e^{-\theta}}{(1-r_{n+1})e^{-\theta}+r_{n+1}}.$$

Then using $x = 1 - r_n$ and $y = 1 - r_{n+1}$, we have, for $\theta > 0$

$$\begin{split} f_n'(\theta) > 0 \quad & \text{iff} \quad \frac{d_{n+1}(1-r_{n+1})}{(1-r_{n+1})e^{-\theta}+r_{n+1}} > \frac{d_n(1-r_n)}{(1-r_n)e^{-\theta}+r_n} \\ & \text{iff} \quad \frac{d_{n+1}}{d_n} \frac{(1-r_{n+1})}{(1-r_{n+1})e^{-\theta}+r_{n+1}} > \frac{(1-r_n)}{(1-r_n)e^{-\theta}+r_n} \\ & \text{iff} \quad \frac{d_{n+1}}{d_n} \frac{y}{ye^{-\theta}+1-y} > \frac{x}{xe^{-\theta}+1-x} \\ & \text{iff} \quad \frac{d_{n+1}}{d_n} \frac{xe^{-\theta}+1-x}{x} > \frac{ye^{-\theta}+1-y}{y} \\ & \text{iff} \quad \frac{d_{n+1}}{d_n}(e^{-\theta}+\frac{1-x}{x}) > e^{-\theta}+\frac{1-y}{y} \\ & \text{iff} \quad (\frac{d_{n+1}}{d_n}-1)e^{-\theta}+\frac{d_{n+1}}{d_n}\frac{1-x}{x}-\frac{1-y}{y} > 0 \\ & \text{iff} \quad (\frac{d_{n+1}}{d_n}-1)e^{-\theta}+A_n > 0. \end{split}$$

Now consider the case where $A_n \ge 0$. In this case, we have $f'_n(\theta) > 0$ for all $\theta \ge 0$. Since $f_n(0) = 0$, we see that $f_n(\theta) > 0$ for all $\theta > 0$.

Consider the case where $A_n < 0$. Note that $f'_n(0) = -d_n(1 - r_n) + d_{n+1}(1 - r_{n+1}) > 0$. Define $h(t) = (\frac{d_{n+1}}{d_n} - 1)t + A_n$ for $0 < t \le 1$ by letting $t = e^{-\theta}$ for $\theta \ge 0$. Then we have $h(1) = \frac{f'_n(0)}{d_n xy} > 0$, $h(0) = A_n < 0$. Since h(t) is a linear function in t, there exists a unique solution $t_0 = \frac{-A_n}{d_{n+1}/d_n - 1}$ satisfying $h(t_0) = 0$. Then for $\theta_0 = -\log(t_0)(>0)$ it follows that $f'_n(\theta_0) = 0$, $f'_n(\theta) > 0$ for $0 < \theta < \theta_0$ and $f'_n(\theta) < 0$ for $\theta > \theta_0$. Hence, $f_n(\theta)$ is strictly increasing for $0 < \theta < \theta_0$ and strictly decreasing for $\theta > \theta_0$. Since $f_n(0) = 0$, there exists a unique solution $f_n(\tilde{\theta}) = 0$ with $\tilde{\theta} > \theta_0$ and consequently we have $f_n(\theta) > 0$ for $0 < \theta < \tilde{\theta}$ and $f_n(\theta) < 0$ for $\theta > \tilde{\theta}$. Then by using Lemma 4.2, we complete the proof.

References

- Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. on Wireless Communications*, vol. 3, no.5, pp. 1746–1755, September 2004.
- [2] Q. Liu, S. Zhou, and G. B. Giannakis, "Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design," *IEEE Trans. on Wireless Communications*, vol. 4, no. 3, pp. 1142–1153, May 2005.
- [3] J. Razavilar, K. J. R. Liu, and S. I. Marcus, "Jointly optimized bit-rate/delay control policy for wireless packet networks with fading channels," *IEEE Trans. on Communications*, vol. 50, no. 3, pp. 484–494, March 2002.

- [4] V. Srivastava and M. Motani, "Cross-Layer Design: A Survey and the Road Ahead," IEEE Communications Magazine, vol. 43, no. 12, pp. 112–119, December 2005.
- [5] C.-S. Chang and J. A. Thomas, "Effective bandwidths in high-speed digital networks," IEEE J. Selected Areas in Communications, vol. 3, no. 6, pp. 1091–1100, August 1995.
- [6] C.-S. Chang, Performance guarantees in communication networks, Springer-Verlag, 2000.
- [7] F. Ishizaki and G. U. Hwang, "Cross-layer Design and Analysis of Wireless Networks Using the Effective Bandwidth Function," to appear in IEEE Trans. on Wireless Communications, 2007.
- [8] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Trans. on Wireless Communications*, vol. 2, no. 4, pp. 630-643, July 2003.
- [9] B. L. Mark, G. Ramamurthy, "Real-Time Estimation and Dynamic Renegotiation of UPC Parameters for Arbitrary Traffic Sources in ATM Networks," *IEEE/ACM Trans. on Networking*, vol. 6, no. 6, pp. 811–827, December 1998.
- [10] M. Hassan, M. M. Krunz and I. Matta, "Markov-based channel characterization for tractable performance analysis in wireless packet networks," *IEEE Trans. on Wireless Communications*, vol. 3, no. 3, pp. 821–831, May 2004.
- [11] H. S. Wang and N. Moayeri, "Finite-State Markov Channel A Useful Model for Radio Communication Channels," *IEEE Trans. on Vehicular Technology*, vol. 4, no. 1, pp. 163–171, 1995.
- [12] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," IEEE Trans. on Communications, vol.47, no.11, pp.1688–1692, 1999.
- [13] C. C. Tan and N. C. Beaulieu, "On First-Order Markov Modeling for the Rayleigh Fading Channel," *IEEE Trans. on Communications*, vol. 48, no. 12, pp. 2032–2040, 2000.
- [14] J. Arauz, P. Krishnamurthy and M. A. Labrador, "Discrete Rayleigh Fading Channel Modeling," Wireless Communications and Mobile Computing, vol. 4, pp. 413–425, 2004.
- [15] E. Seneta, Non-Negative Matrices: An Introduction to Theory and Applications, John Wiley & Sons, Inc., 1973.
- [16] H. Minc, Nonnegative Matrices, John Wiley & Sons, Inc., 1988.

Performance Optimization in Bandwidth-Sharing Networks

Maaike Verloop^a, Sem Borst^{a,b,c} and Rudesindo Núñez-Queija^{a,d} ^aCWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands ^b Dept. Math. & Comp. Sc., Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands ^cAlcatel-Lucent, Bell Laboratories, P.O. Box 636, Murray Hill, NJ 07974, USA ^dTNO ICT, P.O. Box 5050, 2600 GB Delft, The Netherlands

This research is financially supported by The Netherlands Organization for Scientific Research (NWO) and the Dutch Bsik/BRICKS project.

Abstract

Bandwidth-sharing networks as considered by Massoulié & Roberts provide a natural modeling framework for describing the dynamic flow-level interaction among elastic data transfers. Although valuable stability results have been obtained, crucial performance metrics such as flow-level delays and throughputs in these models have remained intractable in all but a few special cases. In particular, it is not well understood to what extent flow-level delays and throughputs achieved by standard bandwidth-sharing mechanisms such as α -fair strategies leave potential room for improvement.

In order to gain a better understanding of the latter issue, we set out to determine the scheduling policies that minimize the mean delay in some simple linear bandwidth-sharing networks. While admittedly simple, linear networks provide a useful model for flows that traverse several links and experience bandwidth contention from independent cross-traffic. We compare the performance of the optimal policy with that of various α -fair strategies so as to assess the efficacy of the latter and gauge the potential room for improvement. The results indicate that the optimal policy achieves only modest improvements, even when the value of α is simply fixed, provided it is not too small. This suggests that (optimization within) the family of α -fair strategies is likely to be adequate for most practical purposes.

1 Introduction

Over the past several years, the processor-sharing discipline has emerged as a useful paradigm for evaluating the flow-level performance of elastic data transfers competing for bandwidth on a single bottle-neck link, see for instance [2, 14]. Bandwidth-sharing networks as considered by Massoulié & Roberts [12] provide a natural extension for modeling the dynamic interaction among competing elastic flows that traverse several links along their source-destination paths. Bonald & Massoulié [3] showed that a wide class of α -fair bandwidth-sharing policies as introduced by Mo & Walrand [13] achieve stability in such networks under the simple (and necessary) condition that no individual link is overloaded, see also [19] for instance. While stability is arguably the most fundamental performance criterion, flow-level delays and throughputs are obviously crucial metrics too. Although useful approximations, bounds [4] and heavytraffic limits [10] have been obtained, the latter performance metrics have largely remained intractable in all but a few special cases. In particular, it is not well understood to what extent the flow-level delays and throughputs achieved by common bandwidth-sharing mechanisms leave potential room for improvement.

The scope for improving flow-level delays and throughputs has been the focus of intense efforts in a somewhat distinct strand of research on size-based scheduling strategies. The rationale for size-based scheduling has been greatly amplified by empirical findings indicating that file sizes in the Internet show huge variability and commonly have infinite variance [7]. Several studies have demonstrated that the Shortest Remaining Processing Time first (SRPT) discipline can achieve significant performance improvements for heavy-tailed service requirements compared to First-Come First-Served or Processor Sharing. The SRPT discipline has therefore been adopted as an effective mechanism for improving the performance of web servers [5, 9]. To some extent, the huge variability in flow sizes also alleviates the long-standing concerns that have surrounded SRPT regarding the perceived unfairness towards extremely large jobs [1, 8, 22]. It turns out that in case of heavy-tailed distributions only an exceedingly small fraction of the jobs is worse off than under Processor Sharing as the prototype of perfect fairness. A critical issue associated with size-based scheduling in general and SRPT in particular, is that it relies on (partial) knowledge of (remaining) service requirements. While such information is usually available in web servers, it is impractical to obtain in Internet routers. An alternative strategy which has hence been advocated for scheduling data flows is the Least Attained Service first (LAS) discipline also known as Foreground-Background Processor Sharing [11, 15]. In case the service requirement distribution has a decreasing failure rate, it has been shown that LAS stochastically minimizes the number of jobs in the system among all strategies that use no knowledge of the remaining job sizes [16].

Nearly all studies on the performance gains from size-based scheduling strategies such as SRPT and LAS have considered single-server settings. Single-server systems provide reasonable models for web servers, but they do not accurately capture scenarios where users require service from several resources simultaneously. Such concurrent resource possession arises in the above-mentioned bandwidth-sharing networks, where data flows traverse several links between their source-destination pairs and consume bandwidth on each of them for the duration of the transfer. (Even though individual packets travel across the network on a hop-by-hop basis, on a somewhat longer time scale a data flow claims roughly equal bandwidth on each of the links along its path since the amount of buffering at intermediate nodes is typically quite limited.)

While single-server systems provide tractable results and useful insights, they do not exhibit the potential non-work-conserving behavior that may occur in scenarios with concurrent resource possession. There are various indications that priority mechanisms in such scenarios may cause starvation effects with possibly severe consequences. For example, Yang & de Veciana [23, 24] demonstrated that SRPT scheduling in network scenarios may yield considerable performance improvements in terms of mean delays and throughputs, but also observed that flows on long routes with large sizes may sustain a marked performance degradation. Recently, it was shown that size-based scheduling strategies such as SRPT and LAS may in fact unnecessarily fail to achieve stability in network settings, even at arbitrarily low loads [21].

In conclusion, the results for size-based scheduling in single-server models do not provide a good indication for the scope for improvement over common bandwidth-sharing mechanisms in network scenarios. In order to gain better insight into the latter issue, we will set out to determine scheduling policies that minimize the mean delay in bandwidth-sharing networks with a linear topology. While admittedly simple, linear networks provide a useful model for flows that traverse several links and experience bandwidth contention from independent cross-traffic. Armed with the knowledge of the optimal policy, we then compare its performance with various α -fair strategies so as to assess the efficacy of the latter and gauge the potential room for improvement. Our results indicate that the optimal policy achieves only modest improvements over an α -fair strategy when the value of α is optimized. In its turn, an optimized α -fair strategy yields only marginal improvements compared to virtually any fixed value of α , as long as this value is not too small. This is particularly so for the important special cases $\alpha = 1$ (proportional fair strategy) and $\alpha = 2$ (which is a modeling abstraction of TCP). In fact, virtually any α -fair strategy shows fairly robust performance over a wide range of traffic parameters, as long as the value of α is not too small. This suggests that (optimization within) the family of α -fair strategies is likely to be adequate for most practical purposes.

The remainder of the paper is organized as follows. In Section 2 we provide a detailed model description and discuss some preliminaries. In Section 3 we derive some sample-path comparisons for the workload processes under various scheduling policies. We use these sample-path inequalities in Section 4 to show that in certain cases with exponential service requirements relatively simple priority-type policies minimize the mean number of users in the system. In Section 5 we examine cases where the optimal policy does not have a simple priority-type structure, and use dynamic programming techniques to prove that in these cases the optimal policy is characterized by a switching curve. Section 6 presents the numerical experiments that we conducted. We summarize our results in Section 7.

2 Model description and preliminary results

We consider a linear network with L nodes. For convenience, we assume each of the nodes to have a unit service rate. In order to present the results in the simplest possible setting, we focus on a traffic scenario with L + 1 classes, where class i requires service at node i only, i = 1, ..., L, while class 0 requires service at all L nodes simultaneously. The above 'toy' scenario appears already sufficiently rich to exhibit many of the qualitative phenomena that may occur for general network topologies and route structures. Class-i users arrive according to independent Poisson processes of rate λ_i , and have generally distributed service requirements B_i with mean β_i , i = 0, ..., L.

Define the traffic load of class i as $\rho_i := \lambda_i \beta_i$. Thus the load at node i is $\rho_0 + \rho_i$, i = 1, ..., L. The obviously necessary conditions for stability $\rho_0 + \rho_i < 1$, for i = 1, ..., L, are known [3] to be sufficient as well for α -fair bandwidth-sharing policies. (For conciseness, these conditions will be referred to as the 'standard' conditions.) In order to examine the effectiveness of α -fair policies we seek policies that in some appropriate sense minimize the total number of active users in the above-described system. We only allow (possibly preemptive) policies that have no knowledge available of the remaining service requirements and denote this class of policies by II. The following policies will play a central role.

- Policy π* gives preemptive priority to class 0 whenever it is non-empty and, otherwise, serves any other class with at least one user.
- Policy π^{**} simultaneously serves all classes i = 1, ..., L whenever at least one user of each class is present. Otherwise class 0 is served. When class 0 is empty, any other class with at least one user present is served.

For both these policies the system is stable under the standard conditions, since policies π^* and π^{**} ensure that each node operates at full rate when it is non-empty.

For a given policy π , denote by $N_i^{\pi}(t)$ the number of class-*i* users at time *t* and by $W_i^{\pi}(t)$ their total residual work. $N^{\pi}(t)$ is defined as $\sum_{i=1}^{L} N_i^{\pi}(t)$. We further define N_i^{π} , W_i^{π} and N^{π}
as random variables with the corresponding time-average distributions (when they exist). For brevity, we use the superscripts * and ** for random variables corresponding to π^* and π^{**} .

Note that class 0 does not notice the presence of other classes under policy π^* . The mean amount of class-0 work is therefore given by the Pollaczek-Khintchine formula:

$$\mathbb{E}(W_0^*) = \frac{\lambda_0 \mathbb{E}(B_0^2)}{2(1-\rho_0)}.$$

With policy π^* , any class $i \neq 0$ sees its service being interrupted by busy periods of class 0 so that [18]:

$$\mathbb{E}(W_i^*) = \frac{\lambda_0 \mathbb{E}(B_0^2) + \lambda_i \mathbb{E}(B_i^2)}{2(1 - \rho_0 - \rho_i)} - \frac{\lambda_0 \mathbb{E}(B_0^2)}{2(1 - \rho_0)}.$$

Note that these formulas hold for any service requirement distribution and scheduling discipline within classes.

In the special case of exponentially distributed service requirements, scheduling within a class (without knowledge of the actual size of jobs) does not affect the distribution of the number of users. Letting $\mu_i = 1/\beta_i$ (and thus $\mathbb{E}(B_i^2) = 2/\mu_i^2$)), the mean number of users can then simply be obtained from $\mathbb{E}(N_i^*) = \mu_i \mathbb{E}(W_i^*)$ for all classes *i*. In particular

$$\mathbb{E}(N_0^*) = \frac{\rho_0}{1-\rho_0}$$

and

$$\mathbb{E}(N_i^*) = \frac{\rho_i}{1 - \rho_0 - \rho_i} + \frac{\mu_i}{\mu_0} \left(\frac{\rho_0}{1 - \rho_0 - \rho_i} - \frac{\rho_0}{1 - \rho_0} \right),$$

so that

$$\mathbb{E}(N^*) = \frac{\rho_0}{1-\rho_0} + \sum_{i=1}^{L} \Big(\frac{\frac{\lambda_i}{\mu_0}\rho_0 + \rho_i^2}{(1-\rho_0)(1-\rho_0-\rho_i)} + \frac{\rho_i}{1-\rho_0} \Big).$$
(1)

For policy π^{**} there is no closed-form expression available for the mean workloads. For L = 2, determining these is equivalent to solving a boundary-value problem [6]: the service rate allocated to any class *i* depends on the workloads of all other classes.

3 Workload

In this section we allow for general service requirement distributions and compare (samplepath wise) the workloads of the various classes under different policies.

Let $\bar{\pi}^i$ be a policy that is work-conserving in node *i*, i.e., the capacity of node *i* is fully used whenever that node is non-empty. Obviously, such a policy minimizes the total workload in node *i* at all times. More specifically, if $W_0^{\bar{\pi}^i}(0) + W_i^{\bar{\pi}^i}(0) \leq_{st} W_0^{\pi}(0) + W_i^{\pi}(0)$ for some arbitrary policy π , then

$$W_0^{\bar{\pi}^i}(t) + W_i^{\bar{\pi}^i}(t) \le_{st} W_0^{\pi}(t) + W_i^{\pi}(t), \ \forall t \ge 0.$$
(2)

Here \leq_{st} denotes the usual stochastic ordering. Note that both policies π^* and π^{**} are workconserving in each node, so inequality (2) holds for all i = 1, ..., L, if $\bar{\pi}^i \in \{\pi^*, \pi^{**}\}$. We call $W_{0,j,k}^{\pi}(t) := W_0^{\pi}(t) + W_j^{\pi}(t) + W_k^{\pi}(t)$ the aggregate workload in nodes j and k. Besides minimizing the workload in one node, at any point in time, policy π^{**} also minimizes the aggregate workload in at least one pair of nodes (these need not always be the same) as is formalized in the following lemma. This result will be useful for the analysis in the next sections. **Lemma 3.1.** If for t = 0 there exist nodes j and k with $j \neq k$, such that

$$W_{0,j,k}^{**}(t) \leq_{st} W_{0,j,k}^{\pi}(t),$$
(3)

then, for any t > 0, there exist j and k (not necessarily the same as at time t = 0) with $j \neq k$ such that (3) holds.

Hence, if L = 2, the lemma states that policy π^{**} stochastically minimizes the total workload in the system. We note that there is no policy that achieves the same for L > 2.

Proof of Lemma 3.1 By assuming the same sequence of arrivals and service requests, we can compare the two policies, π^{**} and π , in the same sample space. Let

$$u = \inf\{t > 0: W^{**}_{0,j,k}(t) > W^{\pi}_{0,j,k}(t), \forall j, k \neq 0, j \neq k\}.$$

We show by contradiction that u cannot be finite. Let us suppose $u < \infty$. Inequality (3) can only be violated for all pairs j and k immediately after time u, if it holds with equality at time u for some j and k, which we fix for the remainder of the proof. In addition, for the equality to cease to be valid, policy π^{**} should not be serving both nodes j and k at full rate, so that $W_0^{**}(u) = 0$ and $W_i^{**}(u) = 0$ for either i = j or i = k. From (2) we have $W_l^{**}(u) \le W_0^{\pi}(u) + W_l^{\pi}(u)$ for all $l \neq 0, l \neq i$; we fix such an l and observe that this inequality is preserved until the next arrival from either class 0 or class i (in the mean time, π^{**} works at full rate in node l and π can not do better than that). Note that $W_0^{**}(t) = 0$ and $W_i^{**}(t) = 0$ until such an arrival occurs and, hence, $W_{0,i,l}^{**}(t) \le W_{0,i,l}^{\pi}(t)$, which contradicts the definition of u.

4 Small class-0 users

In the remainder of the paper we focus on exponentially distributed service requirements and write $\mu_i = 1/\beta_i$. For relatively 'large' values of μ_0 , i.e., when class-0 users are relatively small, we show that either π^* or π^{**} stochastically minimizes the number of users at every point in time. More precisely: this is so when $\mu_0 > \sum_{i \ge 1, i \ne j} \mu_i$ for all $j \ne 0$. In Section 4.1 we first show that the results of the previous section allow us to readily prove that π^* and π^{**} minimize the *mean* number of users in the above-mentioned cases. Because of Little's law, such a policy automatically minimizes the mean overall sojourn time as well. We briefly comment on the stochastic optimality in Section 4.2.

To put our results in context, we recall that the μ -rule is known to stochastically minimize the number of users [17] in a single-server system. The rationale behind this rule is that it maximizes the output rate at all times. In the network we discuss, this can only be accomplished for certain parameter values. Besides trying to maximize the total output rate of the system, we must take into account that when serving class $i \neq 0$ while another class $j \neq 0$ is empty may leave node j underutilized if there are users of class 0. For example, if $\mu_i > \mu_0$ for all $i = 1, \ldots, L$, then giving priority to classes $1, \ldots, L$, myopically maximizes the total output rate of the system but such a discipline unnecessarily causes instability [21] when $\prod_{j=1}^{L}(1-\rho_j) < \rho_0$. In general, there can be a trade-off between maximizing the output rate and using the full capacity in each node whenever that node is non-empty. It is precisely in those cases where these two objectives are compatible, that we can identify the policies that minimize the total number of users.

4.1 Minimizing the mean number of users

The next lemma, together with the results for the workload obtained in Section 3, can be used to prove that, in certain cases, policy π^* or π^{**} minimizes the *mean* total number of users at every point in time.

Lemma 4.1. Let π , $\bar{\pi} \in \Pi$ and assume the service requirements of class *i* are exponentially distributed with mean $\beta_i = 1/\mu_i$. If for some $I \subseteq \{0, \ldots, L\}$, $\sum_{i \in I} W_i^{\bar{\pi}}(t) \leq_{st} \sum_{i \in I} W_i^{\pi}(t)$, $\forall t \ge 0$, then

$$\sum_{i\in I} \frac{1}{\mu_i} \mathbb{E}(N_i^{\bar{\pi}}(t)) \leq \sum_{i\in I} \frac{1}{\mu_i} \mathbb{E}(N_i^{\pi}(t)).$$

Proof Because of the memoryless property of the exponential distribution and the fact that policies π and $\bar{\pi}$ have no knowledge of the remaining service requirements, the workload, $W_i(t)$, is distributed as $\sum_{k=1}^{N_i(t)} E_k^i$. Here E_k^i are i.i.d. random variables from an exponential distribution with mean $1/\mu_i$. Hence,

$$\sum_{i \in I} \sum_{k=1}^{N_i^{\#}(t)} E_k^i \leq_{st} \sum_{i \in I} \sum_{k=1}^{N_i^{\#}(t)} E_k^i, \quad \forall t \ge 0,$$
(4)

and the lemma is proved after taking expectations.

This lemma paves the way for the following two propositions, which state that, in certain cases, π^* or π^{**} is optimal.

Proposition 4.2. Assume $W_i^*(0) \leq_{st} W_i^{\pi}(0)$, for all *i*. If $\sum_{i=1}^{L} \mu_i \leq \mu_0$, then $\mathbb{E}(N^*(t)) \leq \mathbb{E}(N^{\pi}(t))$, $\forall \pi \in \Pi$ and $\forall t \geq 0$.

Proof By (2), policy π^* minimizes the workload in each node, which implies by Lemma 4.1 that $\forall i = 1, ..., L$,

$$\frac{1}{\mu_0}\mathbb{E}(N_0^*(t)) + \frac{1}{\mu_i}\mathbb{E}(N_i^*(t)) \le \frac{1}{\mu_0}\mathbb{E}(N_0^{\pi}(t)) + \frac{1}{\mu_i}\mathbb{E}(N_i^{\pi}(t)).$$
(5)

Combining Lemma 4.1 with the fact that giving preemptive priority to class *i* minimizes the workload of class *i*, we have:

$$\mathbb{E}(N_0^*(t)) \le \mathbb{E}(N_0^{\pi}(t)). \tag{6}$$

Multiplying (5) by $\mu_i \ge 0$, for all i = 1, ..., L, multiplying (6) by $\frac{\mu_0 - \sum_{i=1}^L \mu_i}{\mu_0} \ge 0$ and summing these L + 1 inequalities gives $\sum_{i=0}^L \mathbb{E}(N_i^*(t)) \le \sum_{i=0}^L \mathbb{E}(N_i^{\pi}(t))$.

Proposition 4.3. Assume $W_i^{**}(0) \leq_{st} W_i^{\pi}(0)$, for all *i*. If $\sum_{i=1}^{L} \mu_i \geq \mu_0 \geq \sum_{i=1, i \neq j}^{L} \mu_i$ for all $j \neq 0$, then $\mathbb{E}(N^{**}(t)) \leq \mathbb{E}(N^{\pi}(t))$, $\forall \pi \in \Pi$ and $\forall t \geq 0$.

Proof As in the previous proof we have by (2) and Lemma 4.1 that

$$\frac{1}{\mu_0} \mathbb{E}(N_0^{**}(t)) + \frac{1}{\mu_i} \mathbb{E}(N_i^{**}(t)) \le \frac{1}{\mu_0} \mathbb{E}(N_0^{\pi}(t)) + \frac{1}{\mu_i} \mathbb{E}(N_i^{\pi}(t)).$$
(7)

Similarly, we can conclude from Lemmas 3.1 and 4.1 that at time t there are classes j and k, $j \neq k \in \{1, ..., L\}$, such that

$$\frac{1}{\mu_0} \mathbb{E}(N_0^{**}(t)) + \frac{1}{\mu_j} \mathbb{E}(N_j^{**}(t)) + \frac{1}{\mu_k} \mathbb{E}(N_k^{**}(t)) \\
\leq \frac{1}{\mu_0} \mathbb{E}(N_0^{\pi}(t)) + \frac{1}{\mu_j} \mathbb{E}(N_j^{\pi}(t)) + \frac{1}{\mu_k} \mathbb{E}(N_k^{\pi}(t)).$$
(8)

Now multiply (7) by $\mu_0 - \sum_{l=1, l \neq i}^L \mu_l \ge 0$, for i = j, k and by μ_i for all $i \neq 0, j, k$; multiply inequality (8) by $\sum_{i=1}^L \mu_i - \mu_0 \ge 0$ and sum these L+1 inequalities to obtain $\sum_{i=0}^L \mathbb{E}(N_i^{**}(t)) \le \sum_{i=0}^L \mathbb{E}(N_i^{\pi}(t))$.

4.2 Stochastic optimality

It is worth noting that despite the stochastic inequality (4) the above arguments can not be strengthened to prove that π^* and π^{**} in fact stochastically minimize the number of users for the given parameter values. This can, however, be accomplished using a dynamic programming (DP) approach similar to that in Section 5 below. For the case L = 2 the following two results are proved in [20]:

Proposition 4.4. If $\mu_1 + \mu_2 \leq \mu_0$, then policy π^* stochastically minimizes the total number of users.

Proposition 4.5. If $\mu_1, \mu_2 \leq \mu_0$ and $\mu_1 + \mu_2 \geq \mu_0$, then policy π^{**} stochastically minimizes the total number of users.

5 Large class-0 users

Again assuming exponential service requirements, we now explore the uncovered case when there exists an j = 1, ..., L, such that $\sum_{i=1, i\neq j}^{L} \mu_i \ge \mu_0$. Since a stochastically optimal policy may in general not exist, we focus instead on the average-optimal policy, i.e., the policy that minimizes $\mathbb{E}(N^{\pi})$ over all policies $\pi \in \Pi$.

We again focus on the case of two nodes and hence consider service rates such that $\mu_0 < \mu_i$ for at least one i = 1, 2. Intuitively it is clear that when there are users of both classes 1 and 2 present, serving them will be optimal. When there are only users of classes 0 and 1 present and $\mu_1 < \mu_0$, serving class 0 seems appropriate, since it is work-conserving in both nodes and it maximizes the total output rate. However, when $\mu_0 < \mu_1$, there is no obvious rule which class to serve. The next proposition states that in such situations, there exists a switching curve that determines which class is optimal to serve, i.e. there exists a function $h(\cdot)$ such that it is optimal to serve class 0 at full rate if $N_1(t) \le h(N_0(t))$ and to serve class 1 at full rate otherwise.

Proposition 5.1. Assume that $\mu_1 > \mu_0$. If both classes 1 and 2 are non-empty, then the expected average-optimal stationary policy serves these simultaneously. While class 2 is empty, the optimal policy is characterized by a switching curve (class 1 is only served if there are sufficient class-1 users). If, in addition, $\mu_0 \ge \mu_2$, then class 0 is served while class 1 is empty.

In the remainder of this section we outline the proof of this proposition. We denote by *i*, *j* and *k* the numbers of class-0, class-1 and class-2 users, respectively. It will be convenient to focus on the uniformized Markov chain. That is, transition epochs (possibly 'dummy' transitions that do not alter the system state) are generated by a Poisson process of uniform rate $\nu \doteq \lambda_0 + \lambda_1 + \lambda_2 + \mu_0 + \mu_1 + \mu_2$. We assume $\nu = 1$ without loss of generality. Using DP, we minimize the mean number of users for the embedded uniformized process, which is equivalent to minimizing that of the original process.

The direct costs that are incurred each time state (i, j, k) is visited, are i + j + k, which implies that the objective is to find a policy π that minimizes $\mathbb{E}(N^{\pi})$. The DP equation can be written as:

$$\begin{split} V_{n+1}(i,j,k) &= i+j+k \\ &+ \lambda_0 V_n(i+1,j,k) + \lambda_1 V_n(i,j+1,k) + \lambda_2 V_n(i,j,k+1) \\ &+ \min\{\mu_0 V_n((i-1)^+,j,k) + (\mu_1 + \mu_2) V_n(i,j,k), \\ &\mu_0 V_n(i,j,k) + \mu_1 V_n(i,(j-1)^+,k) + \mu_2 V_n(i,j,(k-1)^+)\}, \end{split}$$

with $V_0(i, j, k) = i + j + k$.

The existence of an optimal switching curve when there are no class-2 users present is equivalent to the value function, V(i, j, k), satisfying Properties 1 and 2 below. By symmetry, similar properties need to hold for the existence of a switching curve when there are no class-1 users.

Property 1: If it is optimal to serve class 1 in state (i, j, 0), then this is optimal in state (i, j+1, 0) as well, or equivalently, if

$$\mu_0 V(i, j, 0) + \mu_1 V(i, j - 1, 0) + \mu_2 V(i, j, 0)$$

$$\leq \mu_0 V(i - 1, j, 0) + \mu_1 V(i, j, 0) + \mu_2 V(i, j, 0),$$

then

$$\mu_0 V(i, j+1, 0) + \mu_1 V(i, j, 0) + \mu_2 V(i, j+1, 0) \\ \leq \mu_0 V(i-1, j+1, 0) + (\mu_1 + \mu_2) V(i, j+1, 0).$$

Note that this property is implied by the following inequality:

$$\begin{split} & \mu_0 V(i,j+1,0) + \mu_0 V(i-1,j,0) + 2\mu_1 V(i,j,0) \\ & \leq \mu_0 V(i,j,0) + \mu_0 V(i-1,j+1,0) \\ & + \mu_1 V(i,j-1,0) + \mu_1 V(i,j+1,0). \end{split}$$

Property 2: If it is optimal to serve class 0 in state (i, j, 0), then this is optimal in state (i + 1, j, 0) as well, or equivalently, if

$$\mu_0 V(i-1,j,0) + \mu_1 V(i,j,0) + \mu_2 V(i,j,0) \\ \leq \mu_0 V(i,j,0) + \mu_1 V(i,j-1,0) + \mu_2 V(i,j,0),$$

then

$$\mu_0 V(i, j, 0) + \mu_1 V(i + 1, j, 0) + \mu_2 V(i + 1, j, 0)$$

$$\leq (\mu_0 + \mu_2) V(i + 1, j, 0) + \mu_1 V(i + 1, j - 1, 0).$$

This property is implied by

$$\begin{aligned} & 2\mu_0 V(i,j,0) + \mu_1 V(i+1,j,0) + \mu_1 V(i,j-1,0) \\ & \leq \mu_0 V(i+1,j,0) + \mu_0 V(i-1,j,0) \\ & + \mu_1 V(i+1,j-1,0) + \mu_1 V(i,j,0). \end{aligned}$$

These properties can be established for V(i, j, k) by proving them for all $V_n(i, j, k)$ using induction on the time index *n*, see [20] for details.

6 Numerical experiments

We now compare the performance of the optimal policy with that of α -fair bandwidth-sharing policies. We denote by $N_i^{(\alpha)}$ the mean number of class-*i* users as function of α . In the linear network, the α -fair allocation is

$$s_0 = rac{n_0}{n_0 + (\sum_{l=1}^L n_l^{lpha})^{1/lpha}} \quad ext{and} \quad s_i = 1 - s_0,$$

where s_j is the rate allocated to class j, see [3].

For the proportional fair allocation ($\alpha = 1$), the mean number of users is given by

$$\mathbb{E}(N_0^{(1)}) = \frac{\rho_0}{1-\rho_0} \left(1 + \sum_{i=1}^L \frac{\rho_i}{1-\rho_0-\rho_i}\right)$$

and $\mathbb{E}(N_i^{(1)}) = \frac{\rho_i}{1-\rho_0-\rho_i}$, i = 1, ..., L, see [12]. For general α -fair allocations ($\alpha \neq 1$) we conducted simulations in order to estimate the mean number of users. In our experiments we chose $\alpha \in A = \{0, 1, 2, 4, 6, 8, 10, \infty\}$. Besides $\alpha = 1$, the case $\alpha = 2$ will receive particular attention as well, because it is a common abstraction for TCP's bandwidth allocation.

Comparing the mean number of users for the proportional fair allocation and policy π^* already provides important insight. For L = 2 we have that $\mathbb{E}(N^*) - \mathbb{E}(N^{(1)})$ equals

$$\frac{\rho_0}{1-\rho_0} \Big(\frac{\rho_1}{1-\rho_0-\rho_1} \Big(\frac{\mu_1}{\mu_0} - 1 \Big) + \frac{\rho_2}{1-\rho_0-\rho_2} \Big(\frac{\mu_2}{\mu_0} - 1 \Big) \Big).$$

Note that for $\mu_0 < \bar{\mu}_0 := \frac{\lambda_1(1-\rho_0-\rho_2)+\lambda_2(1-\rho_0-\rho_1)}{\rho_1(1-\rho_0-\rho_2)+\rho_2(1-\rho_0-\rho_1)}$ (relatively large class-0 users), the proportional fair allocation does better than π^* , and that the difference is unbounded as $\mu_0 \to 0$. For $\mu_0 > \bar{\mu}_0$ (relatively small class-0 users), it is better to prioritize class 0. In fact, π^* achieves the minimum mean number of users among all strategies in Π , if $\mu_0 \ge \mu_1$ and $\mu_0 \ge \mu_2$. Still, the difference is limited by $-\frac{\rho_0}{1-\rho_0} \left(\frac{\rho_1}{1-\rho_0-\rho_1} + \frac{\rho_2}{1-\rho_0-\rho_2}\right)$. Thus, the proportional fair allocation performs well over a wide range of parameter values.

We now proceed to numerically investigate whether the latter finding holds in greater generality. The optimal policy is computed by DP after truncating the state space. In cases where the optimal policy is known explicitly, we verified that the results from DP are accurate. We examined a wide range of scenarios in terms of the values of the parameters λ_i and μ_i , i = 0, 1, 2. Since the results were qualitatively similar in the various scenarios, we only present the results for the cases with $\rho_0 = 0.3$, $\rho_2 = 0.3$, $\mu_1 = 0.5$, $\mu_2 = 1$, with either (A) $\rho_1 = 0.2$ or (B) $\rho_1 = 0.5$, and varying μ_0 .

In Figures 1 and 2 we plot the total mean number of users under different policies as a function of μ_0 for cases A and B, respectively. The smallest mean number of users among all α -fair policies (min_{\alpha \in A}(\mathbb{E}(N^{(\alpha)}))) is labeled with "opt α fair", the mean number of users for the optimal policy in each point is indicated by "dp", and the curve " π *", corresponds to the function in (1). The other curves correspond to proportional fairness ($\alpha = 1$) and an abstraction of TCP ($\alpha = 2$).

From Figures 1 and 2 we see that the performance of α -fair policies compares well with that of the optimal policy. The gap does not exceed 20%. Apparently, α -fair policies succeed in *dynamically* adjusting the rate allocation in an efficient manner, without any knowledge of the service rate parameters. It is also striking that the differences within the class of α -fair policies are small, and that the mean total number of users is fairly insensitive to the value of μ_0 (for fixed ρ_0). In all cases, the optimal value of α is either 0 (for small values of μ_0) or ∞ (for large values of μ_0). The transition point occurs approximately at $\mu_0 = \overline{\mu}_0$.

In Figures 3 and 4 we plot the total mean number of users as a function of α for two values of μ_0 , for cases A and B, respectively. Again, the results agree with what could be expected: for large μ_0 it is optimal to prioritize class 0, while for small μ_0 it is better to achieve a large degree of parallelization. The difference between the best and the worst α -fair allocations is roughly 5% and 10% in cases A and B, respectively.



Figure 1: Total mean number of users in case A.



Figure 2: Total mean number of users in case B.

7 Summary and conclusions

In order to investigate the efficiency of standard allocation mechanisms such as α -fair policies, we have determined the delay-optimal allocation policies in a simple linear network with exponential service requirements. The optimal scheduling policies require a high degree of coordination within the network as well as knowledge of the service requirement distributions, which may prohibit actual implementation. As a benchmark, though, they are extremely useful to assess the effectiveness of other bandwidth-sharing strategies. In all our experiments we observed that (i) the differences within the class of α -fair allocations are not significant, and (ii) these allocations compare well with the optimal strategies.

The above-mentioned results concern rate allocation *across* flow classes (corresponding to flows sharing a common route), and do not account for scheduling *within* classes. As mentioned in the introduction, it was shown in [21] that standard size-based scheduling strategies such as SRPT and LAS applied across all flows can cause instability effects. However, size-based scheduling *within* flow classes may still produce substantial performance benefits, provided the rate allocation *across* flow classes is carefully arbitrated to avoid the above instability



Figure 3: Total mean number of users in case A.



Figure 4: Total mean number of users in case B.

phenomena. Exactly how to combine size-based scheduling within classes with a stable rate arbitration across classes, and what the potential gains might be, is non-trivial and remains as a challenging topic for further research.

References

- Bansal, N., Harchol-Balter, M. (2001). Analysis of SRPT scheduling: investigating unfairness. In: Proc. ACM SIGMETRICS & Performance 2001 Conf., Boston MA, USA, 279–290.
- [2] Ben Fredj, S., Bonald, T., Proutière, A., Regnié, G., Roberts, J.W. (2001). Statistical bandwidth sharing: a study of congestion at the flow level. In: *Proc. ACM SIGCOMM* 2001.
- [3] Bonald, T., Massoulié, L. (2001). Impact of fairness on Internet performance. In: Proc. ACM SIGMETRICS & Performance 2001 Conf., Boston MA, USA, 82–91.
- [4] Bonald, T., Proutière, A. (2004). On stochastic bounds for monotonic processor sharing networks. *Queueing Systems* 47, 81–106.
- [5] Chen, X., Heidemann, J. (2003). Preferential treatment for short flows to reduce Web latency. *Comp. Netw.* **41**, 779–794.

- [6] Cohen, J.W., Boxma, O.J. (1983). Boundary Value Problems in Queueing System Analysis. North-Holland, Amsterdam.
- [7] Crovella, M.E., Bestavros, A. (1997). Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.* 5, 835–846.
- [8] Friedman, E., Henderson, S.G. (2003). Fairness and efficiency in Web servers. In: Proc. ACM SIGMETRICS 2003 Conf., San Diego CA, USA, 229–237.
- [9] Harchol-Balter, M., Schroeder, B., Bansal, N., Agrawal, M. (2003). Size-based scheduling to improve Web performance. ACM Trans. Comp. Syst. 21, 207–233.
- [10] Kelly, F.P., Williams, R.J. (2004). Fluid model for a network operating under a fair bandwidth-sharing policy. Ann. Appl. Prob. 14, 1055-1083.
- [11] Kleinrock, L. (1976). Queueing Systems, Vol. II: Computer Applications. Wiley, New York.
- [12] Massoulié, L. Roberts, J.W. (2000). Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems* 15 185–201.
- [13] Mo, J., Walrand, J. (2000). Fair end-to-end based congestion control. IEEE/ACM Trans. Netw. 8, 556–567.
- [14] Núñez-Queija, R. (2000). Processor-Sharing Models for Integrated-Services Networks. Ph.D. Thesis Eindhoven University of Technology.
- [15] Rai, I.A., Urvoy-Keller, G., Biersack, E.W. (2003). Analysis of LAS scheduling for job size distributions with high variance. In: *Proc. ACM SIGMETRICS 2003 Conf.*, San Diego CA, USA, 218–228.
- [16] Righter, R. (1994). Scheduling. In: Stochastic Orders and their Applications, eds. M. Shaked and J.G. Shanthikumar, Academic Press, 381–432.
- [17] Righter, R., Shanthikumar, J.G. (1989). Scheduling multiclass single-server queueing systems to stochastically maximize the number of successful departures. *Prob. Eng. Inf. Sc.* 3, 323–333.
- [18] Takagi, H. (1991). Queueing Analysis, Vol. I: Vacation and Priority Systems. North-Holland, Amsterdam.
- [19] de Veciana, G., Lee, T.-L., Konstantopoulos, T. (2001). Stability and performance analysis of networks supporting elastic services. *IEEE/ACM Trans. Netw.* 9, 2–14.
- [20] Verloop, I.M. (2005). Efficient flow scheduling in resource-sharing networks. Master thesis, Utrecht University.
- [21] Verloop, I.M., Borst, S.C., Núñez Queija, R. (2005). Stability of size-based scheduling disciplines in resource-sharing networks. In Proc. Performance 2005.
- [22] Wierman, A., Harchol-Balter, M. (2003). Classifying scheduling policies with respect to unfairness in an M/GI/1. In: Proc. ACM SIGMETRICS 2003 Conf., San Diego CA, USA, 238–249.
- [23] Yang, S., de Veciana, G. (2002). Size-based adaptive bandwidth allocation: optimizing the average QoS for elastic flows. In: Proc. IEEE Infocom 2002, New York NY, USA, 657–666.

[24] Yang, S., de Veciana, G. (2004). Enhancing both network and user performance for networks supporting best-effort traffic. *IEEE/ACM Trans. Netw.* 12, 349–360.