Université d'Ottawa • University of Ottawa

PERMISSION DE REPRODUIRE ET DE DISTRIBUER LA THÈSE

PERMISSION TO REPRODUCE AND DISTRIBUTE THE THESIS

NOM DE L'AUTEUR / NAME OF AUTHOR:	Natalija VLAJIC
ADRESSE POSTALE / MAILING ADDRESS:	815-2201 Riverside Drive Ottawa, ON K1H 8K9
GRADE / DEGREE: Ph.D.(Electrical Engineering)	**ANNÉE D'OBTENTION / YEAR GRANTED** 2003
TITRE DE LA THÈSE / TITLE OF THESIS: Single-cell and Hierarchical Wireless Data Broadcast Systems: Modeling, Performance Analysis, and Optimal Scheduling	

L'auteur permet, par la présente, la consultation et le prêt de cette thèse en conformité avec les règlements établis par le bibliothécaire en chef de l'Université d'Ottawa. L'auteur autorise aussi l'Université d'Ottawa, ses successeurs et cessionnaires, à reproduire cet exemplaire par photograhie ou photocopie pour fins de prêt ou de vente au prix coûtant aux bibliothèques ou aux chercheurs qui en feront la demande.

Les droits de publication par tout autre moyen et pour vente au public demeureront la propriété de l'auteur de la thèse sous réserve des règlements de l'Université d'Ottawa en matière de publication de thèses.

N.B. LE MASCULIN COMPREND ÉGALEMENT LE FÉMININ

The author hereby permits the consultation and the lending of this thesis pursuant to the regulations established by the Chief Librarian of the University of Ottawa. The author also authorizes the University of Ottawa, its successors and assignees, to make reproductions of this copy by photographic means or by photocopying and to lend or sell such reproductions at cost to libraries and to scholars requesting them.

The right to publish the thesis by other means and to sell it to the public is reserved to the author, subject to the regulations of the University of Ottawa governing the publication of theses.

May 1, 2003

DATE

Natalija Vlajic

(AUTEUR) SIGNATURE (AUTHOR)

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

VLAJIC, Natalija

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Single-cell and Hierarchical Wireless Data Broadcast Systems: Modeling, Performance Analysis, and Optimal Scheduling

Charalambos Charalambous

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

I. Stojmenovic

A. Banihashemi

I. Lambadaris

M. Maheswaran

J.-M. De Koninck, Ph.D.

Jean-Marie De Koninck

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Single-cell and Hierarchical Wireless Data Broadcast Systems:

Modeling, Performance Analysis, and Optimal Scheduling

by Natalija Vlajic

Ph.D. Thesis
under the supervision of
Prof. C. D. Charalambous and Prof. D. Makrakis

School of Information Technology and Engineering
University of Ottawa
Ottawa, ON, Canada

May, 2003

0-612-79318-4

Abstract

Wireless Data Broadcast (WDB) is known as a highly efficient information delivery mechanism of nearly unlimited scalability. Over the last few years, due to the appealing properties and a wide area of possible application, a large number of new solutions and ideas related to WDB have been proposed. However, most of those solutions employ overly simplified assumptions concerning the system operation and user behaviour, and therefore fail to provide a broader insight into the nature and performance of WDB as found in the real world.

The work presented in this thesis aims to overcome the main limitations of the previous published research works on WDB. In particular, the following contributions are made.

- We propose a new model of TDM-based single-cell WDB systems, which extends over both broadcast and unicast data retrieval principles, and assumes realistic-impatient user behaviour patterns. Based on this model, we define and derive exact mathematical expressions for several performance measures that seem most appropriate for the analysis of WDB systems. Consequently, we prove that there exists a single broadcast scheduling scheme ($S_{optimal}$), which can ensure the optimal system performance with respect to all of the measures at once, as well as the system's throughput, QoS and GoS.
- The actual search for $S_{optimal}$, both in WDB systems of uniform and variable user mobilities, turns out to be a complex non-linear double inequality-constrained optimization problem, without a tractable closed-form solution. However, by exploiting some mathematical properties of the main cost function, we prove that the given optimization problem can be considerably simplified. Based on this simplification, we derive a closed form approximate expression for $S_{optimal}$ and, consequently, we propose an algorithm for fast, CPU conserving estimation of $S_{optimal}$. Experimental results verify that the proposed algorithm requires minimum computation, while providing performance almost identical to $S_{optimal}$ obtained through numerical estimation.
- For the completeness of our discussion on single-cell WDB systems, we also consider the possibility of frequency- division-multiplexing (FDM) WDB. We prove that for any

given FDM-based broadcast schedule there exists a TDM-based broadcast schedule that results in a better overall system performance. With this proof, we ultimately justify our initial decision to make TDM-WDB the main focus of this work.

- In the final chapter of this thesis, we extend our discussion to the case of multi-cell hierarchically organized WDB systems. Namely, although Hierarchical Cell Structure (HCS) has already become a preferred form of cellular organization in most mobile/wireless networks, there are almost no published research works concerning wireless data broadcast in systems of HCS organization. As our contribution to the given subject, we propose a model of multi-cell hierarchically organized WDB (i.e. HCS-WDB) systems and prove their considerable performance superiority over single-layer multi-cell WDB systems. Subsequently, by exploiting some of the earlier results, we propose an algorithm for fast identification of the optimal broadcast schedule in WDB systems of HCS organization, and verify the efficiency of the given algorithm trough a series of test-case simulation results.

Acknowledgments

First, I would like to thank my co-supervisors: Prof. Charalambos Charalambous - for the valuable feedback, numerous discussions and many interesting ideas given throughout my work on this thesis, and Prof. Dimitrios Makrakis - for the helpful insights and suggestions, particularly concerning the application aspects of wireless data broadcast.

I would also like to thank Prof. Thomas Kunz, from Carleton University, for giving me the help and guidance during the first year of my Ph.D. studies.

Special thanks to Prof. Howard Card, my M.Sc. thesis supervisor from the University of Manitoba, for all the life and research related inspiration, help, and encouragement.

Also, special thanks to my family: my brother Sasha – without his enormous understanding, patience and encouragement none of this would ever be possible, and my parents – for the unconditional love, care, and support.

Finally, I would like to thank the numerous members of the Broadband Wireless and Internetworking Research Laboratory, for providing an interesting and inspiring research environment.

Content

List of Figures

List of Tables

List of Standard Acronyms

3G	third generation wireless cellular mobile systems
BS	base station
CBS	cell broadcast service
FBE	fast broadcast scheduling scheme estimation
GoS	grade of service
GSM	global system for mobile communications
HCS	hierarchical cellular structure
HWDB	hierarchical wireless data broadcast
MAC	medium access channel
MC	Markov chain
MU	mobile user
PCS	personal communications services
QoS	quality of service
SDS	service discovery service
SMS	short message service
UMTS	universal mobile telecommunications system
WDB	wireless data broadcast

List of Acronyms and Symbols Introduced in This Work

λ	overall MU arrival rate [*users/time unit*]
λ^*	single user request re-submission rate [*attempt/time unit*]
$\mathbf{a=(1-P_{success})P_{reattempt}}$	
A	broadcast bandwidth cost ratio
b(i)	bandwidth assigned to broadcasting of item(i) [*bits/time unit*]
B(i)	bandwidth assigned to broadcasting of item(i) [*bits*]
BR	broadcast service ratio
BRBC	broadcast service ratio per broadcast bandwidth cost
C	entire broadcast schedule - there exists a $k_i \in N$ such that $C=k_i \cdot s_i$, $(\forall i=1,2,..,M)$
CB	cost per broadcast bandwidth unit [*1/bps*]
CBB	broadcast bandwidth cost
dp	length of data pointer [*bits*]
dp	length of data pointer [*time units*]
HBR	broadcast service ratio in HWDB system
K	Lagrange multiplier
***l*(i)**	length of item(i) [*bits*]
l(i)	length of item(i) [*time units*]
LR	loss service ratio
n(i)	number of item(i) broadcasts within an interval of fixed duration
$\mathbf{n_{BR}(t_1,t_2,item(i),t_3)}$	number of users who arrive in the system within $[t_1,t_2)$ requesting item(i) and eventually get served through the broadcast channel at time t_3
$\mathbf{n_{LR}(t_1,t_2,item(i),t_3)}$	number of users who arrive in the system within $[t_1,t_2)$ requesting item(i), and leave without obtaining any service before time t_3
$\mathbf{n_{UR}(t_1,t_2,item(i),t_3)}$	number of users who arrive in the system within $[t_1,t_2)$

	requesting item(i) and get served through one of the unicast connections before time t_3
p(i)	probability of item(i) being requested by a MU
p(i,j)	probability of item(i) being requested by a MU from pico-cell(j)
$\mathbf{P_{success}}$	probability that a MU successful obtains an on-demand channel on one of his (re)attempts
$\mathbf{P_{reattempt}}$	probability that a MU, after an unsuccessful attempt to seize one of on-demand channels, remains in the system
q(i,k)	probability of item(i) being requested by a class(k) user
r	broadcast channel datarate [*bits/time unit*]
r(j)	fraction of users who appear in pico-cell(j)
$\mathbf{R^{*}}$	ratio of HBR($\breve{S}_{optimal}$) and $BR_{multicell\text{-}pico\text{-}only}(\tilde{S}_{optimal})$ under fixed overall broadcast bandwidth constraint
$\mathbf{R^{**}}$	BRBC of a generalized HWDB system over BRBC of a generalized single-layer WDB system ratio
req(t_1,t_2,item(i))	number of requests for item(i) received by the system within $[t_1,t_2)$
RP	reneging probability
s(i)	broadcast period associated with item(i) in a single-cell WDB system [*time unit*]
$\mathbf{s_{pico}(i,j)}$	broadcast period associated with item(i) in pico-cell(j) of a hierarchical WDB system [*time unit*]
$\mathbf{s_{micro}(i)}$	broadcast period associated with item(i), given the item is placed at the micro-cell level of a hierarchical WDB system [*time unit*]
S={s(1), s(2), .., s(M)}	broadcast scheduling scheme
$\mathbf{S_{optimal}}$	broadcast scheduling scheme that maximizes corresponding BR
$\tilde{S}_{optimal}$	$S_{optimal}$ estimated using FBE algorithm
$\dddot{S}_{optimal}$	$S_{optimal}$ estimated using modified FBE algorithm
$\breve{S}_{optimal}$	approximated $S_{optimal}$ in generalized HWDB based system

sm	service mode (sm=0 – no service, i.e. service loss; sm=1 – broadcast service; sm=2 – unicast service)
$\mathbf{t_a}$	MU's arrival time in the system
$\mathbf{t_d}$	MU's departure time from the system
UR	unicast service ratio
$\mathbf{wt_i(t_1,t_2,sm)}$	waiting time experienced by a MU who arrives at time t_1, wants to obtain item(i), and gets service $sm \in \{0,1,2\}$ at time t_2
$\mathbf{wp(t_d,item(i))}$	expected number of users waiting to obtain item(i) at s time t_d
WP	waiting users population [*users*]
WT	waiting time [*time units*]

Chapter 1

Introduction

1.1 WDB Data Delivery Principles

According to recent statistics ([*UMTS_Report9*] and [*UMTS_ConvergingNetwork*]), the number of data enabled mobile devices capable of accessing the Internet will soon outnumber the quantity of Internet-capable Personal Computers (PCs). Thus, as the statistics indicate, the fusion of mobile-wireless-data services is becoming a preferred users' choice over services of the traditional fixed-wired-data networks. This should come at no surprise, since mobile-wireless-data systems are able to offer a number of features that cannot be provided by fixed-wired-data networks, including: any place any time connectivity, location-based information on-demand, association of a mobile device with the user rather than location, etc. [*Schiller*]. Clearly, as the popularity and, consequently, overall demands on mobile-wireless systems increase, the importance of mechanisms for scalable and efficient data delivery, in these systems, increases accordingly.

There are two fundamental data delivery mechanisms found in mobile-wireless-data communications systems: point-to-point access and wireless data broadcast (WDB) (see Figure 1.1 and 1.2).

In the case of point-to-point access, a logical channel is established between the mobile user (MU) and the server. Queries are submitted to the server and results returned to the user in the same way as in a wired network [*Xu*]. In the case of wireless data broadcast, on the other hand, data are sent simultaneously to all users residing in the broadcast area. It is up to the user to select the data it wants [*Xu*].

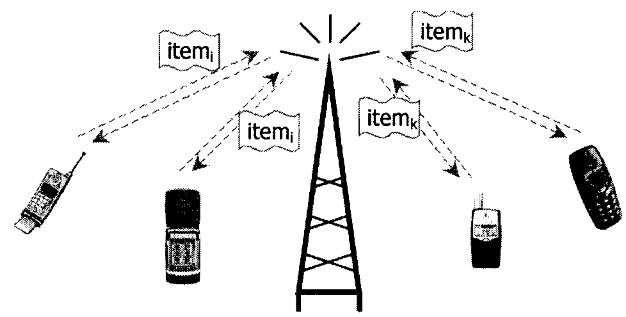


Figure 1.1 Point-to-point data delivery

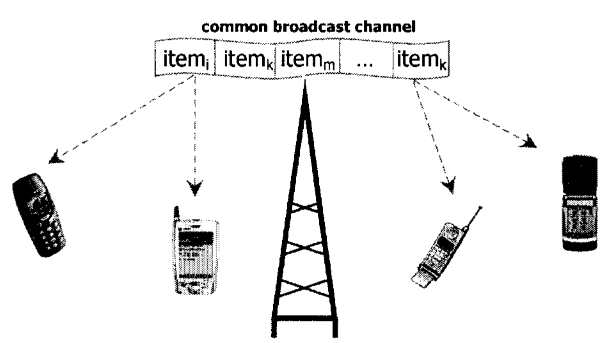


Figure 1.2 WDB data delivery

Compared to point-to-point access, wireless data broadcast offers several significant advantages, including:

(1) ***excellent scalability***, i.e. the potential to serve almost unlimited number of users at only single channel bandwidth requirements. As an illustration, let us imagine an arbitrary

number of users demanding the same information, at approximately the same time. While, in such a case, a point-to-point based system would have to serve each of the users independently on a separate channel, in a WDB based system all of the users, regardless of their actual number, could be served simultaneously on a single common – broadcast channel.)

(2) ***low operational cost*** It should be noted that wireless communication systems essentially employ a broadcast medium, i.e. the air, to deliver information [*Xu*]. Therefore, in contrast to data broadcast in wired networks, wireless data broadcast can be implemented without introducing any additional cost.

(3) ***consequent improvements in QoS[1] offered by the system*** Based on (1), wireless data broadcast saves considerably on the overall utilization of network resources. Hence, the saved resources could be used for the further improvement of QoS given to those users that insist on obtaining point-to-point access.

(4) ***consequent improvements in GoS[2] offered by the system*** By providing service, i.e. information, even to the users whose requests have not been received explicitly, WDB is capable of reducing contention at the random access channel (RACH) level [*Schiller*], and, consequently, improving the overall system's grade of service.

Based on the above, wireless data broadcast has the potential to considerably enhance the corresponding network's performance. However, it should be noted that the actual effectiveness of WDB is not provided by default. It strongly depends on a number of factors, including: 1) the presence of a critical mass of users demanding the same or similar information, 2) the system's ability to identify the information for which there is a vigorous common MU's demand, and 3) the system's ability to appropriately prioritize the broadcasting of highly- over less- relevant, i.e. popular, information.

[1] QoS – set of parameters used to describe a network's performance, including: link delay, available bandwidth, packet delay variation, packet error rate, … [*Schiller*].

[2] GoS – the probability of a call or data request being blocked or delayed for more than a specified interval [*Schiller*].

1.2 Previous Research Works on WDB

Over the last few years, due to the appealing properties and a wide area of possible application, wireless data broadcast has attracted considerable attention of the research community. Recently, a large number of new solutions and ideas related to WDB have been proposed. In this section we provide an overview of those published research works that are directly or somewhat related to our work, as presented in Chapters 2 to 6.

1.2.1 Pure-push vs. Pure-pull vs. Hybrid WDB

From the conceptual and modeling point of view, WDB-based systems are grouped into three categories:

- *Pure-Push WDB Systems*

  In pure-push WDB based systems (see, for example, [*Acharya_1*], [*Hameed*], [*Jain*] and [*Vaidya*]), MUs are just passive listeners – they do not have any direct control over what is being broadcasted. They have to patiently wait for the desired information to appear on the broadcast channel. The relative popularities of items (documents) considered for broadcasting are assumed known. Based on the known items' popularities, the server creates a periodic/aperiodic broadcast schedule and disseminates the information accordingly.

  Among the ideas and concepts regarding push-based WDB systems, the ones presented in [*Jain*] and [*Vaidya*] are already considered classical and have been exploited by many other authors working in the filed. In [*Jain*], it has been implicitly[3] proven that the expected delay experienced by MUs demanding a specific item is minimized if the given item gets

[3] Unfortunately, we were unable to identify a published research work on push-based WDB systems that provides the exact mathematical proof concerning the optimality of 'equal-spacing' broadcast. However, in Appendix 1, by using our mathematical/theoretical framework, we present such a proof.

broadcasted at equal time intervals. In [*Vaidya*], on the other hand, the proof of the well-known 'square-root rule' has been presented. The rule states that the minimum overall MUs' waiting time is achieved provided the spacing between two consecutive instances of a given data item is proportional to the square-root of its length ($\sqrt{l}$) and inversely proportional to the square-root of its access probability ($\sqrt{p}$) [*Xu*].

- ***Pure-Pull WDB Systems*** [4]

As indicated in the above paragraph, in pure-push WDB systems MUs are passive, and the broadcasted information is tailored so as to satisfy the needs of the majority [*Xu*]. On the other hand, in pure-pull WDB systems (see, for example, [*Acharya_2*] and [*Aksoy*]), MUs are assumed active - they have the access to *uplink channels*. Thus, when a client needs a data item, it sends an on-demand request to the server. The server queues up the clients' requests as they come along, then repeatedly chooses one of the requests, explicitly responds to it through the broadcast channel, and finally removes it from the queue.

Among the published works on pure-pull WDB, [*Acharya_2*] and [*Aksoy*] are considered classical. In [*Acharya_2*], the benefits of *pre-empting broadcast* in heterogeneous[5] pure-pull WDB environments have been proven. (In queueing theory, pre-emption is well known as a means to improve the overall response time, typically by letting a chain of small jobs interrupt a larger job.) Also, in [*Acharya_2*] it has been argued that the standard response (waiting+service) time is not a well-suited performance measure in the heterogeneous on-demand WDB setting. Instead, the *stretch of a request*, defined as the ratio of the response time of a request to its service time, has been proposed as a more fair measure. In [*Aksoy*], a heuristics based scheduling algorithm (RxW algorithm) has been proposed. This algorithm schedules for the next broadcast the item with the maximal RxW, where R is the number of outstanding requests for that item, and W is the amount of time that the oldest of these requests has been waiting for. The experimental results have shown that RxW algorithm is

[4] These systems are also referred to as 'on-demand WDB systems'.
[5] Here, 'heterogeneous' refers to 'heterogeneous user requests', i.e. user requests of unequal sizes.

capable of balancing the average and worst-case waiting time, thus providing an overall better performance than some earlier proposed algorithms.

- ***Hybrid WDB Systems***

Hybrid WDB (see, for example, [*Stathatos*] and [*LinCW*]) exploits the characteristics of both, pure-pull and pure-push WDB, and integrates them in a way that better matches the overall MU's demands. In particular, in hybrid WDB-based systems, the users are assumed active and have the access to a number of uplink channels. The access frequencies of the available data items are estimated from the statistics of on-demand requests received on the uplink channels. Consequently, the most frequently accessed items are delivered to MUs through pure-push WDB, while the least frequently accessed items are provided either through pure-pull WDB or through separate point-to-point channels.

The main problem associated with the performance of a hybrid WDB system comes from the users who obtain desired information without participating in the request submission procedure, i.e. the users whose requests are 'invisible' to the system. Namely, if upon arrival in the system users find the information they need already included in the broadcast schedule, they may see no point of submitting any request to the server. Hence, as the content of the pure-push broadcast channel captures the actual MUs' needs more accurately, the population of the 'invisible' users grows accordingly, and the statistics extracted from the submitted on-demand requests becomes less accurate. As a solution to the given problem, [*Stathatos*] suggests that each data item gets periodically removed from the broadcast channel in order to probe its true popularity, at the price of less than optimal overall system performance. On the other hand, [*LinCW*] proposes that each user locally collects statistics regarding data items recently received from the broadcast channel, and then piggybacks this information to the server with one of his next 'on-demand' requests. This way, assuming the full co-operation of the MUs, the broadcast server gets a chance to gain a thorough insight into the actual information demands, but with an inherent delay.

1.2.2 Optimal Broadcast Scheduling Scheme in Hybrid and Pure-Push WDB Systems

One of the biggest problems of hybrid WDB systems is related to the accurate estimation of MUs' access frequencies, as described in Section 1.2.1. However, two equally challenging problems faced by both, hybrid and pure-push WDB systems, are: 1) how to determine the optimal *broadcast schedule* assuming the true access frequencies are known, and 2) once the optimal broadcast schedule is found, how to actually implement the given schedule in real time. Figure 1.3 shows that the three major problems found in hybrid WDB systems are, in fact, inherently related to each other.

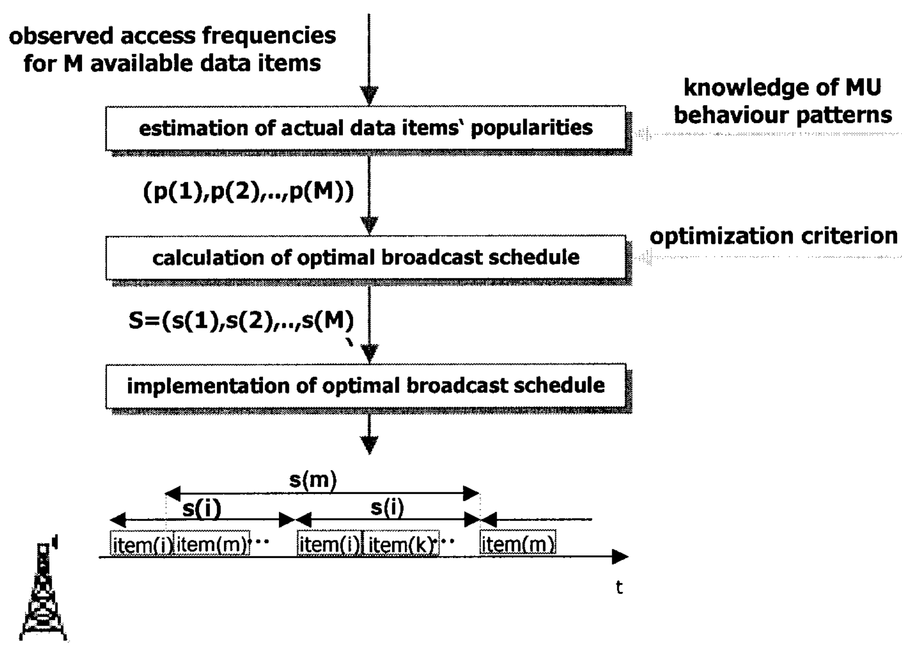


Figure 1.3 Taxonomy of optimization problems in hybrid WDB systems

In general, a *broadcast schedule*[6] determines what data items should be broadcasted by the server and at what frequency [*Vaidya*]. Clearly, different broadcast scheduling schemes may

[6] Please note, in this and the proceeding sections we will interchangeably use the expressions 'broadcast schedule' and 'broadcast scheduling scheme'.

have entirely different affect on the system, i.e. QoS and GoS provided to its users. For example, a broadcast schedule that prioritizes highly- over less- popular items is intuitively expected to provide a better overall throughput, and consequently a better MU service rate. In a similar manner, broadcast scheduling schemes devised so as to satisfy different optimization criteria might turn out to have entirely different characteristics.

In [*Vaidya*], the mean MU waiting time has been considered as the performance metric subject to optimization, under the following scenarios: 1) a WDB system with a single broadcast channel and no probability of transmission errors, 2) a WDB system with a single broadcast channel and the probability of un-correctable transmission errors, 3) a WDB system with multiple broadcast channels and no probability of transmission errors. In all three cases it has been assumed that MUs do not submit any on-demand requests, and yet they patiently wait in the system until the needed information is retrieved. By using a probabilistic approach and assuming that instances of either item are equally spaced, the authors of [*Vaidya*] have arrived at the exact expression concerning the lower bound on achievable mean access time (t_{min}), for each of the considered scenarios. From the necessary conditions for t_{min}, they have derived the implicit requirements concerning the optimal broadcast scheduling scheme (i.e. the broadcast scheduling scheme that can ensure t_{min}).

According to our knowledge, [*Jiang_1*] is the only published research work on the subject of WDB that deals with the issue of user impatience. Namely, as stated in [*Jiang_1*]: "... for various reasons, users may lose patience after waiting too long, and leave, although their requests are still unsatisfied ...". The actual user behaviour model, employed in [*Jiang_1*], assumes that each MU, upon arrival in the system, passively waits w [time units] for the requested information to appear on the broadcast channel, where w is an exponentially distributed random variable. After this time, if the information still has not been retrieved, the MU leaves. The theoretical results presented in [*Jiang_1*] have shown that the broadcast schedule capable of minimizing the average waiting time in a system populated with MUs acting according to the behaviour as previously described, also is capable of maximizing the expected service ratio. The implicit requirements concerning such a broadcast schedule have been presented in [*Jiang_1*] as well.

Most published works on WDB evaluate the effectiveness of broadcast scheduling schemes based on how they reduce the overall mean response time. In [*Jiang_2*], however, the variance of MUs' waiting time has been considered as the performance metric subject to optimization. Namely, according to [*Jiang_2*], "... minimizing the mean response time most benefits a "virtual" user whose request patterns happen to be coincident with the overall item demand pattern on which the broadcast schedule is based. An individual user whose demand pattern differs from the overall demand pattern may experience a mean response time greatly worse than the optimal value." The theoretical results presented in [*Jiang_2*] contain the exact mathematical expression for the minimal variance of the response time in a pure-push WDB system (σ_{min}) and the implicit requirements concerning the broadcast scheduling scheme that can ensure σ_{min}.

1.2.3 Optimal Broadcast Scheduling Scheme Implementation

As Figure 2.1 illustrates, the actual scheduling of data items on the broadcast channel should take place only after an optimal broadcast scheduling scheme, i.e. an optimal set of broadcast periods/frequencies, has been found. Based on the examples of previous section it should be evident that one can devise a broadcast scheduling scheme with various goals in mind, including minimization of expected MUs' waiting time, minimization of expected MUs' waiting time variance, maximization of service ratio, etc. Nevertheless, the actual real-time implementation of an optimal broadcast schedule often turns out to be a rather complex task. Moreover, the studies presented in [*Baruah_1*] and [*Baruah_2*] show that, even theoretically, not every set of n periodic tasks can be scheduled on a single shared resource so as to satisfy the constraints of each individual task. Expressing this in another way, there are cases when it might be hard, if possible at all, to ensure the required periodicity for all broadcast items.

In [*Acharya_1*], a relatively simple model, which can be fine-tuned to approximate a particular broadcast schedule, has been proposed. The model is called *Broadcast Disk* (Bdisk), and as the name implies, it resembles a hierarchy of logical disks spinning at different speeds. Given an actual set of data items and the corresponding broadcast scheduling scheme, the algorithm based on Bdisk model operates as follows: 1) all data items are ordered by their expected broadcast frequencies; 2) data items in the same range of broadcast frequencies are grouped on the same disk; 3) the disks are split into a number of *chunks*, where the chunk size equals the Least Common Multiple (LCM) of the disks' relative broadcast frequencies; 4) chunks from various disks are interleaved resulting in an approximation of the desired broadcast schedule. The most significant advantages of the Bdisk-based scheduling algorithm are its simplicity and the preserved periodicity of the obtained schedule. However, the algorithm provides no firm guarantees regarding how well the obtained schedule can match the analytical optimum.

In [*Hameed*], a useful observation regarding the analogy between broadcast scheduling and packet fair queueing has been provided. Packet fair queueing algorithms attempt to: 1) assign at least fraction f(i) of the output bandwidth to input queue(i), and 2) assign the given bandwidth in 'evenly' rather than 'bursty' manner. Similarly, broadcast scheduling algorithms attempt to: 1) assign a fraction of broadcast channel bandwidth to item(i) as specified by the optimal broadcast schedule, and 2) assign the given bandwidth so as to ensure that instances of item(i) are equally spaced. By exploiting the given analogy, the authors of [*Hameed*] have proposed a low-overhead bucket-based broadcast scheduling algorithm. According to the algorithm, the broadcast data items are partitioned into several buckets, which are kept as cyclical queues. Each bucket is associated with an expression – the expression corresponds to the optimal common broadcast frequency of the items found in the bucket. At any time, the first item in the bucket for which the given expression evaluates to the largest value gets to be broadcasted. The main advantage of the bucket-based broadcast scheduling algorithm is its ability to obtain a broadcast schedule very close to the analytical optimum. Nevertheless, the algorithm cannot ensure the periodicity of the obtained schedule.

1.3 WDB Application Areas

In this section, we provide several examples of real world WDB-based communication systems. In each of the systems, the creation and utilization of an adequate and computationally inexpensive broadcast schedule is shown to be of the uttermost significance. Accordingly, the given systems represent a direct application area for our work, as presented in Chapters 2 to 4.

1.3.1 Cell Broadcast Service (CBS)

Cell Broadcast Service (CBS) is defined within Phase 2+ of the ETSI GSM standard [*ETSI*]. CBS has been also specified in the Release '99 of the UMTS standard, under the name Service Area Broadcast. Thus, the same CBS functionality offered within GSM is also available within UMTS [*CBF*].

CBS has some similarities with Short Message Service[7] (SMS), since both services use the GSM network's signalling path. However, while SMS is one-to-one and one-to-a-few service, CBS is one-to-many geographically focused service [*ETSI*].

The main concept of CBS service is analogous to the Teletex service offered on television. Namely, like Teletex, CBS permits a number of unacknowledged general CBS messages to be broadcasted to all receivers within a particular geographical area [*ETSI*]. This can be an area as large as the entire network, or an area as small as a single cell.

The basic network structure of a CBS system (see Figure 1.4), comprises of the following elements [*ETSI*]:

[7] Short Message Service (SMS) offers transmission of messages of up to 160 characters. SMS messages do not use the standard data channels of GSM, but exploit unused capacity in the signalling channels. Thus, sending and receiving of SMS messages is possible during data or voice transmission [*Schiller*].

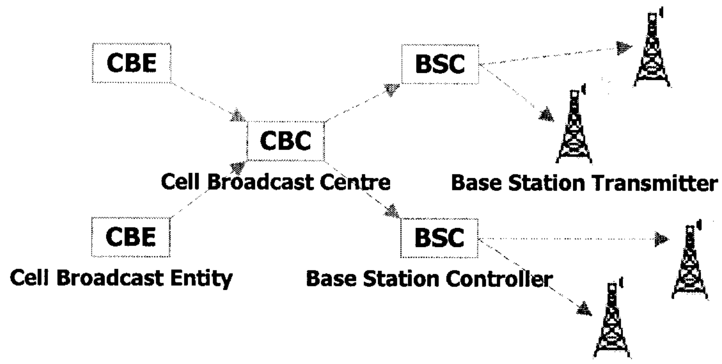


Figure 1.4 Network structure of CBS system

1) Cell Broadcast Entity (CBE)

The functionality of CBEs is outside of the scope of GSM and UMTS specifications. However, it is assumed that CBEs allow content providers to locally or remotely prepare messages for broadcast. Any number of CBEs can be connected to a single Cell Broadcast Centre.

2) Cell Broadcast Centre (CBC)

CBCs are responsible for the management of CBS messages, including:

- allocation of serial numbers to individual messages;
- determining the set of cells to which a CBS message should be broadcasted;
- determining the time at which a CBS message should commence/cease being broadcasted;
- determining the period at which broadcast of a CBS message should be repeated[8];
- determining the actual cell broadcast channel.

3) Base Station Controller (BSC)

BSCs are responsible for:

- storage of CBS messages;
- scheduling of CBS messages on CBCH[9];

[8] Based on the discussion from the previous section, CBC is, in fact, responsible for the creation of an adequate broadcast scheduling scheme.

[9] While CBC is responsible for the creation of broadcast schedule, i.e. for determining the optimal inter-broadcast intervals for a given set of data items (see Sections 1.2.2 and 1.2.3), BSC is responsible for the actual scheduling, i.e. for implementation of the created broadcast scheduling scheme.

- providing an indication to CBC when the desired repetition period cannot be achieved;
- routing CBS messages to the appropriate Base Transceiver Stations, etc.

4) *Base Transceiver Station (BTS)*

BTSs are responsible for forwarding CBS control information and for delivering broadcast messages to mobile subscribers (MSs).

5) *Mobile Subscriber/User Equipment (MS/UE)*

MS/UEs are responsible for recombination of the blocks received via the radio path to reconstitute the CBS message. Also, it is assumed that MS/UEs

- enable the user to activate/deactivate CBS through the use of Man Machine Interface (MMI);
- have the ability to discard a CBS message which has a message identifier indicating that its subject matter is not of interest to the MS;
- have the ability to ignore repeated broadcasts of CBS messages already received;
- optionally read the extended broadcast channels, etc.

The CBS messages comprise of 82 octets, which equates to 93 characters. Up to 15 such messages, i.e. pages, may be concatenated to form a macromessage. In such a case, each page will have the same message identifier (indicating the source of the message), and the same serial number [*CBF*].

Due to its main characteristics – omni-point data delivery through unused capacity of broadcast control signalling channel (BCCH) (for more on the functionality of BCCH see [*Schiller*]), CBS technology is considered particularly suitable for the dissemination of the following types of information [*CBF*]:

- traffic information to alert for accidents, road work, traffic jams, detours, etc.;
- weather information such as storm warnings, tornado watches, etc.;
- advertisements about sales and promotions at a particular location;
- announcements about parking, restaurant, emergency ambulance availability in an area;
- event information at conferences, trade shows, seminars, etc.;

- o tourist information;
- o latest news, stock exchange rates.

Although the potential of CBS to provide information in timely and scalable manner considerably exceeds the potential of other GSM/UMTS services, the actual effectiveness of CBS depends on a number of factors, including:

- ***Choosing the right set of broadcast frequencies***

The frequency at which CBS messages are repeatedly transmitted should dependent on the information they contain. For example, it is intuitively expected that dynamic information, such as road traffic information, requires more frequent transmission than weather information. The repetition period should also be affected by the desire for CBS messages to be received by high-speed mobiles, which rapidly traverse cells [*ETSI*].

- ***Deciding on the number of broadcast channels employed***

In general, CBS messages can be broadcasted on two different cell broadcast channels, which are characterized by different QoS. However, a MS is always able to read the basic channel, while the reading of the extended channel sometimes may collide with other tasks of the MS. Therefore the probability of receiving a CBS message on the extended channel is smaller than on the basic channel. The reading of the extended channel for MS is optional. The scheduling on this channel needs to be done independently [*ETSI*].

1.3.2 *Hybrid UMTS – DxB[10] Networks*

According to [*UMTS_ConvergingNetwork*], migration to satellite-based digital broadcasting will be a major trend worldwide over the next 10 years. (Both satellite-based digital television and digital radio network, i.e. DVB and DAB, are already available in several countries.) Due to their digital nature, DxB networks are capable of delivering a wide range of content to end-users, at exceptionally high data rates. (For mobile users, the achievable data rates are up to 15 Mbit/s.)

[10] Digital Audio Broadcasting and Digital Video Broadcasting [*UMTS_ConvergingNetwork*].

However, in spite of the high available data rates, DxB networks have generally been engineered for distribution of the same content to many users. Thus, they may not be effective in delivering highly personalized information, as this requires the utilization of a return channel.

The cooperation with UMTS-based systems can be considered as a way of extending the application range of DxB networks [*UMTS_ConvergingNetwork*]. These are some of the possible cooperation scenarios:

- ***Network Sharing and Terminal Integration***

Applications can be designed so as to share the resources of a UMTS network and a broadcast network (see Figure 1.5). In such a case, UMTS could be used for all personalized information (point-to-point communication, return channel), whereas the broadcast network could be used for information that is suited for wider distribution (one-to-many) [*UMTS_ConvergingNetwork*].

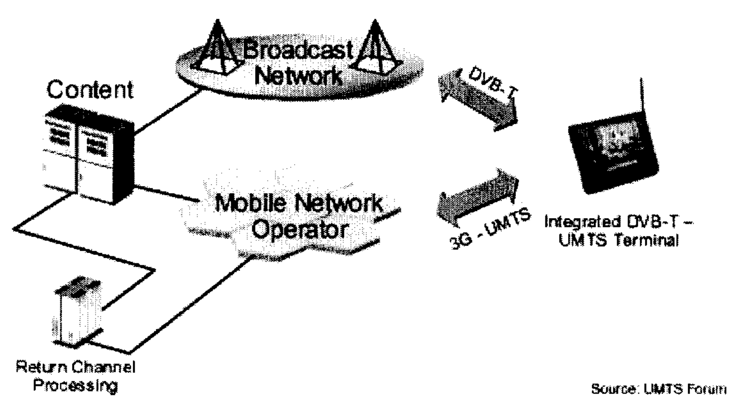


Figure 1.5 UMTS – DxB terminal integration

- ***DxB as an extension to UMTS.***

In contrast to the full integration, as discussed in the previous paragraph, another possibility is to consider DxB just as an extension of UMTS, where DxB would provide information download that is in high demand by users (e.g. electronic newspapers, software, guidebooks and maps).

This would relieve the demand for resources within UTRA, while keeping the air interfaces of the two networks separate [*UMTS_ConvergingNetwork*].

- ***UMTS as Back Channel for Interactive Broadcast Services (TV, Data)***

In this scenario, the UMTS return channel would allow the user to respond to the information received via a DxB network. It would not necessarily have to be from the same terminal (see Figure 1.6).

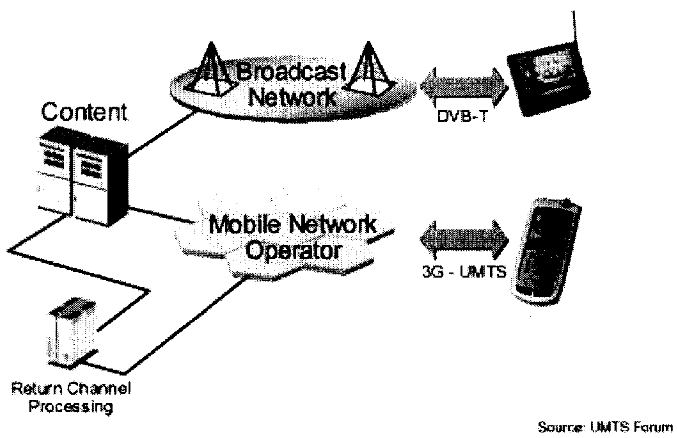


Figure 1.6 UMTS return channel for DxB networks

For example, the user could wish to send a short video of themselves for inclusion in a programme being transmitted via DVB-T or they could wish to take part in a music vote, via their 3G terminal, taking place on a DAB radio station.) A UMTS/3G return channel could provide a number of advantages over other return channels, including

1) relatively high bandwidth – i.e. a UMTS return channel would allow the user to send back multimedia content if desired;

2) interaction with mobile users – i.e. with a UMTS return channel, the user could interact with a DxB service from any location [*UMTS_ConvergingNetwork*].

1.3.3 Networks of Pervasive Devices

The rapid spreading of mobile computerized devices (including laptops, GSM phones, personal digital assistants and digital watches) has already marked the beginning of a new computing paradigm characterized by ad hoc - proximity based networking [*Hermann*]. Further advances in the microprocessor and wireless communications technology are expected to result in a significant number of new devices (e.g. smart home-appliances, sensors, wearable devices, etc.) capable of spontaneous communication [*Hermann*]. In the literature, such devices are commonly referred to as *pervasive devices*.

In most published works on ad-hoc networking (e.g. [*Czerwinski*], [*Gribble*], [*Gupta*], [*Hermann*], [*Nidd*]), networks of pervasive devices are envisioned as highly dynamic environments in which clients, i.e. devices, are aware of their surroundings and peers, and at the same time, are capable of providing services to and using services from peers effectively [*Gupta*]. Given this assumption, one of the major challenges imposed on networks of pervasive devices is the utilization of intelligent mechanisms and protocols (the so-called Service Discovery Service (SDS) or node discovery protocols), through which computing devices would be able to detect the presence of neighbouring devices - as well as share configuration and service information in an efficient manner [*Nidd*].

According to the architecture proposed in [*Czerwinski*], a secure and scalable SDS system is composed of three main components (see Figure 1.7):

1) ***services***;

2) ***clients*** - who want to discover services that are currently running in the network;

3) ***SDS servers*** - which solicit information from the services, and then use this information to fulfill client queries.

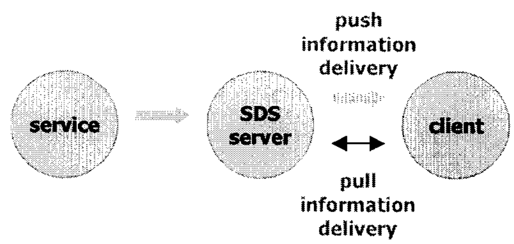


Figure 1.7 Service Discovery Service entities

SDS servers, as proposed in [*Czerwinski*], support both push-based and pull-based information delivery. While pull-based (on-demand) delivery offers clear advantages in terms of information privacy and security, push-based (multicast/broadcast) delivery of SDS advertisements provides considerably higher dissemination efficiency and almost unlimited end-device scalability [*Nidd*].

Push-based SDS information delivery, according to [*Gribble*], relies on the utilization of a well-known global multicast/broadcast channel. SDS advertisements, i.e. SDS messages, are sent periodically through this channel, and the maximum individual advertisement rates are determined by estimating the actual message, i.e. service, popularities [*Czerwinski*][11]. Nevertheless, in a system populated with hundreds of thousands of portable clients, as envisioned in the case of networks of pervasive devices, rapid changes in the popularity and/or availability of certain services can be expected. Accordingly, one of the most important functions of an SDS server would be to quickly react to such changes, and appropriately modify the content and/or its scheduling of SDS messages on the push-based multicast/broadcast channel [*Czerwinski*].

[11] In general, measurement-based periodicity provides better dissemination efficiency and keeps the traffic from overloading SDS servers as the number of clients and services grows [*Czerwinski*].

1.4 Summary of Our Work

The main contributions of our work, and accordingly the content of this thesis, can be summarized as follows:

In Chapter 2, we present the main contributions of our work concerning single-cell wireless broadcast systems. In particular, first we outline the main limitations found in the previous research works on wireless data broadcast, including some common unrealistic system and user behaviour assumptions. Subsequently, we propose a new model of single-cell WDB systems, which extends over both broadcast and unicast data retrieval principles, and also assumes realistic (*probabilistic-impatient*) user behaviour patterns. For the given model, we define several appropriate system performance measures and derive the exact mathematical expression for each. Based on these expressions, we prove that in any given single-cell WDB system there exists a unique broadcast scheduling scheme (we annotate this scheme with $S_{optimal}$), which simultaneously optimizes the system performance with respect to the overall throughput, QoS and GoS. Next, we show that through the appropriate selection of system and user behaviour parameters, the model of single-cell hybrid WDB systems can readily be transformed into a few simpler models of some well-known data retrieval systems. Consequently, the analysis of these systems turns out to be trivial, as it allows for simple re-employment of the results already obtained in the analysis of single-cell hybrid WDB systems. In Chapter 2, we also discuss the significance and complexity associated with the estimation of data-item request probabilities in single-cell hybrid WDB systems. In particular, we prove that an accurate estimation of data-item request probabilities should not be exclusively based on the statistics extracted from the unicast requests, but it should also take into consideration the current value of the broadcast scheduling scheme and the values of other system and user behaviour parameters. Finally, we present a series of experimental results obtained by simulating the performance of a real-world single-cell hybrid WDB system. The significance of these results is multi-fold: 1) they provide the full verification and prove the robustness of the earlier obtained analytical expressions, and 2) they enable us to draw a few significant conclusions concerning both the performance and analysis of single-cell WDB systems, which would be impossible to obtain otherwise (i.e. analytically).

In Chapter 3, we prove that the search for the actual value of $S_{optimal}$ is a rather complex, non-linear double inequality-constrained optimization problem, without a tractable closed-form solution. However, by using the method of slack variable, we show that the given problem can be transformed into a much simpler, single equality-constrained optimization problem. Consequently, by demonstrating the (strict) convexity of both, the problem's feasible region and its main cost function, we prove the existence of a single $S_{optimal}$ in any particular WDB based system. By exploiting some specific properties of the main cost function, we show that the given optimization problem (stated in Section 3.1) can be further simplified, and an approximate closed from expression for $S_{optimal}$ can be obtained. Based on the given closed-form expression, we propose a CPU conserving algorithm for fast estimation of $S_{optimal}$. Subsequently, through a series of experimental results, we verify that the proposed algorithm requires minimum computation, while providing performance almost identical to $S_{optimal}$ obtained through numerical estimation.

In Chapter 4, we extend the discussion from Chapter 3 to the case of single-cell hybrid WDB systems with variable user behaviours and mobility patterns. In particular, first we introduce the concept of 'dwell-time random variable' and explain its significance in characterizing users' mobility. Next, we argue that with a minor modification, our realistic-impatient MU behaviour model can be used in representing diverse user behaviours and mobility patterns, and we show that such a modified model results in a dwell-time random variable that is proven suitable for the analysis of micro-cellular systems with a heterogeneous user population. Subsequently, we propose a modified Fast Broadcast Scheduling Scheme Estimation (FBE) algorithm for the optimal broadcast scheduling scheme estimation in WDB systems with users of variable behaviours and mobilities. Examples of the obtained experimental results suggest that the proposed algorithm requires minimum computation, while providing performance very close to $S_{optimal}$ found through numerical estimation. At the end of Chapter 4, we show that in the presence of a heterogeneous user population, the estimation of data-item request probabilities becomes an increasingly complex problem, as it requires that the actual knowledge of various user mobility patterns be taken into consideration.

The analysis presented in Chapters 2 to 4 was built on the assumption of a time-division-multiplexing (TDM) based broadcast scheduling. However, for the completeness of our discussion, in Chapter 5 we consider the possibility of frequency-division-multiplexing (FDM) based broadcast scheduling. More specifically, we show that for any given FDM-based broadcast schedule there exists a TDM-based broadcast schedule that results in a better overall system performance. Hence, by proving the superiority of TDM- over FDM- WDB, we ultimately justify our initial decision to make TDM-WDB the main focus of this work.

In Chapter 6, we introduce the concept of 'hierarchical cell structure' (HCS) and discuss the numerous advantages associated with the utilization of HCS as a design model for multi-cellular wireless/mobile networks. Next, by employing the main principles of HCS, we propose a model of hierarchically organized multi-cell WDB (HWDB) systems with user retrials. Subsequently, we show that the search for the optimal broadcast schedule ($S_{optimal}$) in an HWDB system exceeds the complexity of the corresponding problem as found in an equivalent single-layer WDB system. However, by exploiting our earlier obtained approximate closed-form expression for single-cell $S_{optimal}$, we show that the given problem can be considerably simplified. Accordingly, we propose an algorithm for a fast approximation of $S_{otpimal}$ in HWDB systems, and prove the efficiency of the given algorithm through a series of test-cases. We finally conclude Chapter 6 with a comparative theoretical analysis of single-layer WDB v.s HWDB systems. The results of this analysis show that the higher operational cost of HWDB systems, compared to the cost of their single-layer counterparts, is well justifiable, as HWDB systems ultimately provide considerably superior performance in terms of the overall throughput, QoS, and GoS.

In Chapter 7, we introduce DESMO-J – a Java-based discrete-event simulation package. Also, we provide an overview of our simulation software, which has been built on the DESMO-J framework, and enabled us to simulate the performance of single-cell hybrid WDB systems.

Chapter 2

Single-Cell Hybrid WDB Systems: Model Description and Performance Analysis

Overview

The content of this chapter is organized as follows.

Section 2.1 First, we enlist the main limitations commonly found in the previous research works on wireless data broadcast, including some overly simplified system and user behaviour assumptions.

Section 2.2 By aiming to overcome the limitations of the previous research works on WDB, in Section 2.2 we propose a new model of single-cell WDB systems. This model, not only extends over both broadcast and unicast data retrieval principles, but also assumes realistic (*probabilistic-impatient*) user behaviour patterns. Furthermore, we show that this model, in fact, represents the model of a hybrid queueing system with jockeying and reneging users. Accordingly, we argue that the analysis and results presented in the subsequent sections can be considered significant, not only with respect to wireless data broadcast, but also when considered from a more general queueing theory perspective.

Section 2.3 For the given model of single-cell WDB based systems, we define several appropriate performance measures and derive the exact mathematical expression for each. Based on these expressions, we prove that there exists a broadcast scheduling scheme ($S_{optimal}$), which simultaneously optimizes the WDB system performance with respect to the entire set of the proposed performance measures. Moreover, we argue that the same broadcast scheduling scheme also meets the major WDB system goals – the provision of higher throughput, better QoS and GoS. [1]

Section 2.4 Next, we show that through the appropriate selection of system and user behaviour parameters, the model of single-cell hybrid WDB systems (as proposed in Section 2.2) can readily be transformed into a few simpler models of some well-known data retrieval systems. Consequently, the analysis of these systems turns out to be trivial, as it allows for simple re-employment of the results already obtained in the analysis of single-cell hybrid WDB systems.

Section 2.5 In Section 2.5, we discuss the significance and complexity associated with the estimation of data-item request probabilities in single-cell hybrid WDB systems. In particular, we prove that an accurate estimation of data-item request probabilities should not be exclusively based on the statistics extracted from the unicast requests, but it should also take into consideration the current value of the broadcast scheduling scheme and the values of other system and user behaviour parameters.

Section 2.6 Finally, in Section 2.6, we present a series of experimental results obtained by simulating the performance of single-cell hybrid WDB systems, as studied in the preceding sections. The significance of these results turns out to be multi-fold: 1) they provide the full verification and prove the robustness of some of the earlier obtained analytical expressions; 2) they enable us to draw a few significant conclusions concerning both the performance and analysis of single-cell WDB systems, which would be impossible to obtain otherwise (i.e. analytically).

[1] Please note, the actual procedure for finding $S_{optimal}$ is the subject of Chapter 3.

2.1 Motivation

2.1.1 Major Limitations of Previous Research Works on WDB

As discussed in the preceding chapter, wireless data broadcast (WDB) has been recognized as a highly efficient information delivery mechanism. High efficiency, in this particular case, refers to more than one aspect of a WDB-based system performance. For example, by serving almost unlimited number of users at minimum bandwidth requirements, WDB is capable of improving the overall system's throughput, while not compromising the provided quality-of-service (QoS). Also, by being able to provide service even to the users whose requests have not been sent or received explicitly, WDB is proven effective in reducing contention at the random access channel (RACH) level, and, consequently, improving the system's grade-of-service (GoS). (The exact proofs concerning these properties of WDB, in a hybrid-WDB system with MUs of realistic-impatient behaviour, can be found in Section 2.3.)

Over the last few years, due to these appealing properties and a wide area of possible application, wireless data broadcast has attracted considerable attention of the research community. As a result, a large number of new solutions and ideas related to WDB have been proposed.[2] Nevertheless, a deeper study reveals that most of the proposed solutions employ rather unrealistic assumptions, among which the following three appear to be the most common ones:

Limitation 1: Unrealistic system/network scenarios considered.

The observed WDB systems are typically assumed to be of either pure-push or pure-pull type. Pure-push systems are, in fact, pure broadcast. Pure-pull systems, on the other hand, are pure-unicast – they use the broadcast channel just as a return path for point-to-point communication. (For more, see [*Xu*].) In reality, however, most systems combine both

[2] In Chapter 1, we have enlisted the major published works on WDB that are closely or somewhat related to our work. A more extensive overview of the published research works on WDB can be found in [*Xu*].

forms of data communication, unicast and broadcast. For example, the unicast communication is typically used for personalized information, while the broadcast communication deals with information that is in high demand by users. Therefore, the analysis of hybrid broadcast unicast systems, (we call them ***hybrid WDB systems***) seems much more significant from an application point of view, than the analysis of either pure-push or pure-pull systems. ([*Stathatos*] and [*LinCW*] are examples of just a few published research works that deal with WDB systems of the hybrid type.)

Limitation 2: Simplified user-behaviour models employed.

It is typically assumed that every user, during his presence in a pure-pull or a hybrid-WDB system, makes either none or at most one request-submission per each information item retrieved. However, it is well known that most wireless systems employ the so-called 'distributed random access schemes' [*Schiller*], which provide no guarantee in terms of collision avoidance. Hence, in times of a heavy network load, a user may be forced to make a number of attempts before, eventually, succeeding in the request submission. (In the wireless communication literature, such an occurrence is typically referred to as retrial phenomenon. For more, see Section 2.1.2.) In addition to network congestion, users' impatient is another possible cause of multiple submission attempts. Namely, the waiting time is known to have significant impact on users' behaviour. In particular, if after a certain limited time period from the submission of the first request the data does not arrive, either through a dedicated connection or on the broadcast channel, users have the tendency of issuing another request. If even after several request re-submissions there is still no adequate response from the system, users have the tendency of withdrawing from the waiting procedure.

Limitation 3: Homogeneous mobility models assumed.

It is almost exclusively assumed that WDB systems get populated with MUs who behave in a similar manner. However, it is well known that the users found in most real world wireless/mobile systems have rather heterogeneous behaviours. In particular, the users tend to exhibit different levels of mobility, impatience, etc.

Clearly, by being limited in scope with such over-simplifying assumptions, the existing works on wireless data broadcast fail to provide a broader insight into the nature and performance of WDB based systems as found in the real world.

2.1.2 User Retrial Phenomenon

As indicated in Section 2.1.1, the previous research works on wireless data broadcast have largely overlooked the possibility of a single user making repeated request submission attempts. Nevertheless, the retrial phenomenon (see Figure 2.1) has been a subject of numerous studies dealing with the performance aspects of unicast wireless networks (e.g. [*Tran-Gia_1*], [*Tran-Gia_2*], [*Marsan*], [*Kinoshita*], [*Onur*]). In these studies, it has been commonly recognized that the user behaviour in general, and the retrial phenomenon in particular, may have a significant impact on the system performance. For example, as a result of the retrial phenomenon under overload network conditions, a snowballing effect of call/request arrivals can occur, leading to dramatic degradation of the entire network performance [*Tran-Gia_2*].

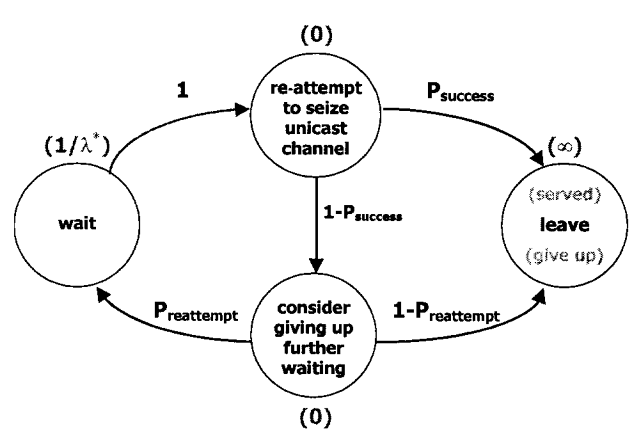


Figure 2.1 User behaviour with retrials in unicast wireless system (imbedded[3] Markov Chain representation)

[3] For more on imbedded MC, see [*Bharucha-Reid*].

Figure 2.1 depicts the repeated-attempt user behaviour model, as found in the literature on unicast wireless systems ([*Tran-Gia_1*], [*Tran-Gia_2*], [*Marsan*], [*Kinoshita*], [*Onur*]). According to the given model, the time between two successive user's retrials is exponentially distributed with mean $1/\lambda^*$. On each retrial, the user succeeds in acquiring a channel with probability $P_{success}$[4]. After each unsuccessful retrial, the user becomes impatient and terminates the retrial sequence, i.e. leaves the system with probability ($1-P_{reattempt}$). Please note, the bracketed values in Figure 2.1 represent the average state occupancy times. Hence, clearly, based on these values, two of the states are instantaneous, with zero occupancy time, while one state is absorbing, with infinity occupancy time [*Cinlar*]. It should be noted that in reality, the time between a user attempting to seize a channel and realizing the attempt has been unsuccessful may not be zero. However, it seems reasonable to assume assumed that this time is negligible, i.e. several orders of magnitude smaller, compared to the time between two consecutive user attempts. For more information regarding the actual values of the model parameters, as found in real-world wireless systems, and their impact on the network performance see [Onur].

[4] Please note, ($1-P_{success}$) represents the well-known 'blocking probability'.

2.2 Model Description

In this section, the main assumptions regarding the model of single-cell hybrid-WDB systems that we consider in this work are outlined.

2.2.1 System Assumptions

The set of data items available through the system is finite ({item(i) | i=1,..,M}). Also, each item is assumed to be of a finite length ($l_i<\infty$, i=1,..,M). The items can be retrieved either through one of the unicast or through the broadcast channel, as illustrated in see Figure 2.2.

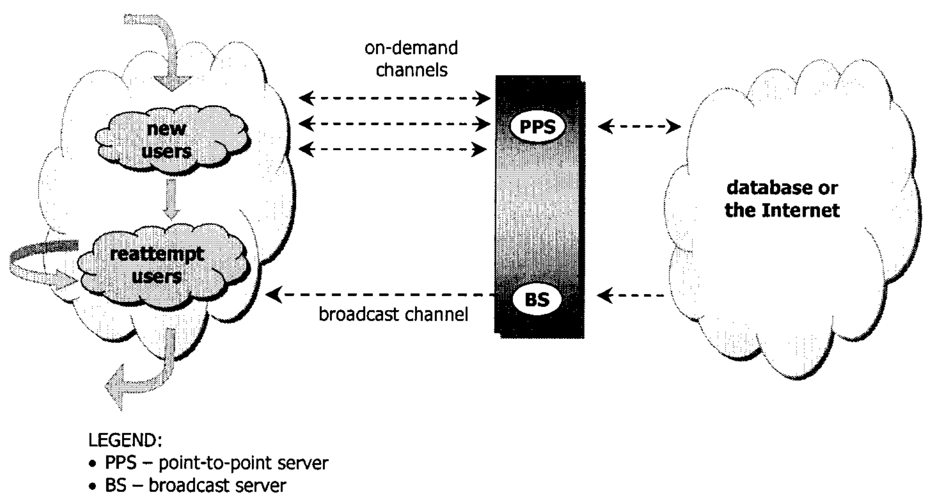


Figure 2.2 Single-cell hybrid WDB system

(Please note, a broadcast retrieval is possible provided the requested item is included in the current broadcast schedule.) Item(i) (i=1,2,..,M) is scheduled to appear on the broadcast channel every s(i) time units, i.e. equal-spacing broadcast is assumed (see Figure 2.3), where s(i) is a real value in the range [0,+∞]. Note, s(i)=+∞ implies that, in fact, item(i) is excluded from the current broadcast schedule. The proof regarding the optimality of equal-spacing broadcast in a generalized WDB system is provided in Appendix 1. Once item(i) appears on the broadcast channel, all MUs which need the given item and happen to be in the waiting phase of the retrieval cycle (see Figure 2.4) are served simultaneously.

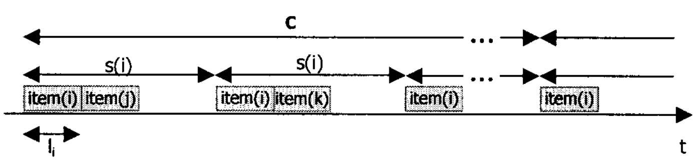


Figure 2.3 Equal-spacing broadcast channel schedule

2.2.2 User Behaviour Assumptions

The arrival of mobile users (MU) in the system is governed by a Poisson process [5] (process arrival rate = λ). Every mobile user who appears in the system wants to retrieve a single data item. In particular, a MU wants to retrieve item(i) with probability p(i) (i=1,..,M), where $\sum p(i)=1$ at any point in time. The behaviour of individual users follows the model presented in Figure 2.4. This model is a version of the model in Figure 2.1, extended for the possibility of data retrieval through wireless data broadcast.

[5] According to our knowledge, all previous research works on WDB have been based on the assumption of a Poisson MU arrival process. This assumption can be justified by the fact that the aggregate behaviour of a large yet finite group of independent and statistically identical mobiles users, situated within the coverage area of a wireless cell, corresponds to a Poisson-like process. For example, if there is a very small yet non-negligible probability for each of the users to initiate a call at any given time unit, then the average number of calls arriving within a time unit is still moderate, i.e. finite. Furthermore, if users decide whether or not to initiate a call procedure independently of each other, then the actual call inter-arrivals times will be also mutually independent.
Due to the above-mentioned reasons, we base our analysis of WDB systems on the Poisson user arrival assumption as well. Nevertheless, in Section 2.6.4, through a series of simulation results, we prove the robustness of our theoretically obtained results to various forms non-Poissonian user arrival processes.

More specifically, the main difference between the models from Figures 2.1 and 2.4 lies in the nature of the listen, i.e. wait, state. According to the model in Figure 2.1, the end-device suspends any activity for as long as the user remains in the wait state. (Please recall, the model in Figure 2.1 has been employed in the literature on pure unicast systems, hence no possibility of data retrieval through wireless data broadcast has been assumed.) On the other hand, in the same state of our model in Figure 2.4, the end-device continues being active by listening to the broadcast channel. Hence, if it happens that the requested item appears on the broadcast channel while the end-device is listening, the end-device will fetch the item and make the user move to the leave, i.e. served state.

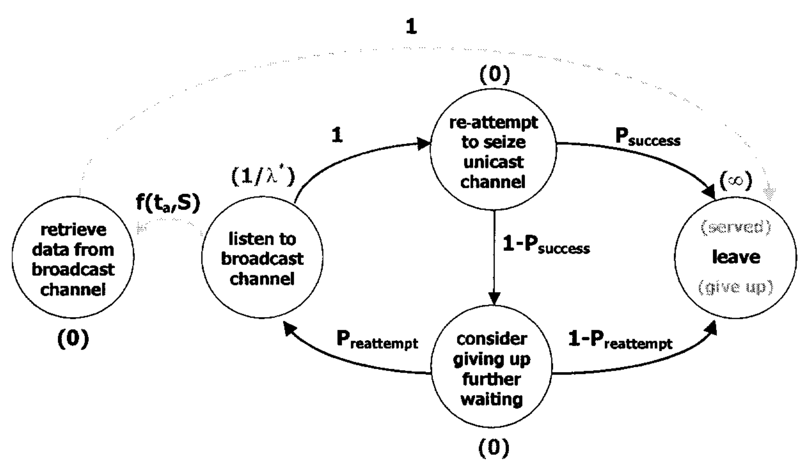


Figure 2.4 User behaviour with retrials in hybrid WDB system (imbedded Markov Chain representation)

More specifically, the main difference between the models in Figures 2.1 and 2.4 lies in the nature of the listen, i.e. wait, state. According to the model in Figure 2.1, the end-device suspends any activity for as long as the user remains in the wait state. (Please recall, the model in Figure 2.1 has been employed in the literature on pure unicast systems, hence no possibility of data retrieval through wireless data broadcast has been assumed.) On the other hand, in the same state of our model in Figure 2.4, the end-device continues being active by listening to the broadcast channel. Hence, if it happens that the requested item appears on the broadcast channel while the end-device is listening, the end-device will fetch the item and make the user move to the leave, i.e. served state.

In contrast to all the other transition probabilities of the model in Figure 2.4, the transition probability between the state of listening and the state of being served through the broadcast channel is not the same for all the users. Instead, it depends on the time when the requested item is scheduled to appear on the broadcast channel relative to the user's arrival time in the system. Consequently, the transition probability between the state of listening and the state of being served through broadcast is expected to be most significant in the case of users whose arrivals in the system slightly precedes the actual broadcast time. Also, please note that we consider the state of listening, i.e. waiting, as the initial state of the model in Figure 2.4.

2.2.3 Model Description: Queueing Theory Perspective

According to the description provided in the preceding section, the hybrid-WDB system could, as well, be perceived as a hybrid queueing system with jockeying and reneging users[6] (see Figure 2.5).

In particular, if we assume that every physical (wireless) channel represents a server (in the queueing theory sense), then there are two different service disciplines found in the generalized WDB system; one corresponding to the unicast, the other to the broadcast channel. Hence, based on the service discipline, the following two queueing subsystems can be identified within the hybrid-WDB system:

[6] Both jockeying and reneging represent special forms of user impatience [*Gross*]. Namely, jockeying occurs when a user moves back and forth among several queues, while reneging occurs when a user leaves the queue (system) after joining it, because the waiting time has become too long. According to [*Gross*], " ... although both phenomena are quite interesting and clearly applicable to numerous real-life situations, they are difficult to analyze, since the probability functions become too complicated and the general concepts become hazy."

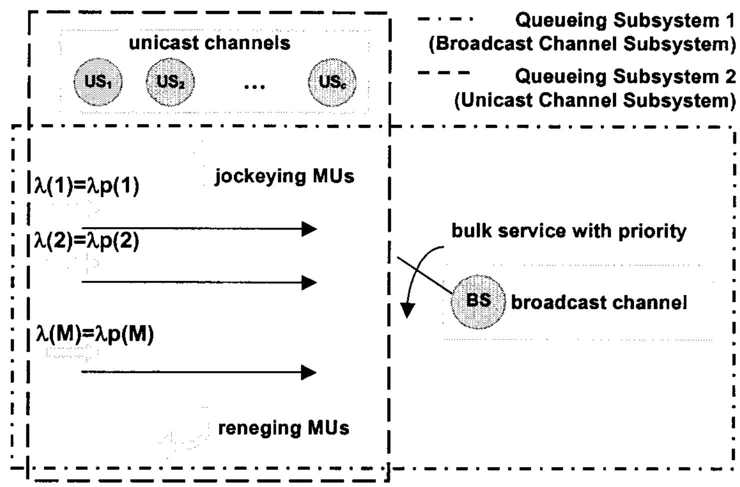


Figure 2.5 Hybrid WDB system as hybrid queueing system

Broadcast Channel Subsystem

From the queueing theory perspective, this subsystem can be characterized as a *M/D/1/∞/PR*[7] *system with multiple queues, bulk service, and jockeying and reneging*[8] *users*, with the following arguments in mind:

- ***multiple queues, jockeying and reneging users*** – The users who arrive in the system requiring to retrieve item(i), and enter the initial state of listening to the broadcast channel, can readily be perceived as a separate queue, e.g. queue(i) (i=1,..,M) (see Figure 2.5). However, the users of such a queue do not remain passive while waiting for service (see Figure 2.4). Namely, they periodically attempt to obtain the requested items through one of the unicast channels, i.e. they periodically jockey to and from Subsystem 2. Also, on every return from Subsystem 1 the users renege, i.e. they decide to give up further waiting and leave the system with a certain probability.

[7] A queueing process is described by a series of symbols and slashes such as A/B/W/Y/Z, where A indicates the interarrival-time distribution, B the service pattern, W the number of parallel service channels, Y the restriction on system capacity, and Z the queue discipline. For more, see [Gross].

[8] Both jockeying and reneging represent special forms of user impatience [Gross]. Namely, jockeying occurs when a user moves back and forth among several queues, while reneging occurs when a user leaves the queue after joining it, because the waiting time has become too long.

- ***bulk service*** – According to the assumptions from the preceding section, once item(i) (i=1,..,M) appears on the broadcast channel, all the users waiting to retrieve this item are served simultaneously. Thus, the operation of Subsystem 1 results in 'bulk-service discipline' with respect to each of its queues.

- ***M*** – As indicated in the preceding section, the user arrivals are assumed to follow a Poisson process with rate λ. Consequently, the user arrivals with respect to $queue_i$ (i=1,..,M) follow a Poisson process with the rate λ(i)=λp(i) (see Appendix 2).

- ***D*** – Again, according to the assumptions from the preceding section, item(i) is scheduled to appear on the broadcast channel exactly every s(i) time units (i=1,..,M). Furthermore, if the size of item(i) is *l*(i) [bits], and r [bps] is the broadcast channel datarate, then the broadcast of item(i) takes exactly l(i)=*l*(i)/r time units (i=1,..,M). Hence, all queues of Subsystem 1 obtain entirely deterministic service.

- ***1*** – The presence of only a single broadcast channel, i.e. a single server in Subsystem 1 is assumed.

- ∞ - There is no waiting capacity constraint imposed on Subsystem 1.

- ***PR*** – If we assume that the broadcast schedule employed by Subsystem 1 is aimed at maximizing the overall system's throughput, then the more popular items get to be broadcasted more often. Accordingly, the users from the queues that correspond to more popular items obtain 'higher-priority service', since they get to be served more frequently.

Unicast Channel Subsystem

This subsystem can be characterized as a *G/M/c/∞/RSS queueing system*. Here,

- ***G*** - indicates that the user/request arrival process, with respect to the unicast channels, follows a general distribution. Namely, the actual user arrivals with respect to the entire system are assumed to be of a Poisson type. However, upon arrival in the system, the users first attempt to obtain service through the broadcast channel, and then they occasionally jockey to and from Subsystem 2. Thus, the assumption of the Poisson arrival process, with respect to Subsystem 2, alone, no longer applies.

- ***M*** - indicates that the service times associated with the servers of Subsystem 2, i.e. the service times on all unicast channels, are iid exponentially distributed random variables. Please note, the actual performance aspects of unicast service are not a subject of this work. However, the assumption regarding exponentially distributed unicast service time is commonly found in the literature [Ortigoza-Guerrero]. Hence, we adopt this assumption as well.
- ***c*** – As stated in the preceding section, the presence of more than one but still a finite number of unicast channels, i.e. servers, in Subsystem 2 is assumed.
- ∞ - Again, as in the case of Subsystem 1, no waiting capacity constraint is imposed on Subsystem 2.
- ***RSS*** – Most wireless systems employ the so-called 'distributed random access schemes', which provide no guarantee in terms of the ordering and delay with which users seize unicast channels or succeed in submitting their requests. Accordingly, it is assumed that Subsystem 2 employs 'random selection for service' discipline.

According to our knowledge, no readily available results concerning the performance of hybrid queueing systems, as presented in Figure 2.5, exist in the literature. Therefore, the analysis and results presented in the subsequent sections could be considered significant from both, the wireless data broadcast and queueing theory perspective.

2.3 Performance Measures

In Section 2.3.2 to 2.3.7 we provide definitions and derive exact mathematical expressions for several performance measures that seem most appropriate for the analysis of a single-cell hybrid WDB system, as described in Section 2.2. These performance measures include: broadcast service ratio (BR), unicast service ratio (UR), loss service ratio (LR), waiting time (WT), reneging probability (RP), and waiting user population (WP). Furthermore, in section 2.3.8, we claim that a broadcast scheduling scheme (S) capable of optimizing the WDB system performance with respect to BR, optimizes the system's performance with respect to all the other measures as well. Subsequently, we argue that such S also meets the major WDB system goals – the provision of improved throughput, QoS and GoS.

Nevertheless, we begin this section by taking a closer look at the Markov chain (MC) representation of the retrial phenomenon, i.e. the user behaviour model, as depicted in Figures 2.1 and 2.4. In particular, by using the standard theory of continuous-time Markov processes, we find the expressions for the most significant transient state probabilities of the given model. (As we will see in Sections 2.3.2 to 2.3.7, these probabilities considerably simplify the actual derivation of the main system performance measures.) Furthermore, we introduce a time-variant version of the user behaviour model, and we highlight the main advantages and difficulties associated with the utilization of such a model.

2.3.1 Preliminary Analysis

2.3.1.1 Preliminary Analysis Assuming Time Invariant User-Behaviour

The model in Figure 2.6 is a modified, but equivalent, version of the imbedded Markov Chain user-behaviour model presented in Figure 2.1 and Figure 2.4. It is more suited for the analysis in continuous-time domain. The following should be noted regarding the states of the modified MC model in Figure 2.6:

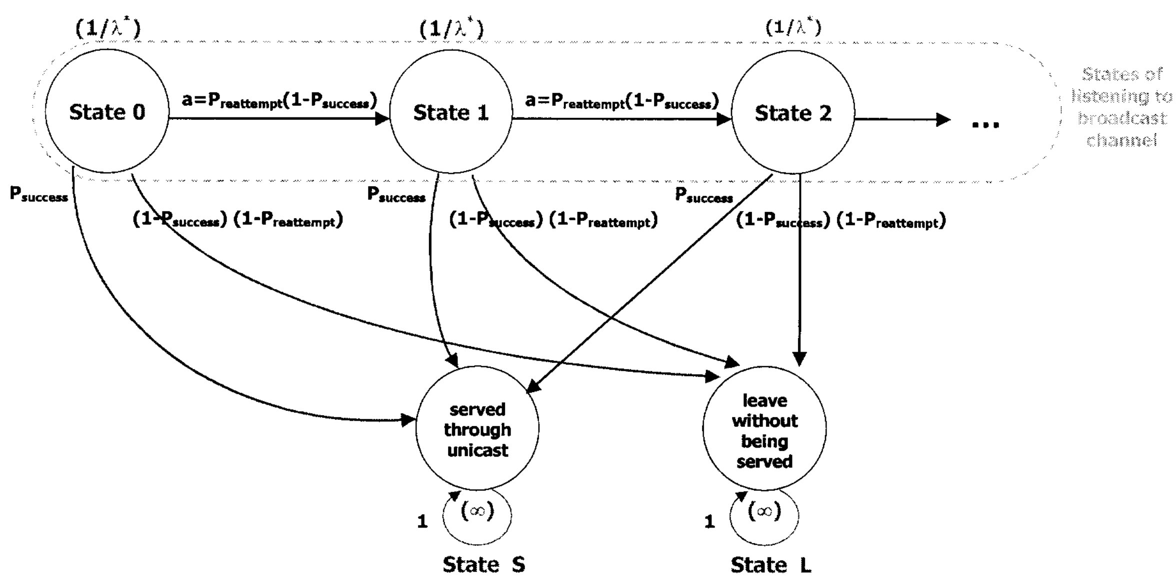


Figure 2.6 Terminating "Poisson process" representation of the model in Figure 2.4

States 0,1,... If the user, who arrived in the system at time t_a, is found in '*State n*' (n=0,1,2,...) at some arbitrary time t_x, this implies that the user has made n unsuccessful attempts of seizing one of the unicast channels in $[t_a, t_x)$. However, in spite the unsuccessful attempts, he still remains in the system, waiting to obtain service. Accordingly, for as long as the user remains in either of *States 1,2 ...* , he has a chance of getting served through broadcast, once the requested item appears on the broadcast channel.

State S If the user, who arrived in the system at time t_a, is found in '*State S*' at time t_x, this implies that, at some point within $[t_a, t_x)$, the user has seized a unicast connection and got served 'on demand'.

State L If the user, who arrived in the system at time t_a, is found in '*State L*' at time t_x, it implies that, at some point within $[t_a, t_x)$, the user got impatient and left the system without obtaining any service.

Now, let us employ the following notation:

$$P(X(t_x) = i \mid t_a = t_y) = P_i(t_y, t_x), \qquad i = 0,1,2,... \tag{2.1}$$

$$P(X(t_x) = S \mid t_a = t_y) = P_S(t_y, t_x) \tag{2.2}$$

$$P(X(t_x) = L \mid t_a = t_y) = P_L(t_y, t_x) \tag{2.3}$$

where X(t) represents the continuous-time Markov chain as illustrated in Figure 2.6, while t_a represents the user's arrival time in the system. Due to the Markovian property [*Leon-Garcia*] of the random process associated with the model in Figure 2.6, it can be easily verified that $P_i(t_y,t_x)=P_i(t_x-t_y)$ $(i=1,2,\ldots)$, $P_S(t_y,t_x)=P_S(t_x-t_y)$, and $P_L(t_y,t_x)=P_L(t_x-t_y)$. Accordingly, based on the preceding discussion, and by substituting $(t_x-t_a)=\tau$, we can write

$$P(\text{user listens to broadcast channel at } t = (t_y + \tau) \mid t_a = t_y) = \sum_{i=0}^{\infty} P_i(\tau) \tag{2.4}$$

$$P(\text{user served through unicast before } t = (t_y + \tau) \mid \text{user arrived at } t_a = t_y) = P_S(\tau) \tag{2.5}$$

$$P(\text{user give up waiting before } t = (t_y + \tau) \mid \text{user arrived at } t_a = t_y) = P_L(\tau) \tag{2.6}$$

According to the theory of continuous-time Markov processes [*Bharucha-Reid*], the actual expressions for the transient state probabilities of the model in Figure 2.6, and subsequently (2.4) to (2.6), can be obtained as a solution to the set of differential equations, known as Chapman-

Kolmogorov differential equations ([*Bharucha-Reid*], [*Leon-Garcia*], [*Cinlar*]) - see (2.7), provided the initial conditions $P(\tau=0)$ are given.

$$P'(\tau) = P(\tau) \cdot Q \tag{2.7}$$

In (2.7), $Q=[q_{kj}]$ represents the so-called generator matrix, and each $q_{kj}=p_{kj}q_j$ corresponds to the rate of moving from *State k* to *State j* ($k,j \in \{1,2,\ldots,S,L\}$); here, p_{kj}-s represent the imbedded MC's transition probabilities, while each q_j represents the rate of moving out of *State j*.

Now, the values of p_{kj}-s and q_i-s, and accordingly the actual value of the generator matrix Q, can be obtained directly from Figure 2.6 and are given in (2.8), where $a=P_{reattempt}(1-P_{success})$.

$$\begin{array}{c}
\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & \ldots & S & L \end{array} \\
Q = \begin{bmatrix}
-\lambda^* & \lambda^* a & 0 & 0 & 0 & \ldots & \lambda^* P_{success} & \lambda^*(1-P_{success})(1-P_{reattempt}) \\
0 & -\lambda^* & \lambda^* a & 0 & 0 & \ldots & \lambda^* P_{success} & \lambda^*(1-P_{success})(1-P_{reattempt}) \\
0 & 0 & -\lambda^* & \lambda^* a & 0 & \ldots & \lambda^* P_{success} & \lambda^*(1-P_{success})(1-P_{reattempt}) \\
\ldots & & & & & & & \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}
\begin{array}{|c} 0 \\ 1 \\ 2 \\ \ldots \\ S \\ L \end{array}
\end{array} \tag{2.8}$$

By employing (2.8) in (2.7), (2.7) gets transformed to the following set of differential equations:

$$P_0'(\tau) = -\lambda^* P_0(\tau) \tag{2.9}$$

$$P_i'(\tau) = \lambda^* a\, P_{i-1}(\tau) - \lambda^* P_i(\tau), \quad i > 0 \tag{2.10}$$

$$P_S'(\tau) = \lambda^* P_{success} \left(\sum_{i=0}^{\infty} P_i(\tau) \right) \tag{2.11}$$

$$P_L'(\tau) = \lambda^* (1 - P_{success})(1 - P_{reattempt}) \left(\sum_{i=0}^{\infty} P_i(\tau) \right) \tag{2.12}$$

Based on the differentiation and integration theory, and by recalling the initial assumptions from Section 2.2.2 (see (2.13)),

$$P_k(0) = \begin{cases} 1, & k = 0 \\ 0, & k = 1,2,\ldots,S,L \end{cases} \tag{2.13}$$

the unique solutions to (2.9), and (2.10), i.e. (2.14) is obtained. (The actual procedure for obtaining the transient state probabilities of a continuous-time Markov process, i.e. a Poisson process, is presented in Appendix 3.)

$$P_i(\tau) = e^{-\lambda^* \tau} \frac{(\lambda^* a\tau)^i}{i!}, \quad \forall i \geq 0 \tag{2.14}$$

By employing (2.14) back in (2.4), the probability that a user happens to be in either of the waiting states (*State i*, i=1,2,...), τ time units upon his arrival in the system, turns out to be

$$\begin{aligned} &P(\text{user listens to broadcast channel at } t = (t_y + \tau) \mid t_a = t_y) = \\ &= \sum_{i=0}^{\infty} e^{-\lambda^* \tau} \frac{(\lambda^* a\tau)^i}{i!} = \\ &= e^{-\lambda^* (1-a)\tau} \end{aligned} \tag{2.15}$$

Similarly, by employing (2.15) in (2.11) and (2.12), the probabilities that a user happens to be in *State S* or *State L*, τ time units upon his arrival in the system, turn out to be of the form (2.16) and (2.17), respectively.

$$\begin{aligned} &P_S(\tau) = P(\text{user served through unicast before } t = (t_y + \tau) \mid t_a = t_y) = \\ &= \lambda^* P_{success} \int_0^{\tau} e^{-\lambda^* (1-a)\tau'} d\tau' = \\ &= \frac{P_{success}}{1-a}\left[1 - e^{-\lambda^* (1-a)\tau}\right] \end{aligned} \tag{2.16}$$

$$
\begin{aligned}
&P_L(\tau) = \mathrm{P}(\text{user give up waiting before } t = (t_y + \tau) \mid t_a = t_y) = \qquad (2.17)\\
&= \lambda^* \frac{(1 - P_{success})(1 - P_{reattempt})}{1-a} \int_0^{\tau} e^{-\lambda^*(1-a)\tau'} d\tau' = \\
&= \frac{(1 - P_{success})(1 - P_{reattempt})}{1-a} \left[1 - e^{-\lambda^*(1-a)\tau}\right]
\end{aligned}
$$

2.3.1.2 Preliminary Analysis Assuming Time-Variant User-Behaviour

The user behaviour model, which we employ in this work (see Section 2.2 and 2.3.1.1), is based on time-invariant assumptions. In particular, we assume that neither of the three user behaviour determining parameters ($P_{success}$, $P_{reattempt}$, λ^*) changes with time. (Please note, according to our knowledge, the other research works that deal with either of the two subjects - retrial phenomenon or wireless data broadcast, employ similar time-invariant assumptions.)

Nevertheless, one might argue that such assumptions do not match the reality, and it is to expect that $P_{success}$ and $P_{reattempt}$ exhibit some form of time-dependency and/or randomness. For example:

random change in $P_{success}$ - due to the stochastic nature of user arrival process and unicast service time (see Section 2.2), it is likely that the number of unoccupied unicast channels in the cell, and consequently $P_{success}$, also change in a random fashion;

time-decreasing change in $P_{reattempt}$ - based on the knowledge of common user behaviour, it seems reasonable to expect that the longer a user waits for service, the more likely he will withdraw from the waiting procedure the next time he needs to decide (see Section 2.2) Hence, $P_{reattempt}$ should be modelled as a time-decreasing value, e.g. as depicted in Figure 2.7.

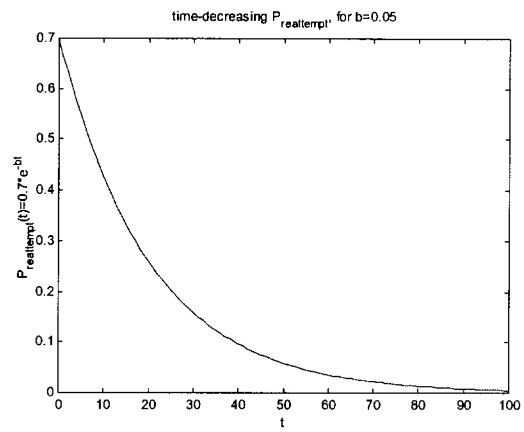


Figure 2.7 Time-decreasing $P_{reattempt}$

It is important to note that even for $P_{reattempt}$=const, the overall probability that a user remains waiting for a requested item decreases, as the user's overall waiting time increases. This can be easily verified by examining the first derivative of (2.15), as shown in (2.18).

$$\frac{\partial P(\text{user listens to broadcast channel at } t = t_y + \tau) \mid t_a = t_y)}{\partial \tau} = \tag{2.18}$$
$$= -\lambda^*(1-a)e^{-\lambda^*(1-a)\tau} < 0$$

Nevertheless, for the reasons discussed in the preceding paragraph, it is reasonable to assume that $P_{reattempt}$, alone, is also a time-decreasing value.

The mathematical framework presented in Section 2.3.1.1 can be easily extended to deal with the case of a time decreasing $P_{reattempt}$. In particular, if we assume that $P_{reattempt}$ changes according to (2.19) (see Figure 2.7),

$$P_{reattempt}(t) = P_{reattempt} \cdot e^{-bt}, \quad b > 0 \tag{2.19}$$

then the MC user behaviour model in Figure 2.6 gets transformed to the model shown in Figure 2.8, while the corresponding generator matrix gets transformed to (2.20).

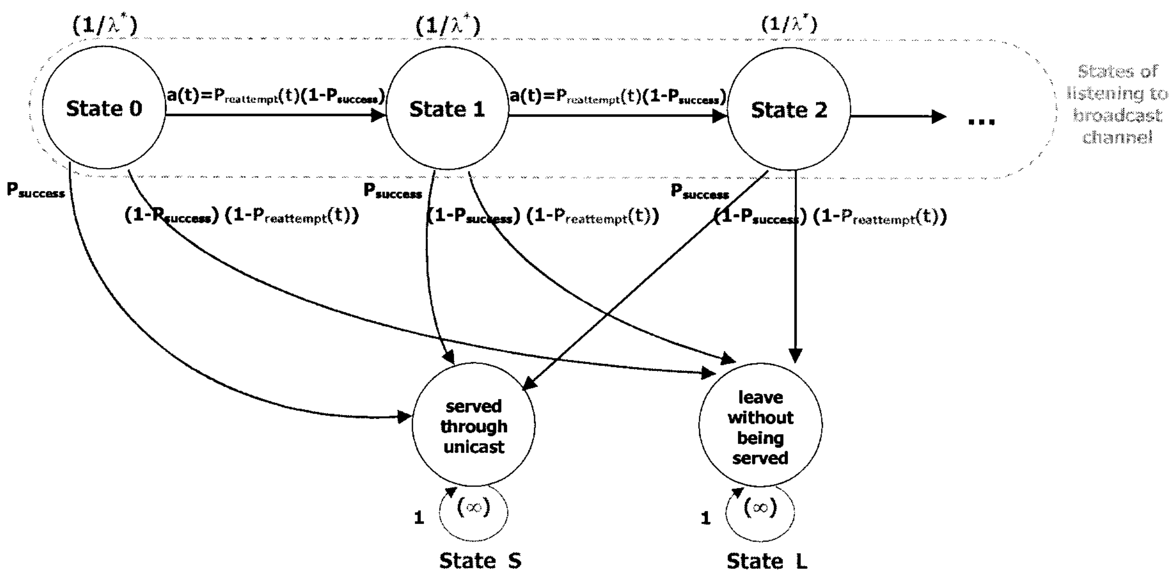


Figure 2.8 User behaviour under the assumption of time-decreasing $P_{reattempt}$

$$
Q = \begin{array}{c}
\begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & \cdots & S & & L \end{array} \\
\left[\begin{array}{cccccccc}
-\lambda^* & \lambda^* a(t) & 0 & 0 & 0 & \cdots & \lambda^* P_{success} & \lambda^*(1-P_{success})(1-P_{reattempt}(t)) \\
0 & -\lambda^* & \lambda^* a(t) & 0 & 0 & \cdots & \lambda^* P_{success} & \lambda^*(1-P_{success})(1-P_{reattempt}(t)) \\
0 & 0 & -\lambda^* & \lambda^* a(t) & 0 & \cdots & \lambda^* P_{success} & \lambda^*(1-P_{success})(1-P_{reattempt}(t)) \\
\cdots & & & & & & & \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0
\end{array}\right]
\end{array}
\begin{array}{|c} 0 \\ 1 \\ 2 \\ \\ S \\ L \end{array}
\qquad (2.20)
$$

Subsequently, by solving a set of differential equations analogous to (2.7), under the same initial conditions (2.21)

$$
P_k(0) = \begin{cases} 1, & k = 0 \\ 0, & k = 1,2,\ldots,S,L \end{cases} \qquad (2.21)
$$

the main transient state probabilities of the new model in Figure 2.8 are obtained, as given in (2.22), (2.23), and (2.24). (A detailed procedure for obtaining (2.22) to (2.24) is presented in Appendix 4).

$$\text{P(user listens to broadcast channel at } t=(t_y+\tau)\,|\,t_a=t_y) = e^{-\lambda^*\left(\tau-\frac{a}{b}\left(1-e^{-b\tau}\right)\right)} \tag{2.22}$$

$$P_S(\tau)=\frac{P_{success}}{a}\left(e^{\lambda^*\frac{a}{b}}\right)\left(\frac{b}{\lambda^* a}\right)^{\frac{\lambda^*}{b}-1}\left[\Gamma\left(\frac{\lambda^*}{b},\lambda^*\frac{a}{b}\right)-\Gamma\left(\frac{\lambda^*}{b},\lambda^*\frac{a}{b}e^{-b\tau}\right)\right] \tag{2.23}$$

$$P_L(\tau)=1-e^{-\lambda^*\left(\tau-\frac{a}{b}\left(1-e^{-b\tau}\right)\right)}-\frac{P_{success}}{a}\left(e^{\lambda^*\frac{a}{b}}\right)\left(\frac{b}{\lambda^* a}\right)^{\frac{\lambda^*}{b}-1}\left[\Gamma\left(\frac{\lambda^*}{b},\lambda^*\frac{a}{b}\right)-\Gamma\left(\frac{\lambda^*}{b},\lambda^*\frac{a}{b}e^{-b\tau}\right)\right] \tag{2.24}$$

$\Gamma(a,x)$ in (2.23) and (2.24) represents the so-called *incomplete Gamma function* [*Stone*]. The exact definition of this function is given in (2.25).

$$\Gamma(a,x)=\int_0^x t^{a-1}e^{-t}dt \approx x^a e^{-x}\sum_{n=0}^{\infty}\frac{x^n}{a(a+1)(a+2)..(a+n)} \tag{2.25}$$

Now, it should be noted that no explicit formula (in terms of standard mathematical functions), for the values of $\Gamma(a,x)$ ($a,x \in R$) exist, and accordingly (2.23) and (2.24) cannot be simplified any further, i.e. no closed-form expressions for $P_S(\tau)$ and $P_L(\tau)$ can be found. Hence, based on (2.23) and (2.24), we conclude that the assumption of a time-decreasing $P_{reattempt}$ results in a user behaviour model of considerable analytical complexity. Consequently, it is to expect that the utilization of system-level solutions, derived on the assumption of such a user behaviour model, would require extensive, mostly numerical, computation.

Nevertheless, our simulation results, presented in Section 2.6.3, suggest that the analytical and computational complexity associated with the assumption of time decreasing $P_{reattempt}$ could be avoided altogether. Namely, although the presence of time decreasing $P_{reattempt}$, in a real-world WDB-system, is proven to affect the overall system performance, the theoretical analysis and solutions based on the assumption of a constant yet adequately chosen $P_{reattempt}$, such that $P_{reattempt} < \max(P_{reattempt}(t))$[9], still can ensure satisfactory system performance.

Similarly, the simulation results of Sections 2.6.1 and 2.6.2 confirm that, in reality, $P_{success}$ indeed changes in a random manner. However, as in the case of $P_{reattempt}$, the theoretical analysis and proposed solutions based on the assumption of an adequate but constant $P_{success}$ again provide satisfactory performance.

2.3.2 Broadcast Service Ratio (BR)

The following are the exact definition of all random variables, and the corresponding annotation, employed in this and the proceeding sections.

1. ***MU arrival time*** **(t_a)**

- Interpretation: r.v. related to the time of a given MU's arrival in the system
- event space: S_{ta}={MU arrives at x [time units] | $x \in R^+$}
- definition: $t_a: S_{ta} \mapsto R^+$, e.g. t_a(MU arrives at x [time units])=x

2. ***MU departure time*** **(t_d)**

- interpretation: r.v. related to the time of a given MU's departure from the system due to 1) impatience (before obtaining the requested item), or 2) after obtaining the requested item through a unicast connection
- event space: S_{td}={MU leaves at x [time units] | $x \in R^+$}
- definition: $t_d: S_{td} \mapsto R^+$, e.g. t_d(MU leaves at x [time units])=x

[9] Here, $\max(P_{reattempt}(t))$ annotates the maximum value that $P_{reattempt}(t)$ takes. In the case of a time-decreasing $P_{reattempt}(t)$, as discussed subsequently, $\max(P_{reattempt}(t))=P_{reattempt}(0)$.

3. ***requested item*** **(item)**

- interpretation: r.v. related to the information item that a given MU wants to retrieve while being present in the system
- event space: S_{item}={MU requests item(i) | i=1,..,M}
- definition: item: $S_{item} \mapsto \{1,2,..,M\}$, e.g. item(MU requests item(i))=i

4. ***service mode*** **(sm)**

- interpretation: r.v. related to the actual service provided to a given MU
- event space: S_{sm}={MU receives no service, MU receives broadcast service, MU receives unicast service}
- definition: sm: $S_{sm} \mapsto \{0,1,2\}$, e.g. sm(MU receives no service)=0, sm(MU receives broadcast service)=1, sm(MU receives unicast service)=2

5. ***broadcast time of item(i), i=1,..,M,*** **(t_{br}(i))**

- interpretation: r.v. related to the actual time of item(i)'s appearance on the broadcast channel next
- event space: $S_{t_{br}(i)}$={item(i) appears on the broadcast channel at x [time units] | $x \in R^+$}
- definition: $t_{br}(i)$: $S_{t_{br}(i)} \mapsto R^+$, e.g. $t_{br}(i)$(item(i) broadcasted at x [time units])=x

6. ***attempts count*** **(att)**

- interpretation: r.v. related to the number of attempts that a MU makes, or is capable of making, in an interval of fixed duration Δt, in order to seize a unicast connection
- event space: S_{att}={MU makes i attempts in Δt time units | i=1,..,∞}
- definition: att: $S_{att} \mapsto N^+$, e.g. att(MU makes i attempts in Δt time units)=i

Definition 2.1 Broadcast Service Ratio (**BR**) is a real number BR$\in$[0,1] which represents the expected fraction of users who arrive in the system[10] and get served through the broadcast channel within an interval of length C$\in$R [time units], where C satisfies Assumption 2.1 (see below).

Assumption 2.1 For $\forall$ s(i)$\in R^+$, $\exists$ k(i)$\in N^+$, such that C=k(i)·s(i), (i=1,..,M).

In practical terms k(i)=C/s(i) represents the number of times that item(i) gets broadcasted within C time units. In [*Jiang_1*] C is referred to as *the duration of an entire broadcast schedule.*

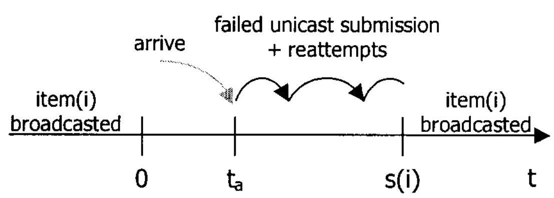


Figure 2.9 Data retrieval through broadcast channel

In order to obtain the exact expression for BR, first let us observe a single user, who arrives in the system at $t_a=\tau\in[0,s(i))$ and wants to retrieve item(i) (i=1,..,M). According to the discussion from Section 2.3.1, and (2.16) in particular, the probability that this user happens to be listening to the broadcast channel, at the time of item(i)'s broadcast ($t_{br}(i)=s(i)$) (see Figure 2.9), corresponds to (2.26).

$$P(\text{user listens to broad. ch. at } s(i) \mid t_a = \tau, \text{item} = i) = e^{-\lambda^*(1-a)(s(i)-\tau)} \qquad (2.26)$$

[10] Here, and in the other definitions, we assume a system of the generalized WDB type, as defined in Section 2.2.1.

Please recall, according to the system and user behaviour assumptions from Section 2.2, (2.26) in fact represents the probability that the given user gets served from the broadcast channel. Hence, by employing the notation from the preceding page, (2.26) can be rewritten as

$$P(sm = 1 \mid t_a = \tau, item = i, t_{br}(i) = s(i)) = e^{-\lambda^*(1-a)(s(i)-\tau)} \tag{2.27}$$

Now, let us annotated the number of users who arrive in the system in an interval $[\tau_1,\tau_2)$ and eventually obtain item(i) through the broadcast channel at time s(i) with $n_{BR}(\tau_1,\tau_2,item(i),s(i))$. On average, $\lambda(i)\Delta\tau$ users interested in retrieving item(i) arrive in the system within the interval $[\tau,\tau+\Delta\tau)$, where $\Delta\tau$ can be of any arbitrary duration. If $\Delta\tau$ is chosen sufficiently small so that (2.27) can be assumed constant for every $t_a \in [\tau,\tau+\Delta\tau)$, then the expected number of MUs who arrive within $[\tau,\tau+\Delta\tau)$ and get served through broadcast at time s(i) can be expressed as

$$E[n_{BR}(\tau,\tau+\Delta\tau,item(i),s(i))] = \lambda(i)\cdot\Delta\tau\cdot P(sm = 1 \mid t_a = \tau, item = i, t_{br}(i) = s(i)) \tag{2.28}$$

Furthermore, if we partition the whole $[0,s(i))$ into $\lceil s(i)/\Delta\tau \rceil$ such non-overlapping intervals, so that the following holds,

$$[0,s(i)) = \bigcup_{n=1}^{\lceil s(i)/\Delta\tau \rceil} [\tau_n, \tau_n + \Delta\tau) \tag{2.29}$$

then based on 'independent and stationary increments' property of the assumed Poisson arrival process [*Leon-Garcia*], the expected number of users served through the broadcast channel at time s(i), regardless of their arrival times in [0,s(i)), can be obtained from

$$\begin{aligned} &E[n_{BR}(0,s(i),item(i),s(i))] = \\ &= \lim_{\Delta\tau\to 0} \sum_{n=1}^{\lceil s(i)/\Delta\tau \rceil} E[n_{BR}(\tau_n,\tau_n+\Delta\tau,item(i),s(i))] = \\ &= \lim_{\Delta\tau\to 0} \sum_{n=1}^{\lceil s(i)/\Delta\tau \rceil} \lambda(i)\cdot e^{-\lambda^*(1-a)(s(i)-\tau)}\cdot\Delta\tau = \int_0^{s(i)} \lambda(i)\cdot e^{-\lambda^*(1-a)(s(i)-\tau)} d\tau \end{aligned} \tag{2.30}$$

Hence,

$$E[n_{BR}(0,s(i),item(i),s(i))]=\frac{\lambda p(i)}{\lambda^{*}(1-a)}(1-e^{-\lambda^{*}(1-a)s(i)}) \tag{2.31}$$

Based on Assumption 2.1, k(i) intervals of duration s(i) are found in the entire broadcast schedule (C). As there are no outstanding requests for item(i) transferred from one interval to another (Section 2.2.1), $E[n_{BR}(0,s(i)\,|\,s(i))]$ is the same for all intervals. Thus, the expected number of users who obtain item(i) from the broadcast channel within C is

$$\begin{aligned}&E[n_{BR}(0,C,item(i),j\cdot s(i)\,|\,j=1,..,k(i))]=\\&=k(i)\cdot E[n_{BR}(0,s(i),item(i),s(i))]=\\&=\frac{C}{s(i)}\frac{\lambda p(i)}{\lambda^{*}(1-a)}(1-e^{-\lambda^{*}(1-a)s(i)})\end{aligned} \tag{2.32}$$

Accordingly, the expected number of users served through the broadcast channel with either of the M items is

$$\begin{aligned}&E[n_{BR}(0,C,item(i),j\cdot s(i)\,|\,j=1,..,k(i),i=1,..,M)]=\\&=\sum_{i=1}^{M}E[n_{BR}(0,C,item(i),j\cdot s(i)\,|\,j=1,..,k(i))]=\\&=\frac{C\lambda}{\lambda^{*}(1-a)}\sum_{i=1}^{M}\frac{p(i)}{s(i)}(1-e^{-\lambda^{*}(1-a)s(i)})\end{aligned} \tag{2.33}$$

Finally, considering that $C\lambda$ users are expected to arrive in the system within the entire C, and $E[n(sm=1,t_a\in[0,C))]$ of them are expected to be served through the broadcast channel, then

$$\begin{aligned}&BR=\frac{E[n_{BR}(0,C,item(i),j\cdot s(i)\,|\,j=1,..,k(i),i=1,..,M)]}{C\lambda}=\\&=\frac{1}{\lambda^{*}(1-a)}\sum_{i=1}^{M}\frac{p(i)}{s(i)}(1-e^{-\lambda^{*}(1-a)s(i)})\end{aligned} \tag{2.34}$$

2.3.3 Unicast Service Ratio (UR)

Definition 2.2 Unicast Service Ratio (**UR**) is a real number $UR \in [0,1]$, which represents the expected fraction of users who arrive in the system and get served through on-demand connections within an interval of length $C \in R$ [time units], where C satisfies Assumption 2.1.

In this section, we omit the detailed explanation regarding how certain expressions have been derived. Instead, we only provide the final results. However, based on the results of Section 2.3.1, and by following a procedure similar to the one presented in Section 2.3.2, the validity of the provided results can easily be confirmed.

The probability that a MU, which arrives in the system at time $t_a = \tau \in [0, s(i))$ and requires to retrieve item(i), seizes one of the point-to-point channels and gets served on-demand before time s(i) is given in (2.35). For illustration see Figure 2.10. (Please note, similar as in Section 2.3.2, (2.35) is readily obtained from (2.17).)

$$P(t_d < s(i) \cap sm = 2 \mid t_a = \tau, item = i, t_{br}(i) = s(i)) = \frac{P_{success}}{1-a}\left[1 - e^{-\lambda^*(1-a)(s(i)-\tau)}\right] \qquad (2.35)$$

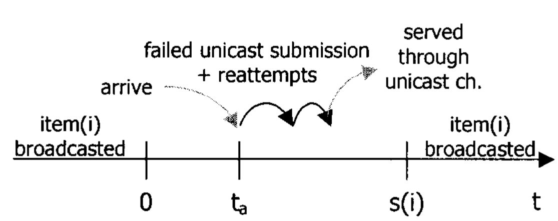


Figure 2.10 Data retrieval through on-demand connection

The expected number of MUs who arrive within $[\tau, \tau+\Delta\tau)$, and obtain item(i) through on-demand connections before time s(i), is given by (2.36) (we annotate this value with

$n_{UR}(\tau,\tau+\Delta\tau,item(i),s(i))$, while (2.37) represents the expected number of users served on-demand regardless of their actual arrival times within [0,s(i)).

$$E[n_{UR}(\tau,\tau+\Delta\tau,\text{item}(i),s(i))] = \lambda(i)\Delta\tau\cdot\frac{P_{success}}{1-a}\left[1-e^{-\lambda^*(1-a)(s(i)-\tau)}\right] \quad (2.36)$$

$$E[n_{UR}(0,s(i),\text{item}(i),s(i))] = \frac{P_{success}}{1-a}\left[\lambda(i)s(i)-\frac{\lambda(i)}{\lambda^*(1-a)}e^{-\lambda^*(1-a)s(i)}\right] \quad (2.37)$$

(2.38) corresponds to the expected number of MUs who obtain item(i) on-demand within the entire C. (2.39) represents the expected number of users who obtain either of the M items within the same interval.

$$E[n_{UR}(0,C,\text{item}(i),j\cdot s(i)\mid j=1,..,k(i))] = \frac{C}{s(i)}\frac{P_{success}}{1-a}\left[\lambda(i)s(i)-\frac{\lambda(i)}{\lambda^*(1-a)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)\right] \quad (2.38)$$

$$E[n_{UR}(0,C,\text{item}(i),j\cdot s(i)\mid j=1,..,k(i);i=1,..,M)] = \sum_{i=1}^{M}\frac{C}{s(i)}\frac{P_{success}}{1-a}\left[\lambda(i)s(i)-\frac{\lambda(i)}{\lambda^*(1-a)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)\right] \quad (2.39)$$

Finally, the expression for UR is provided in (2.40).

$$UR = \frac{E[n_{UR}(0,C,\text{item}(i),j\cdot s(i)\mid j=1,..,k(i);i=1,..,M))]}{C\lambda} = \quad (2.40)$$
$$= \frac{P_{success}}{1-a}\left[1-\frac{1}{\lambda^*(1-a)}\sum_{i=1}^{M}\frac{p(i)}{s(i)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)\right]$$

Based on (2.34), (2.40) can be transformed to

$$UR = \frac{P_{success}}{1-a}[1-BR] \quad (2.41)$$

Please note, expression (2.41) indicates that there exists a direct interdependence between UR and BR. In particular, the maximization of BR results in the minimization of UR, and vice versa.

The given analytical expression seems to be in agreement with our intuitive expectation. Namely, by increasing the percentage of users served through broadcasting, fewer users end up requesting and obtaining unicast service, thus ultimately resulting in a smaller UR.

2.3.4 Service Loss Ratio (LR)

Definition 2.3 Service loss Ratio (**LR**) is a real number $LR \in [0,1]$, which represents the expected fraction of users who arrive in the system within an interval of length $C \in R$ [time units], but do not succeed in obtaining the desired information – these users get impatient and leave the system before being served. (Again, C needs to satisfy Assumption 2.1.)

Similar to the preceding section, we omit the detailed explanation regarding how certain expressions have been derived. Only the final results are provided.

The probability that a MU, which arrives in the system at time $t_a = \tau \in [0, s(i))$ and requires to retrieve item(i), leaves the system without being served before time s(i) is given in (2.42). For illustration see Figure 2.11. (Similar as in the case of the two preceding sections, (2.42) is readily obtained from (2.18).)

$$P(t_d < s(i) \cap sm = 0 \mid t_a = \tau, \text{item} = i, t_{br}(i) = s(i)) = \frac{(1 - P_{success})(1 - P_{reattempt})}{1 - a}\left[1 - e^{-\lambda^*(1-a)(s(i)-\tau)}\right] \quad (2.42)$$

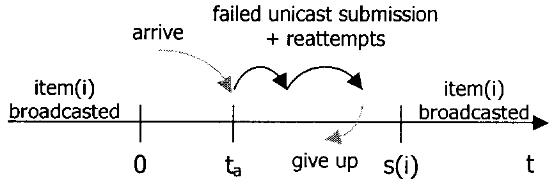


Figure 2.11 MU departure due to impatience within observed interval

The expected number of MUs who arrive within $[\tau,\tau+\Delta\tau)$ and leave the system without being served before time s(i) is given in (2.43) (we annotate this value with $n_{LR}(\tau,\tau+\Delta\tau,item(i),s(i))$, while (2.44) represents the expected number of users who leave the system before time s(i) regardless of their actual arrival times within [0,s(i)).

$$E[n_{LR}(\tau,\tau+\Delta\tau,\text{item}(i),s(i))] = \lambda(i)\Delta\tau\cdot\frac{(1-P_{success})(1-P_{reattempt})}{1-a}\left[1-e^{-\lambda^*(1-a)(s(i)-\tau)}\right] \quad (2.43)$$

$$E[n_{LR}(0,s(i),\text{item}(i),s(i))] = \frac{(1-P_{success})(1-P_{reattempt})}{1-a}\left[\lambda(i)s(i)-\frac{\lambda(i)}{\lambda^*(1-a)}e^{-\lambda^*(1-a)s(i)}\right] \quad (2.44)$$

(2.45) corresponds to the expected number of MUs, interested in item(i), who leave the system without being served within the entire C. (2.46) represents the expected number of users who leave the system without being served within the same interval, regardless of the item they want to obtain.

$$E[n_{LR}(0,C,\text{item}(i),j\cdot s(i)\,|\,j=1,..,k(i))] = \quad (2.45)$$
$$= \frac{C}{s(i)}\frac{(1-P_{success})(1-P_{reattempt})}{1-a}\left[\lambda(i)s(i)-\frac{\lambda(i)}{\lambda^*(1-a)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)\right]$$

$$E[n_{LR}(0,C,\text{item}(i),j\cdot s(i)\,|\,j=1,..,k(i);\,i=1,..,M)] = \quad (2.46)$$
$$= \sum_{i=1}^{M}\frac{C}{s(i)}\frac{(1-P_{success})(1-P_{reattempt})}{1-a}\left[\lambda p(i)s(i)-\frac{\lambda p(i)}{\lambda^*(1-a)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)\right]$$

Accordingly, the expression for LR is provided in (2.47).

$$LR = \frac{E[n_{LR}(0,C,\text{item}(i),j\cdot s(i)\,|\,j=1,..,k(i);\,i=1,..,M)]}{C\lambda} = \quad (2.47)$$
$$= \frac{(1-P_{success})(1-P_{reattempt})}{1-a}\left[1-\frac{1}{\lambda^*(1-a)}\sum_{i=1}^{M}\frac{p(i)}{s(i)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)\right]$$

Finally, based on (2.47), (2.48) can be transformed to

$$LR = \frac{(1 - P_{success})(1 - P_{reattempt})}{1 - a}\left[1 - BR\right] \qquad (2.48)$$

Similarly as in the case of (2.41), (2.48) indicates that there exists a direct relationship between LR and BR. Hence, the maximization of BR results in the minimization of UR, and vice versa. Again, as in the case of (2.41), this result appears quite intuitive. Namely, it seems reasonable to expect that an increase the efficiency of broadcast service, i.e. an increase in BR, ensures a decrease in the fraction of users which leave the system without obtaining any service.

2.3.5 Average Waiting Time (WT)

Definition 2.4 Average Waiting Time (**WT**[11]) represents the average time that a randomly arrived MU spends in the system before either of the following happens: 1) the MU gets served through the broadcast channel, 2) the MU gets served through a unicast connection, 3) the MU leaves without being served.

First, based on Definition 2.4, let us define the actual waiting time (experienced by a MU who arrives at time t_a, wants to obtain item(i), and leaves/gets served at time t_d) as a function of the corresponding random variables (t_a, t_d and sm).

$$wt_i(t_a, t_d, sm) = \begin{cases} s_i - t_a, & sm = 1 \\ t_d - t_a, & sm = 0, 2 \end{cases} \qquad (2.49)$$

An illustration for (2.49) is provided in Figure 2.12.

[11] In a single-queue system, WT corresponds to the average time spent in the queue.

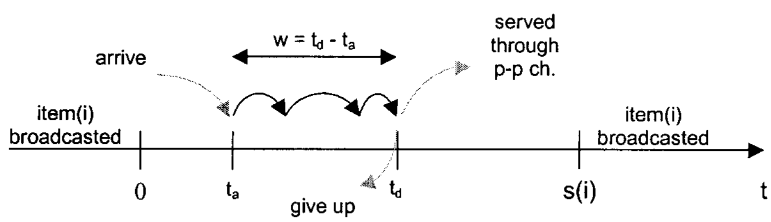


Figure 2.12 Waiting time experienced by MU who gets served through unicast or leaves without service

Accordingly, the average value of $wt_i(t_a,t_d,sm)$ can be obtained from the following expression

$$E[wt_i(t_a,t_d,sm)] = \sum_{j=0}^{2} \int_0^{s(i)} \int_{t_a}^{\infty} wt_i(t_a,t_d,sm=j)p(t_a,t_d,sm=j \mid item=i)dt_a dt_d \tag{2.50.a}$$

(2.50.a), with the right-side expression given in the extended form in (2.50.b).

$$E[wt_i(t_a,t_d,sm)] = \tag{2.50.b}$$
$$= \sum_{j=0}^{2} \int_0^{s(i)} \int_{t_a}^{\infty} wt_i(t_a,t_d,sm=j) \cdot p(t_a \mid item=i) \cdot p(t_d \mid t_a, item=i) \cdot p(sm=j \mid t_a,t_d,item=i) dt_a dt_d$$

Please note, each of the elements of the above summation can also be written as shown in (2.51).

$$\int_0^{s(i)} \int_{t_a}^{\infty} wt_i(t_a,t_d,sm=j)p(t_a,t_d,sm=j \mid item=i)dt_a dt_d = \tag{2.50.c}$$
$$= p(sm=j) \int_0^{s(i)} \int_{t_a}^{\infty} wt_i(t_a,t_d,sm=j)p(t_a,t_d \mid sm=j, item=i)dt_a dt_d =$$
$$= p(sm=j)E[wt_i(t_a,t_d,sm=j)]$$

Hence, based on (2.50.c), each elements of (2.50.a) represents the product of the probability that a MU, who wants to retrieve item$_i$, obtains service sm (sm=0,1, or 2) and the expected waiting time associated with the given service.

The actual expressions for each of the elements figuring in (2.50.a) are given in (2.51). (A detailed procedure for obtaining these expressions is provided in Appendix 5.)

$$\int_0^{s(i)}\int_{\tau}^{\infty} wt_i(t_a=\tau, t_d=\mu, sm=1)p(t_a=\tau, t_d=\mu, sm=1 \mid item=i)d\tau d\mu = \qquad (2.51.a)$$
$$=\frac{1}{s(i)\left(\lambda^*(1-a)\right)^2}\left[1-\left(1+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]$$

$$\int_0^{s(i)}\int_{\tau}^{\infty} wt_i(t_a=\tau, t_d=\mu, sm=2)p(t_a=\tau, t_d=\mu, sm=2 \mid item=i)d\tau d\mu = \qquad (2.51.b)$$
$$=\frac{P_{success}}{\left(\lambda^*\right)^2(1-a)^3 s(i)}\left[-2+\lambda^*(1-a)s(i)+\left(2+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]$$

$$\int_0^{s(i)}\int_{\tau}^{\infty} wt_i(t_a=\tau, t_d=\mu, sm=0)p(t_a=\tau, t_d=\mu, sm=0 \mid item=i)d\tau d\mu = \qquad (2.51.c)$$
$$=\frac{(1-P_{success})(1-P_{reattempt})}{\left(\lambda^*\right)^2(1-a)^3 s(i)}\left[-2+\lambda^*(1-a)s(i)+\left(2+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]$$

By employing (2.51.a), (2.51.b), (2.51.c) back in (2.50), (2.52) is obtained

$$\begin{aligned}
&E[wt_i(t_a, t_d, sm)] = \qquad (2.52)\\
&=\frac{1}{s(i)\left(\lambda^*(1-a)\right)^2}\left[1-\left(1+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]+\\
&\quad+\frac{1}{s(i)\left(\lambda^*(1-a)\right)^2}\left[-2+\lambda^*(1-a)s(i)+\left(2+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]=\\
&=\frac{1}{\lambda^*(1-a)}\left(1-\frac{1}{\lambda^*(1-a)s(i)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)\right)
\end{aligned}$$

According to the assumption from Section 2.2.2, the probability that a randomly arrived MU wants to retrieve item(i), and consequently experiences $E[wt_i(t_a,t_d,sm)]$, is p(i) (i=1,..,M). Hence,

$$WT = \sum_{i=1}^{M} p(i) \cdot E[wt_i(t_a, t_d, sm)] = \tag{2.53}$$
$$= \frac{1}{\lambda^*(1-a)}\left(1 - \frac{1}{\lambda^*(1-a)}\sum \frac{p(i)}{s(i)}\left(1 - e^{-\lambda^*(1-a)s(i)}\right)\right)$$

Based on (2.34), (2.53) can be finally transformed to

$$WT = \frac{1}{\lambda^*(1-a)}(1 - BR) \tag{2.54}$$

Hence, (2.54) indicates that the minimization of the average waiting time (WT) can be expected provided BR gets maximized. Similar as in the case of (2.41) and (2.48), (2.54) appears rather intuitive, for two main reasons. On one side, it is clear that an increase in the efficiency of broadcast service (BR) leads to a decrease in the average broadcast service waiting time. Nevertheless, based on (2.41), an increase in BR also results in a decrease in the number of users that need to be served through point-to-point connections, thus leading to a decrease in the average unicast service waiting time.

2.3.6 Reneging Probability

Definition 2.5 Reneging Probability (**RP**) represents the probability that a user, who has joined the system, eventually leaves before being served, i.e. without obtaining the requested information item.

The probability that a MU requests item(i) (p(i)) is assumed independent of the probability that he requests item(j) (p(j)), where $i,j \in \{1,..,M\}$ and $i \neq j$. Accordingly, RP can be expressed as a sum of independent joint probabilities that the MU wants to retrieve item(i) ($i \in \{1,..,M\}$) and that he leaves the system before obtaining the item. By employing the corresponding conditional probabilities, the given sum can be further extended as presented in (2.55).

$$\begin{aligned}
RP &= \sum_{i=1}^{M} p(\text{item} = i \cap sm = 0) = \\
&= \sum_{i=1}^{M} p(\text{item} = i) \cdot P(sm = 0 \mid \text{item} = i) = \\
&= \sum_{i=1}^{M} p(i) \cdot P(sm = 0 \mid \text{item} = i)
\end{aligned} \tag{2.55}$$

The conditional probability that a MU departs from the system without being served, given he wants to retrieve item(i), can be obtained through the mathematical procedure presented in (2.56). (Please note, as in Section 2.3.5, here we assume a randomly arrived user, hence $p(t_a \cap sm=0|item=i)=1/s(i)$.)

$$\begin{aligned}
&P(sm = 0 \mid \text{item} = i) = \\
&= \int_0^{s(i)} \int_{t_a}^{s(i)} p(t_a \cap t_d \cap sm = 0 \mid \text{item} = i) dt_a dt_d = \\
&= \int_0^{s(i)} \int_{t_a}^{s(i)} p(t_a \mid \text{item} = i) \cdot p(t_d \cap sm = 0 \mid t_a, \text{item} = i) dt_a dt_d = \\
&= \frac{1}{s(i)} \int_0^{s(i)} dt_a \int_{t_a}^{s(i)} p(t_d \cap sm = 0 \mid t_a, \text{item} = i) dt_d = \\
&= \frac{1}{s(i)} \int_0^{s(i)} dt_a P(t_d < s(i) \cap sm = 0 \mid t_a, \text{item} = i) = \\
&= \frac{1}{s(i)} \int_0^{s(i)} \frac{(1 - P_{success})(1 - P_{reattempt})}{1 - a} \left[1 - e^{-\lambda^*(1-a)(s(i)-t_a)}\right] dt_a = \\
&= \frac{(1 - P_{success})(1 - P_{reattempt})}{1 - a} \left[1 - \frac{1}{\lambda^*(1 - a)s(i)} \left(1 + e^{-\lambda^*(1-a)s(i)}\right)\right]
\end{aligned} \tag{2.56}$$

Based on (2.56), (2.55) now becomes

$$\mathrm{RP}=\sum_{i=1}^{M}p(i)\cdot\frac{(1-P_{success})(1-P_{reattempt})}{1-a}\left[1-\frac{1}{\lambda^{*}(1-a)s(i)}\left(1+e^{-\lambda^{*}(1-a)s(i)}\right)\right]=$$
$$=\frac{(1-P_{success})(1-P_{reattempt})}{1-a}\left[1-\sum_{i=1}^{M}\frac{p(i)}{\lambda^{*}(1-a)s(i)}\left(1+e^{-\lambda^{*}(1-a)s(i)}\right)\right] \tag{2.57}$$

Finally, by employing (2.34), RP can be expressed as

$$\mathrm{RP}=\frac{(1-P_{success})(1-P_{reattempt})}{1-a}(1-\mathrm{BR}) \tag{2.58}$$

Please note that the final expression for RP (2.58) turns out to be identical to the expression for LR (2.48). Also, as in the case of Section 2.3.4, (2.58) indicates that maximization of BR results in the minimization of RP, and vice versa.

2.3.7 Waiting Users Population (WP)

Definition 2.6 Waiting User Population (**WP**[12]) represents the expected number of waiting users[13] in the system, encountered by a new randomly arrived MU. WP is defined over an interval of fixed duration C∈R [time units], where C is assumed to satisfy Assumption 2.1.

Based on the discussion from Section 2.3.2, and (2.31) in particular, it can be easily seen that the expected number of users waiting to obtain item(i) at some observation time $t_d \in [0,s(i))$, regardless of their actual arrival times within $[0, t_d)$, is

[12] In a single-queue system, WP would correspond to the expected queue length.

[13] 'Waiting user' is a MU which happens to be in the waiting phase of the retrieval cycle (Figure 2).

$$wp(t_d, item(i)) = E[n_{BR}(0, t_d, item(i), t_d)] = \frac{\lambda p(i)}{\lambda^*(1-a)}(1 - e^{-\lambda^*(1-a)t_d}) \tag{2.59}$$

Now, if we assume that a new user joins the system at time t_d, and t_d can be any arbitrary time point between two consecutive broadcastings of item(i), i.e. the probability of any specific $t_d = \mu \in [0, s(i))$ is $1/s(i)$), then the expected number of users waiting to retrieve item(i) encountered by this new MU is

$$E[wp(t_d, item(i))] = \int_0^{s(i)} wp(\mu, item(i)) pdf_{t_d}(\mu) d\mu = \tag{2.60}$$

$$= \int_0^{s(i)} E[n_{BR}(0, \mu, item(i), \mu)] \cdot pdf_{t_d}(\mu) d\mu =$$

$$= \frac{1}{s(i)} \frac{\lambda p(i)}{\lambda^*(1-a)} \left[s(i) - \frac{1}{\lambda^*(1-a)}(1 - e^{-\lambda^*(1-a)s(i)}) \right]$$

Accordingly, the expected number of users waiting to retrieve either of the M items at the time of the new MU's arrival can be obtained from

$$E[wp(t_d, item(i) \mid i = 1,.., M)] = E[\sum_{i=1}^{M} wp(t_d, item(i))] = \tag{2.61}$$

$$= \sum_{i=1}^{M} E[wp(t_d, item(i))] =$$

$$= \frac{\lambda}{\lambda^*(1-a)} \left[1 - \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^*(1-a)s(i)}) \right]$$

As $E[wp(t_d)]$ is in fact WP, then based on (2.34) the following implies

$$WP = \frac{\lambda}{\lambda^*(1-a)}(1 - BR) \tag{2.62}$$

(2.62) shows that by maximizing BR it is possible to minimize WP and vice versa. This result again seems to be in agreement with our intuitive expectation. Namely, it is already proven that an increase in the efficiency of broadcast service (BR) would lead to a decrease in the average user waiting time (WT) (see 2.54). This, together with the well-known Little's formula [*Leon-Garcia*], clearly implies that an increase in BR would also results in a smaller average number of users waiting in the system, i.e. smaller WP.

2.3.8 Discussion

The major goal of any data communication network is to provide sufficiently high levels of throughput[14], quality of service (QoS)[15] and grade of service (GoS)[16] to its users. In the case of a hybrid WDB system with user retrials, the measures defined and studied in Sections 2.3.2 to 2.3.7 can be used to reflect how well the system accomplishes these goals.

(1) system's throughput

The throughput of a hybrid WDB system can be expressed simply as throughput=(BR+UR)=1-LR. By employing (2.48), the given expression gets transformed to

$$\text{throughput} = 1 - \frac{(1-P_{\text{success}})(1-P_{\text{reattempt}})}{1-a}\left[1-\text{BR}\right] \tag{2.63}$$

(2) system's QoS

QoS offered by a network most closely depends on the load that the network is exposed to, i.e. the overall occupancy of the network resources (links, bandwidth, etc.). In that sense, by minimizing the fraction of users who get served through unicast connections (UR), fewer

[14] In this case, we define throughput as the fraction of users who get served, through either the broadcast or a unicast channel.
[15] QoS – set of parameters used to describe a network's performance, including: link delay, packet delay variation, packet error rate, … [*Schiller*].
[16] GoS – the probability of a call or data request being blocked or delayed for more than a specified interval [*Schiller*].

resources end up being used. This, in turn, enables subsequent improvements in the offered QoS and an increase in the network's capacity. Hence, based on the above, inverse proportionality between QoS and UR can be claimed. That is, by employing (2.41), we can write,

$$\mathrm{QoS} \sim \frac{1}{\mathrm{UR}} = \frac{(1-\mathrm{a})}{\mathrm{P}_{\mathrm{success}}\left[1-\mathrm{BR}\right]} \tag{2.64}$$

(3) system's GoS

While QoS describes how efficiently a request is handled once it has been accepted for service, GoS is supposed to reflect the probability that a request gets accepted for service, in the first place. Optionally, GoS may also be an indicator of the service (i.e. waiting) delay.

In a hybrid WDB system, as studied in this work, the probability of a request being accepted for service is (1-RP). Hence,

$$\mathrm{GoS} \sim 1-\mathrm{RP} = 1-\frac{(1-\mathrm{P}_{\mathrm{success}})\left(1-\mathrm{P}_{\mathrm{reattempt}}\right)}{(1-\mathrm{a})}\left[1-\mathrm{BR}\right] \tag{2.65}$$

The service delay, on the other hand, is simply WT. Moreover, it can be expected that the service delay improve, i.e. decrease, as the number of waiting users in the system (WP) decreases[17]. Accordingly, we can claim that GoS is inversely proportional to both, WT and WP.

$$\mathrm{GoS} \sim \frac{1}{\mathrm{WT}} = \lambda^{*}(1-\mathrm{a})\frac{1}{\left[1-\mathrm{BR}\right]} \tag{2.66}$$

$$\mathrm{GoS} \sim \frac{1}{\mathrm{WP}} = \frac{\lambda^{*}(1-\mathrm{a})}{\lambda}\frac{1}{\left[1-\mathrm{BR}\right]} \tag{2.67}$$

[17] Decreasing WP implies fewer waiting, i.e. competing, users in the system, thus the probability of obtaining an on-demand connection, i.e. service, in a shorter interval of time improves.

We started our discussion, back in Section 2.3.2, by assuming a fixed set of broadcast intervals $S=\{s_i| i=1,..,M\}$. (Please note, here and in the proceeding discussion, we employ $S=\{s_i| i=1,..,M\}$ as a mathematical representation of a ***broadcast schedule***, i.e. a ***broadcast scheduling scheme***.) However, broadcast scheduling scheme is entirely controlled by the system[18] and, from the optimization point of view, can be modified so as to meet certain objectives.

The maximization of BR emerges as the most natural objective of a WDB system. However, (2.41), (2.48), (2.54), (2.58), and (2.62) show that the broadcast scheduling scheme which succeeds in maximizing BR, at the same time ensures the minimization with respect to the other performance measures: UR, LR, WT, RP, WP. (Please note, individually these objectives seem reasonable for a WDB system as well.) Most importantly, though, (based on (2.63) through (2.67)), the very same broadcast scheduling scheme which accomplishes all of the above, also meets the major WDB system goals - the provision of improved throughput, QoS and GoS.

[18] Note that all the other parameters ($P_{success}$, $P_{reattempt}$, λ^*, p(i) (i=1,2,..,M), λ) are not directly controlled by the system, and therefore cannot be employed for optimization purpose.

2.4 Special Case Analysis

In this section, we attempt to illustrate the versatility of both, our model of single-cell hybrid WDB system with user retrials (presented in Section 2.2), and the theoretical results obtained employing the given model (as presented in Section 2.3). In particular, we show that through appropriate selection of the main system parameters, the single-cell hybrid WDB model gets easily transformed to the model of a pure-push broadcast, a pure-unicast, or a system as presented in [*Jiang_1*]. Consequently, the values of major parameters concerning the performance of these systems are obtained simply by finding the appropriate mathematical limits of the corresponding expressions derived for the single-cell hybrid WDB model.

2.4.1 Pure-Push Broadcast System

In the pure-push broadcast scenario there is no direct interaction between MUs and the system, thus MUs get service through the broadcast channel exclusively. (For more about pure-push broadcast systems see Chapter 1.)

The generalized WDB model can be transformed into the model of a pure-push broadcast system simply by setting $\lambda^*=0$, i.e. $1/\lambda^* \to \infty$. In practical terms, $1/\lambda^* \to \infty$ implies that MUs have infinite patience (see Figure 2.13), i.e. they never consider either issuing an on-demand request, or leaving the system, before obtaining the desired information from the broadcast channel.

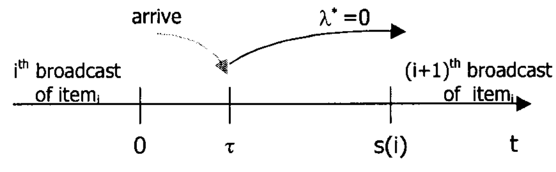


Figure 2.13 Pure-push broadcast system ($\lambda^*=0$)

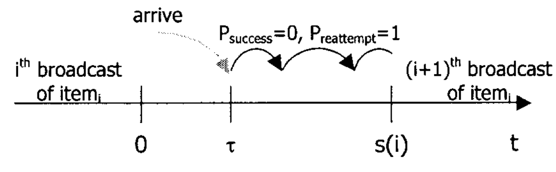


Figure 2.14 Pure-push broadcast system ($P_{success}=0$, $P_{reattempt}=1$)

The model of another type of pure-push broadcast system can be obtained by setting $P_{success}$=0 and $P_{reattempt}$=1, that is (1-a)=0 (see Figure 2.14). In practical terms, $P_{success}$=0 implies one of the following two: 1) MUs never attempt to seize, or 2) MUs never get a chance to seize one of the on-demand connections. $P_{reattempt}$=1, on the other hand, implies that regardless of possibly unsuccessful attempts to seize point-to-point channels, MUs remain in the system until they eventually obtain the desired information from the broadcast channel.

Now, expression (2.31) (see Section 2.3.2) concerning the expected number of MUs interested in retrieving item(i) that are found waiting in the system at some arbitrary time $t=\mu\in[0,s(i))$, can be rewritten as follows.

$$
\begin{aligned}
&E[n_{BR}(0,s(i),item(i),s(i))] = \frac{\lambda p(i)}{\lambda^*(1-a)}(1-e^{-\lambda^*(1-a)\mu}) \\
&= \frac{\lambda p(i)}{\lambda^*(1-a)}(1-1+\lambda^*(1-a)\mu - \frac{1}{2!}(\lambda^*(1-a)\mu)^2 + \frac{1}{3!}(\lambda^*(1-a)\mu)^3 - ...) \\
&= \lambda p(i)(\mu - \frac{1}{2!}\lambda^*(1-a)(\mu)^2 + \frac{1}{3!}(\lambda^*(1-a))^2(\mu)^3 - ...)
\end{aligned}
\tag{2.68}
$$

By setting λ^*=0, or a=1, (2.68) becomes

$$
\begin{aligned}
&\lim_{\lambda^*\to 0} E[n_{BR}(0,s(i),item(i),s(i))] = \\
&= \lim_{a\to 1} E[n_{BR}(0,s(i),item(i),s(i))] = \\
&= \lambda p(i)\tau = \lambda(i)\mu
\end{aligned}
\tag{2.69}
$$

As it could have been anticipated, (2.69) suggests that μ seconds after the last broadcasting of item(i) in a pure-push broadcast system, the expected number of new users waiting to obtain this item (next time it gets broadcasted) is $\lambda(i)\mu$.

In a similar fashion, simply by finding the appropriate limits of the expressions presented in the Section 2.3 (i.e. by finding the limits at $\lambda^*\to 0$, or $a\to 1$), the values of other pure-push broadcast system parameters are obtained, as given in Table 2.1.

Parameter	Value
BR	1
UR	0
LR	0
WT	$\sum_{i=1}^{M} \frac{p(i)s(i)}{2}$
RP	0
WP	$\lambda \cdot \sum_{i=1}^{M} \frac{p(i)s(i)}{2} = \lambda \cdot WT$ [19]

Table 2.1 Pure-push broadcast system parameters

It should be observed that in the case of a pure-push broadcast system, broadcast service ratio (BR) is not an appropriate objective function. Namely, according to the assumptions concerning this type of systems, all MUs are served through the broadcast channel by default. Consequently, BR is always one (BR=1), irrespective of the actual broadcast scheduling scheme employed. Therefore, based on Table 2.1, the only appropriate objective function in a pure-push broadcast system would be either waiting time (WT) or waiting user population (WP).

2.4.2 Pure-Unicast System

In contrast to pure-push broadcast, in pure-unicast systems MUs are served through point-to-point connections only.

The generalized WDB model becomes the model of a pure-unicast system by setting $s(i) \to \infty$, $(i=1,..,M)$. $s(i) \to \infty$ implies that item(i) never appears on the broadcast channel, thus a MU can obtain the given item only through one of the on-demand connections (Figure 2.15). Also,

[19] Please note that this expression coincide with the well-known Little's formula [*Gross*], which applies to queues with patient users.

$P_{reattempt}$ should be set to one ($P_{reattempt}$=1), if the possibility of a MU getting impatient and leaving the system is not assumed.

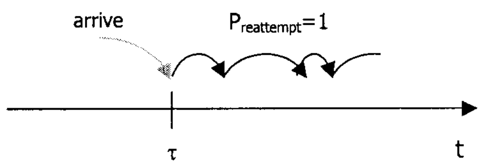


Figure 2.15 Pure-unicast system ($s(i) \rightarrow \infty$)

Hence, in the case of a pure-unicast system expression (2.34), for example, can be rewritten as follows.

$$BR = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^*(1-a)s(i)}) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - \frac{1}{e^{\lambda^*(1-a)s(i)}}) \qquad (2.70)$$

Accordingly, for $s(i) \rightarrow \infty$, i=1,..,M, (2.70) becomes

$$\lim_{\substack{s(i) \rightarrow \infty \\ (i = 1,\ldots,M)}} BR = 0 \qquad (2.71)$$

Again, as it could have been anticipated, (2.71) implies that the expected fraction of MUs served through the broadcast channel (BR) of a pure-unicast system is always zero.

Similarly, by finding the limits at $s(i) \rightarrow \infty$, i=1,..,M, the values of other pure-unicast system parameters can obtained, as given in Table 2.2. However, it should be noted that as a consequence of the assumptions concerning pure-unicast systems, neither of the parameters form Table 2.2 depends on the broadcast scheduling scheme S={s(1),..,s(M)}. Accordingly, neither of them represents an appropriate objective function with respect to S={s(1),..,s(M)}.

Parameter	Value
BR	0
UR	1
LR	0
WT	$\frac{1}{\lambda^{*} P_{success}}$
RP	0
WP	$\frac{\lambda}{\lambda^{*} P_{success}}$

Table 2.2 Pure-unicast system parameters

2.4.3 WDB System with Impatient users as Studied in [Jiang_1]

In [*Jiang_1*] it has been assumed that each newly arrived user waits Δt seconds for the desired information to appear on the broadcast channel, where Δt is an exponentially distributed random variable with $E(\Delta t)=1/\lambda^{*}$. After Δt seconds elapse, the MU leaves the system regardless of whether he obtained the information or not (see Figure 2.16).

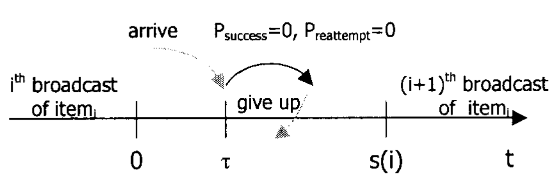


Figure 2.16 Broadcast system from [*Jiang_1*]

The generalized WDB model becomes the model of the system presented in [*Jiang_1*] by setting $P_{success}=0$ and $P_{reattempt}=0$. ($P_{success}=0$ corresponds to the first assumption from [*Jiang_1*], that there is no point-to-point connections available, while $P_{reattempt}=0$ corresponds to the second assumption that MUs leave the system immediately after the first waiting interval.) Thus, by setting both $P_{success}$ and $P_{reattempt}$ to zero, equation (2.34), expressing broadcast service ratio (BR), becomes

$$\lim_{a \to 0} BR = \lim_{a \to 0} \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^*(1-a)s(i)}) = \frac{1}{\lambda^*} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^* s(i)}) \qquad (2.72)$$

By inspection, it can be easily verified that (2.72), and the expression for WT from Table 2.3, entirely match the corresponding expressions from [*Jiang_1*]. The values of other parameters concerning the broadcast system presented in [*Jiang_1*] are given in Table 2.3.

Parameter	Value
BR	$\sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^* s(i)})$
UR	0
LR	$1 - BR$
WT	$\frac{1-BR}{\lambda^*}$
RP	$1 - BR$
WP	$\frac{\lambda}{\lambda^*}(1 - BR)$

Table 2.3 Parameters concerning broadcast system presented in [*Jiang_1*]

2.5 Estimation of Data-item Request Probabilities and 'Invisible User Problem'

The discussion and results presented in the preceding sections have been based on the assumption that the relative popularities of individual data items (p(i), i=1,...,M), in a single-cell hybrid WDB system are known. (Please note, in the analysis of pure-push WDB systems, for example (see Chapter 1), this assumption is not only common, but absolutely required. Namely, due to the lack of up-link communication, pure-push WDB systems do not have the ability to collect any feedback information from the users, including the information on data item popularities.) Nevertheless, in contrast to pure-push, hybrid WDB systems have the potential to autonomously obtain such information through simple monitoring of user request that arrive on unicast connections. Hence, in this section, we take a closer look at the class of single-cell hybrid WDB systems with reattempt users, that perform autonomous estimation of data item popularities, and we discuss the challenges that such systems face.

2.5.1 Single-Cell Hybrid WDB Systems with Autonomous Estimation of Data-Item Popularities

The operation of a hybrid WDB system with autonomous estimation of data item popularities (see Figure 2.17) is based on the following principles. First, by monitoring the users' requests received through unicast connections, the system continually re-evaluates the actual popularities of individual data items. Subsequently, by identifying the most popular items, and placing them on the broadcast channel, the system attempts to maximize the number of users who get served without participating in the request submission procedure (i.e. attempts to maximize BR). Eventually, the maximization of BR, i.e. the minimization of UR, is expected to result in the improvements of the overall system's GoS and QoS, as explained in Section 2.3.

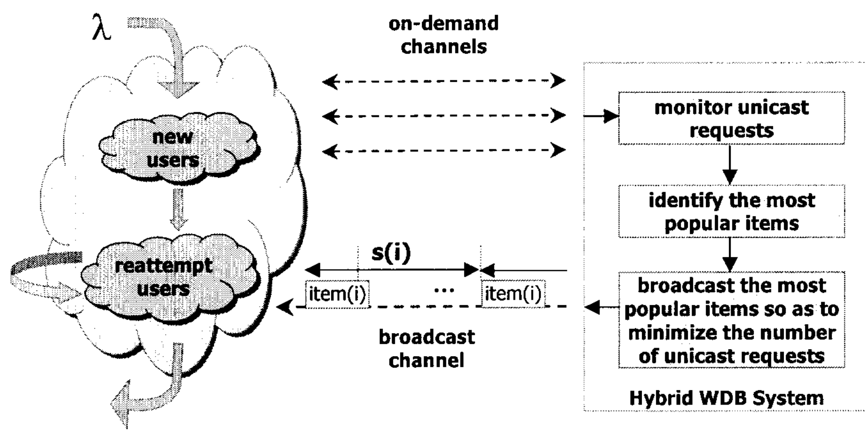


Figure 2.17 Hybrid WDB system with autonomous estimation of data-item popularities

Clearly, for systems as shown in Figure 2.17, the accurate identification of the most popular items is highly critical, as it represents the main precondition for the maximization of the number of users who get served without participating in the request submission procedure. (Please note, we refer to such users as 'invisible', since the system provides them with service without being aware of their presence.) However, it should be observed that the maximization of the invisible user population, although being the ultimate goal, has a negative return effect on the identification of the most popular items. Namely, as the population of the 'invisible' users grows, and the system's performance improves, fewer requests for the most popular items end up being submitted to the system. Consequently, the statistics extracted from the submitted requests becomes less accurate, possibly resulting in a false identification of the most popular items, at the next round of popularity re-evaluation.

Obviously, the phenomenon described in the above paragraph could eventually lead to unstable system performance unless adequately recognized by the system. In the remainder of this section, we provide a thorough insight into the significance and nature of the invisible user problem, as found in hybrid WDB systems with user retrials.

2.5.2 Invisible User Problem

In order to prove the existence of the invisible user problem in a WDB system with user retrials, let us find the expression concerning the number of requests for item(i) (i=1,..,M) received by the system within an arbitrary interval of time $\mathcal{W}=[t_1,t_2)$. (Here, first we will assume that $\mathcal{W}$ is of arbitrary duration, yet satisfies constraint (2.73) as illustrated in Figure 2.18.)

$$\mathcal{W} = [t_1, t_2) \subseteq [0, s(i)) \tag{2.73}$$

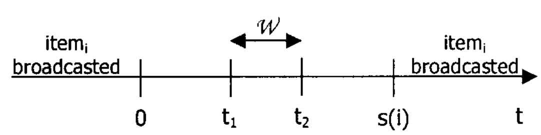


Figure 2.18 Reduced window of observation in estimation of data-item popularities

Through the reuse of (2.31), the expected number of requests for item(i), generated and received by the system within $[0,t_1)$ and $[0,t_2)$, can readily be found, as given in (2.74) and (2.75) respectively.

$$E[n_{UR}(0,t_1,\text{item(i)},t_1)] = \frac{P_{success}\lambda p(i)}{1-a}\left[t_1 - \frac{1}{\lambda^*(1-a)}e^{-\lambda^*(1-a)t_1}\right] \tag{2.74}$$

$$E[n_{UR}(0,t_2,\text{item(i)},t_2)] = \frac{P_{success}\lambda p(i)}{1-a}\left[t_2 - \frac{1}{\lambda^*(1-a)}e^{-\lambda^*(1-a)t_2}\right] \tag{2.75}$$

Consequently, the expected number of requests for item(i), received by the system within the actual $\mathcal{W}=[t_1,t_2)$ (we annotate this value with $E[req(t_1,t_2,item(i))]$), can be simply obtained as a difference between (2.75) and (2.74) (see (2.76)).

$$E[req(t_1, t_2, item(i))] = \tag{2.76}$$
$$= E[n_{UR}(0, t_2, item(i), t_2)] - E[n_{UR}(0, t_1, item(i), t_1)] =$$
$$= \frac{P_{success}\lambda p(i)}{1-a}\left[|\mathcal{W}| - \frac{1}{\lambda^*(1-a)}\left(1 - e^{-\lambda^*(1-a)|\mathcal{W}|}\right)e^{-\lambda^*(1-a)t_1}\right]$$

(In the above expression, $|\mathcal{W}|$ stands for (t_2-t_1), i.e. the length of the observation window $\mathcal{W}=[t_1,t_2)$.)

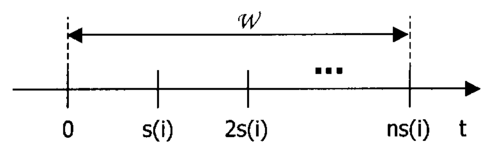


Figure 2.19 Extended window of observation in estimation of data-item popularities

Now, let us relax constraint (2.73 i.e. let us assume that $\mathcal{W}=[t_1,t_2)$ stretches over an arbitrary number of item(i)'s broadcast periods[20] (see Figure 2.19). In particular,

$$\mathcal{W} = [t_1, t_2) = [0, ns(i)) \tag{2.77}$$

where $n \in N^+$. Under such assumption, and by recalling that no users requesting item(i) remain pending in the system once this item gets broadcasted, the expected number of received item(i) requests within $\mathcal{W}=[0,ns(i))$ turns out to be

$$E[req(0, ns(i), item(i))] = \tag{2.78}$$
$$= n \cdot E[req(0, s(i), item(i))] =$$

[20] Here, even further generalization of the constraint is possible, e.g. $\mathcal{W} = [t_1, t_2) = [t_1, ns(i) + \Delta t)$, where $t_1, \Delta t \in [0, s(i))$. However, for the simplicity purposes, we assume that the observation window is aligned with the broadcast schedule ($t_1=0$) and stretches over a finite number of full broadcast periods ($\Delta t=0$).

$$= n \cdot \frac{P_{success} \lambda p_i}{1-a}\left[s(i) - \frac{1}{\lambda^*(1-a)}\left(1 - e^{-\lambda^*(1-a)s(i)}\right)\right]$$

(Please note, E[req(0,s(i),item(i))] in the above expression is obtained from (2.76 using t_1=0 and t_2=s(i), i.e. $|\mathcal{W}|$=s(i).)

By taking a closer look at (2.78) it can be easily observed that the number of received requests for item(i), in a hybrid WDB system with user retrials, is not a straightforward indicator of the item's actual popularity p(i) (i=1,..,M). Namely, although this number is always linearly proportional to p(i), it is also shown to be a non-linear function of other parameters, including:

1) the current system's broadcast schedule policy, i.e. the length of broadcast period associated with item(i) (s(i));
2) the specific nature of users' behaviour, i.e. the actual values of λ^* and $P_{reattempt}$.

Please note, by assuming $P_{reattempt}$=0 and $P_{success}$=1, i.e. by taking the limit of (2.78) at $a \to 0$[21], (2.78) becomes

$$\lim_{a \to 0} E[req(0, ns(i), item(i))] = \frac{n\lambda p(i)}{\lambda^*}\left[\frac{1}{2!}\left(\lambda^* s(i)\right)^2 - \frac{1}{3!}\left(\lambda^* s(i)\right)^3 + \ldots\right] \tag{2.79}$$

Thus, (2.79) proves that the non-linear relationship between E[req(0,ns(i),item(i))] and s(i) also holds for uncongested hybrid WDB systems in which users submit only one request per each data item retrieved.

Figure 2.20 illustrates to what extent the number of received requests can be affected by the actual value of s(i). More specifically, the figure shows how the number of requests, received by the system within an interval of duration $|\mathcal{W}|$ = 64 [time units], changes with respect to both, $P_{success} \in [0,1]$ and s(i)={2,4,8,16,32,64} [time units]. (Here, the other parameters are assumed

---

[21] Recall, $a=(1-P_{success})P_{reattempt}$

constant: p(i)=0.3, λ=5 [users per time unit], λ^*=0.2 [requests per time unit], $P_{reattempt}$=0.7.) In particular, according to Figure 2.20, if an item gets broadcasted (e.g.) every s(i)=16 [time units], then the system receives twice the number of requests for this item than what it would receive if the item was broadcasted every s(i)=4 [time units], regardless of the actual value of $P_{success}$. Put another way, Figure 2.20 verifies that the more frequently an item appears on the broadcast channel (i.e. the smaller s(i) gets), the fewer requests for this item end up arriving to the system.

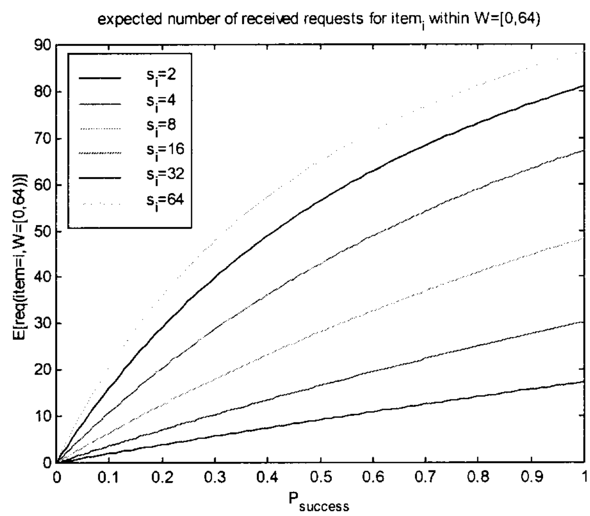


Figure 2.20 Effect of s(i) on number of received requests E[req(0,s(i))|item=i]

Thus, based on both, expression (2.78) and Figure 2.20, we conclude that the estimation of data items' popularities (p(i), i=1,..,M), in a hybrid WDB system with user retrials, should not be exclusively based on the statistics extracted from the requests received through unicast connections. Instead, the estimation of p(i)-s should take into consideration the current values of broadcast periods (s(i), i=1,..,M) and the values of other system and user behaviour parameters, including λ, $P_{success}$, λ^*, and $P_{reattempt}$.

2.6 Simulation-Based Performance Analysis

In this section, we present a series of experimental results obtained by simulating the performance of single-cell hybrid WDB systems with user retrials, as studied in the preceding sections. The significance of these results is twofold. On one side, they provide the full verification and prove the versatility (i.e. robustness) of some of the earlier obtained analytical expressions. On the other hand, they enable us to draw a few significant conclusions, concerning both the performance and analysis of single-cell hybrid WDB systems, which would be impossible to obtain otherwise (i.e. analytically).

The simulation software that we have developed and used in the analysis of single-cell hybrid WDB systems has been entirely implemented in Java. The implementation has been significantly facilitated through the use of DESMO-J – a Java-based discrete-event simulation package [DesmoJ]. A detailed overview of both DESMO-J package and our simulation software is provided in Chapter 7.

Our choice of simulation parameters (λ^*=0.5, $P_{reattempt}$=0.7) was based on [Onur]. The decision to employ the values of λ^* and $P_{reattempt}$ as in [Onur] was justified by the fact that, at the time when we conducted this work, [Onur] represented the only known study on 'user retrial phenomenon', which dealt with real-world statistics. In particular, the values of λ^* and $P_{reattempt}$ presented in [Onur] were estimated by following the behaviour of 45 GSM network subscribers, over certain interval of time.

2.6.1 $P_{success}$ – Stochastic Behaviour

Statement 2.1 In real-world single-cell hybrid WDB systems, $P_{success}$ exhibits stochastic behaviour[19].

The simulation results presented in Sections 2.6.1.1 to 2.6.1.3 provide the proof for Statement 2.1. These results were collected by observing the change in the number of occupied (i.e. seized) unicast channels in three different single-cell hybrid WDB-system simulation settings. Each of the simulation settings was defined with a unique set of data-item request probabilities, as shown in Table 2.4, 2.5, and 2.6, and in each of the settings an overall number of 10, 20 and 30 available unicast channels was employed, respectively. In all the observed cases it was evident that the number of occupied unicast channels was changing in time (see Figure 2.24, 2.27, and 2.30), which indirectly implied that $P_{success}$ was also changing, in an alike manner. Please recall, at any point in time the actual value of $P_{success}$ is proportional to the number of unoccupied unicast channels in the system. Thus, as the number of unoccupied unicast channels increases, $P_{success}$ is expected to increase accordingly, and vice versa.

2.6.1.1 $P_{success}$ – Stochastic Behaviour: Experiment 1

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l./p**
MU reattempt probability	$P_{reattempt}$**=0.7**
MU mean reattempt time [time units]	λ^***=0.5**
Mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.4 Simulation parameters (2.6.1.1)

[19] That is, $P_{success}$ evolves randomly in time.

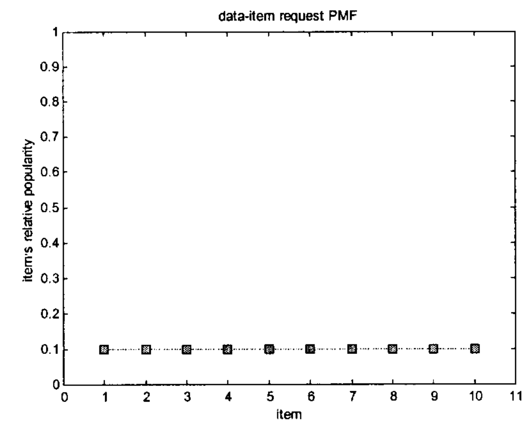


Figure 2.22 Data-item request probabilities (1)

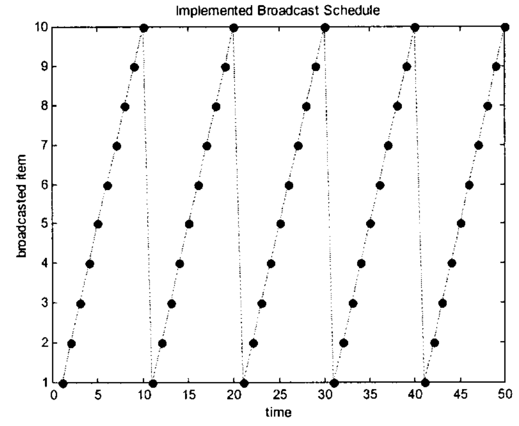


Figure 2.23 Implemented broadcast schedule (1)

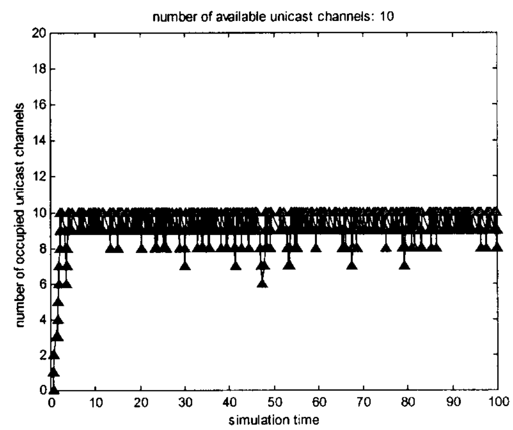


Figure 2.24.a) Unicast channel occupancy – 10 unicast channels available (1)

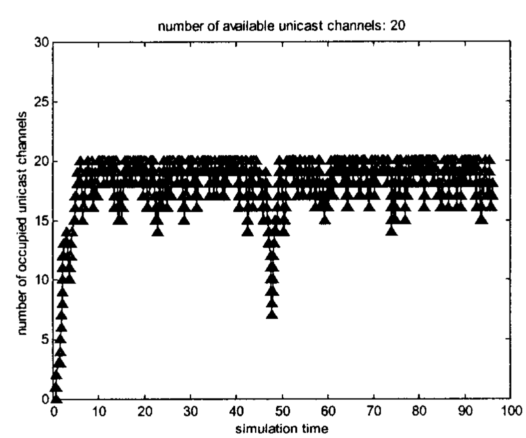


Figure 2.24.b) Unicast channel occupancy – 20 unicast channels available (1)

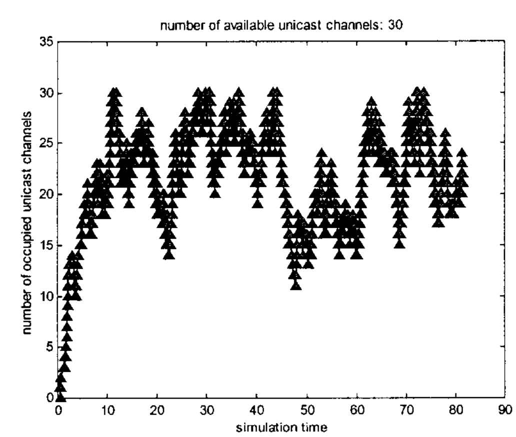


Figure 2.24.c) Unicast channel occupancy – 30 unicast channels available (1)

2.6.1.2 $P_{success}$ – Stochastic Behaviour: Experiment 2

data-item request probabilities	**p=[0.3, 0.2, 0.2, 0.1, 0.1, 0.1, 0.0, 0.0, 0.0, 0.0]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p=[3.33 5.0 5.0 10.0 10.0 10.0 Inf Inf Inf Inf]**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5 [users per time unit]**
mean MU arrival rate	**λ=10 [user per time unit]**
simulation time	**300 [time units]**

Table 2.5 Simulation parameters (2.6.1.2)

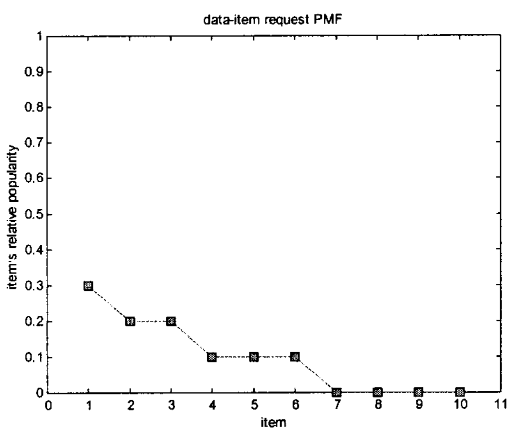


Figure 2.25 Data-item request probabilities (2)

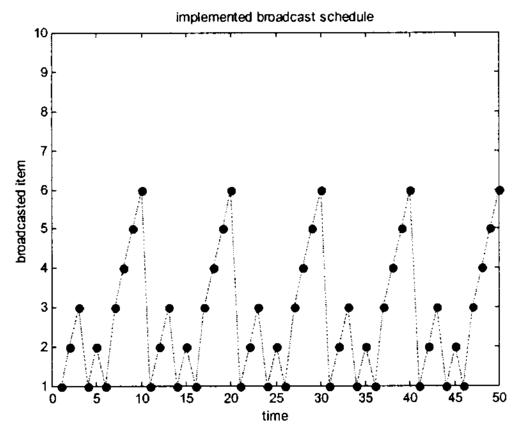


Figure 2.26 Implemented broadcast schedule (2)

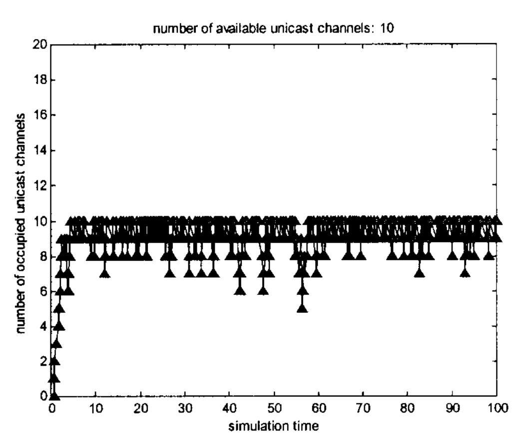


Figure 2.27.a) Unicast channel occupancy – 10 unicast channels available (2)

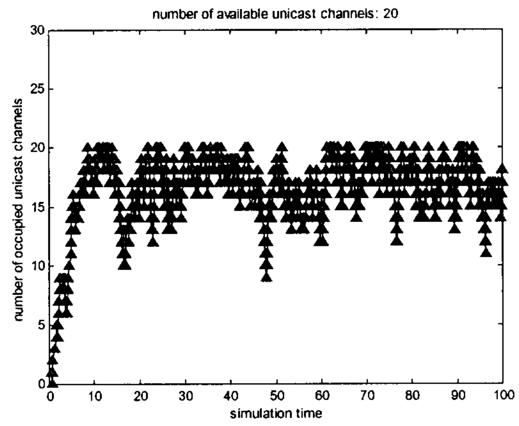


Figure 2.27.b) Unicast channel occupancy – 20 unicast channels available (2)

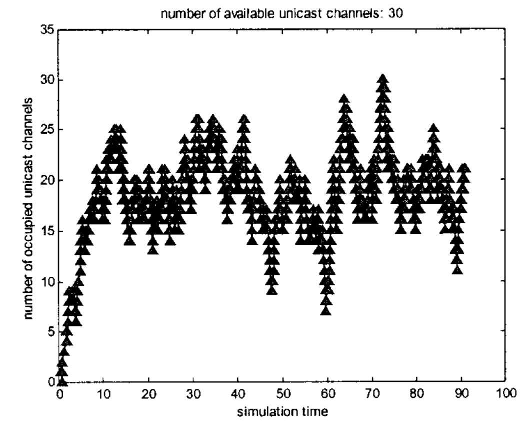


Figure 2.27.c) Unicast channel occupancy – 30 unicast channels available (2)

2.6.1.3 $P_{success}$ – Stochastic Behaviour: Experiment 3

data-item request probabilities	**p=[0.3, 0.2, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p=[3.33 5.0 10.0 10.0 20.0 20.0 20.0 20.0 20.0 20.0]**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.6 Simulation parameters (2.6.1.3)

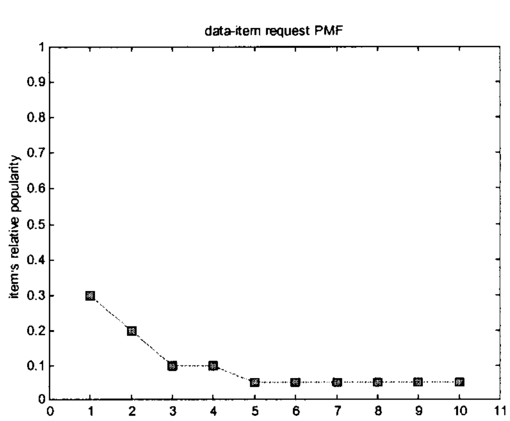


Figure 2.28 Data-item request probabilities (3)

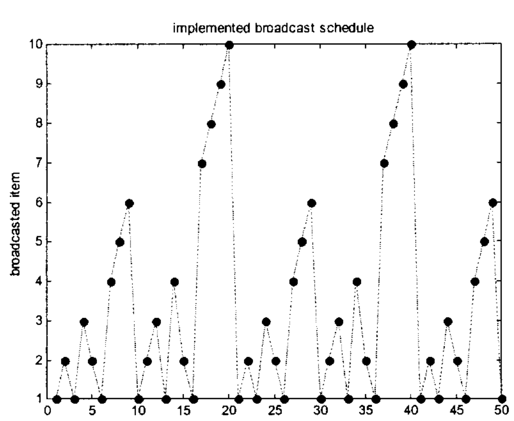


Figure 2.29 Implemented broadcast schedule (3)

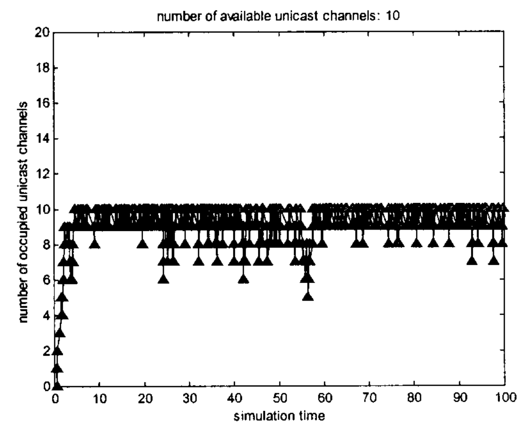


Figure 2.30.a) Unicast channel occupancy – 10 unicast channels available (3)

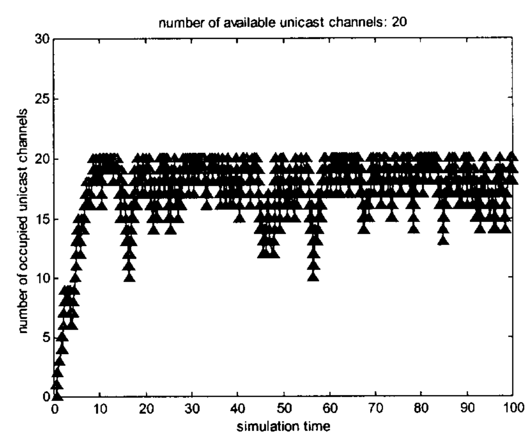


Figure 2.30.b) Unicast channel occupancy – 20 unicast channels available (3)

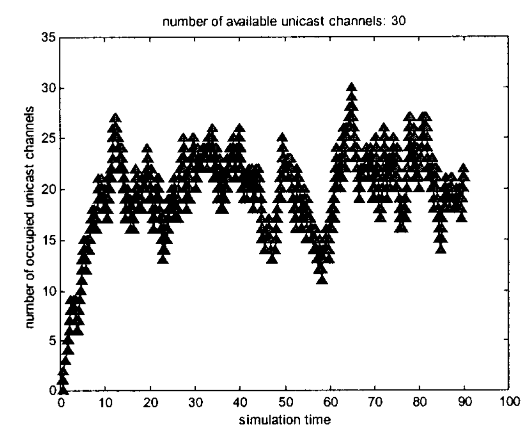


Figure 2.30.c) Unicast channel occupancy – 30 unicast channels available (3)

2.6.2 Analytical BR Obtained on Assumption of Constant $P_{success}$ as Sufficient Approximation of Actual BR

Statement 2.2 In spited of the fact that the actual $P_{success}$ exhibits non-stationary behaviour (see Statement 2.1), the assumption of a constant, yet adequately chosen $P_{success}$, enables sufficiently accurate analysis of the corresponding real-world single-cell hybrid WDB system.

Statement 2.3 Our analytical BR (see (2.34)), based on the assumption of a constant $P_{success}$, provides an accurate approximation of the actual BR, as observed in the corresponding real-world single-cell hybrid WDB system.

The simulation results presented in Sections 2.6.2.1 to 2.6.2.3 provide the proof for both of the above statements. Namely, in the three simulation settings, as mentioned in the previous section, and for k·10, k=0,1,..,5, available unicast channels, the overall number of users served through broadcast, served through unicast and give-up users were observed (see Table 2.9, 2.11, and 2.13). Based on these observed values, the actual (i.e. simulation-based) BR and UR were obtained simply as

$$BR_{simulation} = \frac{\text{number of users served through broadcast}}{\text{overall number of arrived users}} \tag{2.80}$$

$$UR_{simulation} = \frac{\text{number of users served through unicast}}{\text{overall number of arrived users}} \tag{2.81}$$

Subsequently, the corresponding simulation-based $P_{success}(t)$ ($t \in [0, maxSimulationTime]$) was estimated as the inverse of $UR(P_{success})$ (see (2.41) and (2.82)).

$$\overline{P}_{success} = UR^{-1}_{simulation}(P_{success}(t)) \tag{2.82}$$

This, finally, enabled that $BR_{simulation}=BR_{simulation}(\overline{P}_{success})$ be plotted (see Figure 2.31, 2.32, and 2.33). (Note, $\overline{P}_{success}$ in (2.89) is a constant value, and it, in fact, represents an approximated average of the actual time-varying $P_{success}(t)$, $t \in [0, maxSimulationTime]$. Also, note that $\overline{P}_{success}$ is being calculated as the inverse of $UR_{simulation}$, instead of $BR_{simulation}$, since UR is the only performance measure, out of the ones presented in Section 2.3, directly observable by the system.)

Now, Figures 2.31, 2.32, and 2.33 clearly indicate that in all three simulation settings $BR_{analytical}=BR_{analytical}(P_{success})$ (derived from (2.34) and presented in Tables 2.8, 2.10, and 2.12) and $BR_{simulation}=BR_{simulation}(\overline{P}_{success})$ were almost identical, as indicated in (2.83).

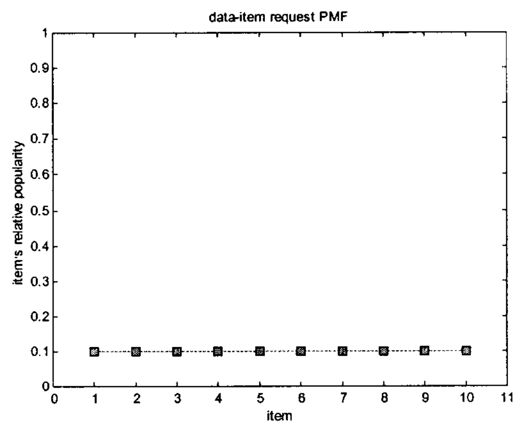
$$BR_{analytical}(P_{success}) = BR_{simulation}(\overline{P}_{success}), \qquad \text{where} \quad \overline{P}_{success} \approx E[P_{success}(t)] \tag{2.83}$$

Hence, (2.90) provides the ground for both of our claims: 1) in the analysis of a real-world single-cell hybrid WDB system, the true time-varying $P_{success}(t)$ can be sufficiently substituted by a constant, yet adequately approximated $\overline{P}_{success}$; 2) our analytical BR, obtained on the assumption of a constant $P_{success}$, provides an accurate approximation of the actual BR.

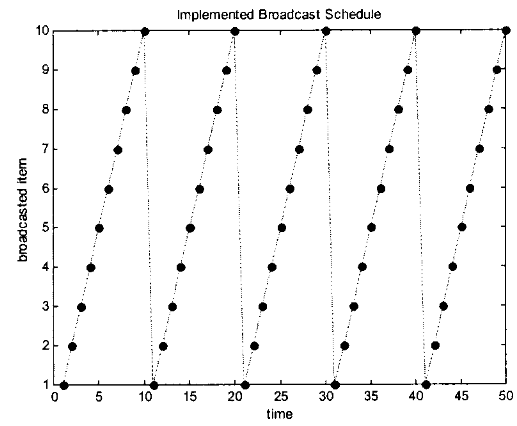
2.6.2.1 Theoretical vs. Simulation-obtained BR: Experiment 1

data-item request probabilities:	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]:	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]:	**s=l./p**
MU reattempt probability:	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]:	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Simulation parameters (1)



Data-item request probabilities (1)



Implemented broadcast schedule (1)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	$BR_{simulation}$	$UR_{simulation}$	$\bar{P}_{success}$	BR_{theory}	Relative Mean Error (BR_{theory} vs. $BR_{simulation}$)
50	604	2362	0	0.2036	0.7964	0.9868	0.2004	1.57 %
40	604	2361	1	0.2036	0.7960	0.9857	0.2006	1.47 %
30	615	2321	30	0.2073	0.7825	0.9510	0.2054	0.92 %
20	787	1874	302	0.2656	0.6325	0.6401	0.2610	1.73 %
10	1144	1009	815	0.3854	0.3400	0.2639	0.3760	2.44 %
0	1558	0	1403	0.5262	-	0.0	0.5179	1.58 %

Table 2.7 Simulation-obtained results (1)

$P_{success}$	$BR_{theoretical}$
1.0	0.1987
0.9	0.2130
0.8	0.2294
0.7	0.2483
0.6	0.2702
0.5	0.2958
0.4	0.3259
0.3	0.3615
0.2	0.4042
0.1	0.4555
0.0	0.5179

Table 2.8 Theoretical BR obtained assuming $P_{success}$=const (1) (see (2.34))

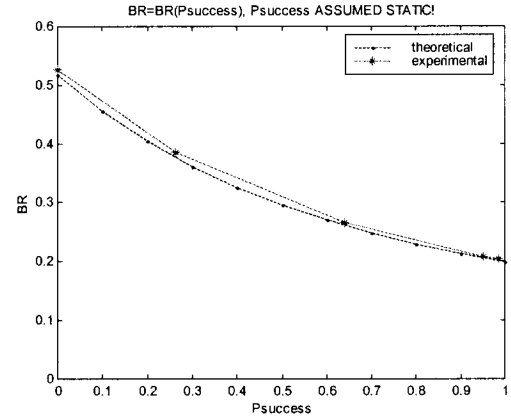
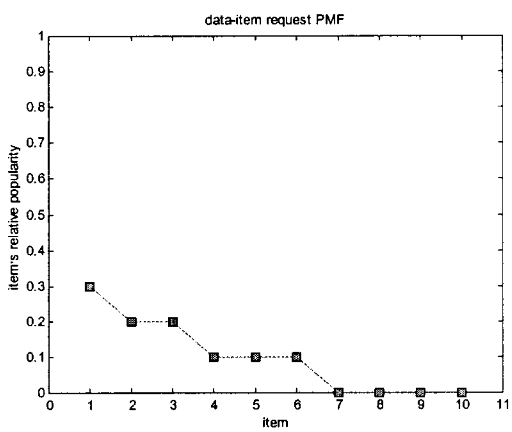


Figure 2.31 Theoretical vs. simulation-obtained values of BR (1)

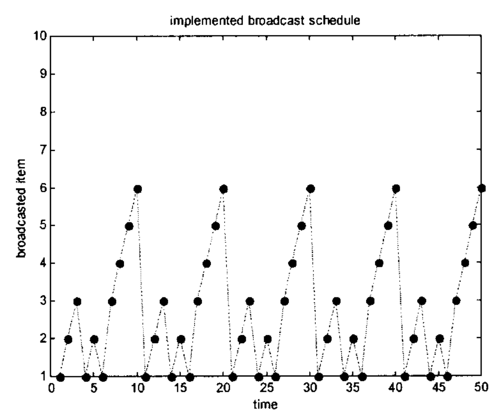
2.6.2.2 Theoretical vs. Simulation-obtained BR; Experiment 2

data-item request probabilities	**p=[0.3, 0.2, 0.2, 0.1, 0.1, 0.1, 0.0, 0.0, 0.0, 0.0]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p=[3.33 5.0 5.0 10.0 10.0 10.0 Inf Inf Inf Inf]**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Simulation parameters (2)



Data-item request probabilities (2)



Implemented broadcast schedule (2)

# of unicast channels	**# of users served through broadcast**	**# of users served through unicast**	**# of give-up users**	**$BR_{simulation}$**	**$UR_{simulation}$**	**$\bar{P}_{success}$**	**BR_{theory}**	**Relative Mean Error (BR_{theory} vs. $BR_{simulation}$)**
50	1005	1962	0	0.3387	0.6612	1.0	0.3525	4.07 %
40	1005	1962	0	0.3387	0.6612	1.0	0.3525	4.07 %
30	1012	1953	2	0.3410	0.6582	1.0	0.3525	3.37 %
20	1125	1730	119	0.3782	0.5817	0.8359	0.3848	1.75 %
10	1472	981	512	0.4964	0.3308	0.3791	0.5083	2.40 %
0	1976	0	995	0.6650	0	0	0.6729	1.19 %

Table 2.9 Simulation-obtained results (2)

$P_{success}$	$BR_{theoretical}$
1.0	0.3525
0.9	0.3716
0.8	0.3926
0.7	0.4157
0.6	0.4412
0.5	0.4696
0.4	0.5012
0.3	0.5366
0.2	0.5764
0.1	0.6215
0.0	0.6729

Table 2.10 Theoretical BR obtained assuming $P_{success}$=const (2) (see (2.34))

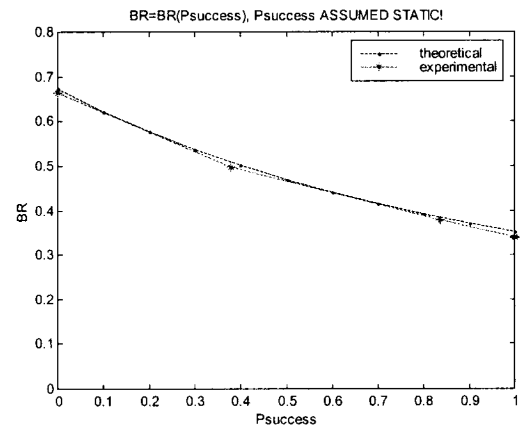
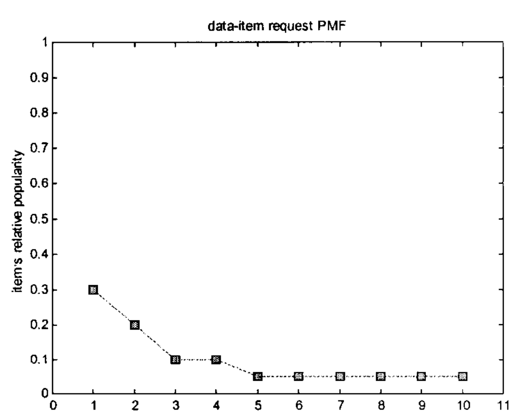


Figure 2.32 Theoretical vs. simulation-obtained values of BR (2)

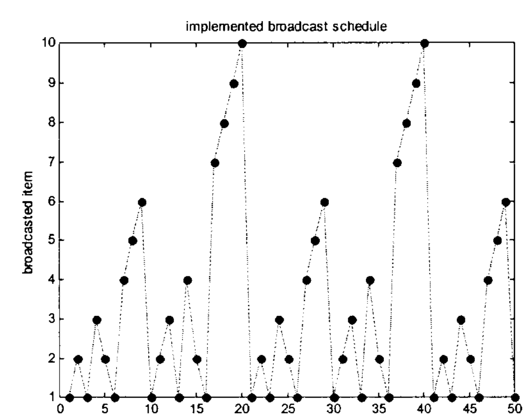
2.6.2.3 Theoretical vs. Simulation-obtained BR: Experiment 3

data-item request probabilities	**p=[0.3, 0.2, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p=[3.33 5.0 10.0 10.0 20.0 20.0 20.0 20.0 20.0 20.0]**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Simulation parameters (3)



Data-item request probabilities (3)



Implemented broadcast schedule (3)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	$BR_{simulation}$	$UR_{simulation}$	$\bar{P}_{success}$	BR_{theory}	Relative Mean Error (BR_{theory} vs. $BR_{simulation}$)
50	861	2102	0	0.2905	0.7094	1.0	0.2892	0.45 %
40	861	2102	0	0.2905	0.7094	1.0	0.2892	0.45 %
30	866	2092	5	0.2922	0.7060	1.0	0.2892	1.03 %
20	991	1797	175	0.3344	0.6064	0.8941	0.3059	8.52 %
10	1293	1001	665	0.4369	0.3382	0.3899	0.4176	4.42 %
0	1698	0	1262	0.5736	-	0	0.5754	0.31 %

Table 2.11 Simulation-obtained results (3)

$P_{success}$	$BR_{theoretical}$
1.0	0.2892
0.9	0.3049
0.8	0.3223
0.7	0.3416
0.6	0.3631
0.5	0.3873
0.4	0.4146
0.3	0.4459
0.2	0.4821
0.1	0.5246
0.0	0.5754

Table 2.12 Theoretical BR obtained assuming $P_{success}$=const (3) (see (2.34))

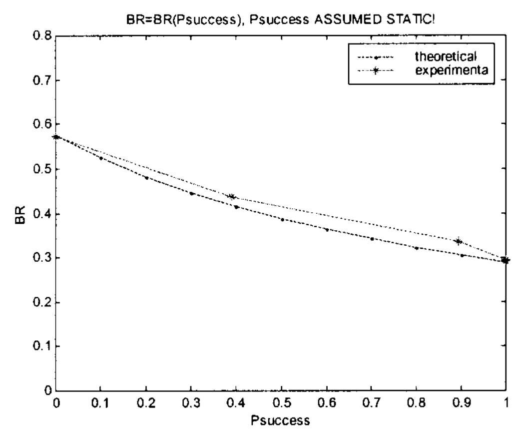


Figure 2.33 Theoretical vs. simulation-obtained values of BR (3)

2.6.3 $P_{reattempt}$ - Validity of Constant Approximation

Statement 2.4 In the cases when users' patience, and accordingly $P_{reattempt}$, show significant time-decreasing tendency (see Section 2.3.1.2), the assumption of a constant, yet adequately chosen $P_{reattempt}$, still can ensure sufficiently accurate analysis of the corresponding real-world single-cell hybrid WDB system.

The simulation results presented in Sections 2.6.3.1.a) to 2.6.3.1.d) provide the proof for Statement 2.4. These results were collected under the same set of main system-related simulation parameters (see Table 2.13, 2.16, 2.19, 2.22), while different form of user impatience (i.e. different form of time-decreasing $P_{reattempt}$) was employed in each case (Figure 2.34, 2.37, 2.40, 2.43). The simulation results obtained in all the cases clearly indicated that the time-decreasing nature of $P_{reattempt}$ had a significant effect on the overall system performance. Accordingly, the analytical BR obtained on the assumption of a non-decreasing $P_{reattempt}$ ($P_{reattempt}$=0.7) failed to provide an adequate approximation of the actual (i.e. simulation-based) BR (see Figure 2.35, 2.38, 2.41, 2.44). Nevertheless, in each of the four cases, through simple trial and error, we were able to identify a constant value of $P_{reattempt}$ whose corresponding analytical BR eventually provided a good match to the simulation-based BR (Figure 2.36, 2.39, 2.42, 2.45). More sophisticated methods for capturing a constant, replacing value of $P_{reattempt}$ remain in the scope of our future work.

2.6.3.1 Experiment 1: $P_{reattempt}(t)=0.7 \cdot e^{-bt}$, where $b=0.05$

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	**$P_{reattempt}=0.7 \cdot e^{-bt}$, b=0.05**
MU mean reattempt time [time units]	**$\lambda^*=0.5$**
mean MU arrival rate	**$\lambda=10$ [users per time unit]**
simulation time	**300 [time units]**

Table 2.13 Simulation parameters 2.6.3.1

# of unicast channels	# of users served thr. broadcast	# of users served thr. unicast	# of give-up users	BR	UR	$\bar{P}_{success}$
50	604	2362	0	0.2036	0.7963	0.9865
40	604	2360	2	0.2036	0.7956	0.9847
30	612	2321	33	0.2063	0.7825	0.9510
20	754	1863	348	0.2543	0.6283	0.6330
10	1016	1009	942	0.3424	0.3400	0.2639
0	1318	0	1643	0.4451	-	0.0

Table 2.14 Simulation obtained BR for $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=0.05

$P_{success}$	BR
1.0	0.1987
0.8	0.2245
0.6	0.2573
0.4	0.2998
0.2	0.3560
0.0	0.4323

Table 2.15 Theoretical BR obtained for $P_{reattempt}=0.6$

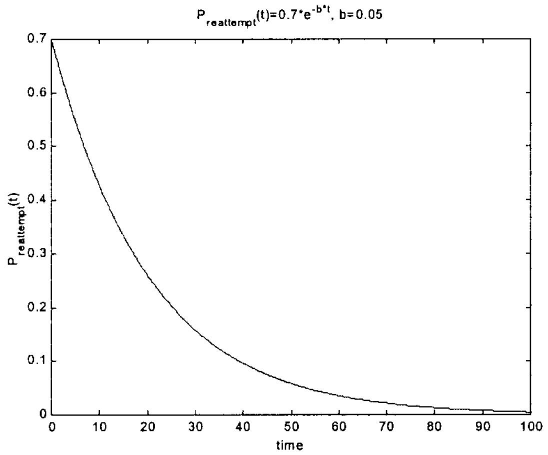


Figure 2.34 $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=0.05

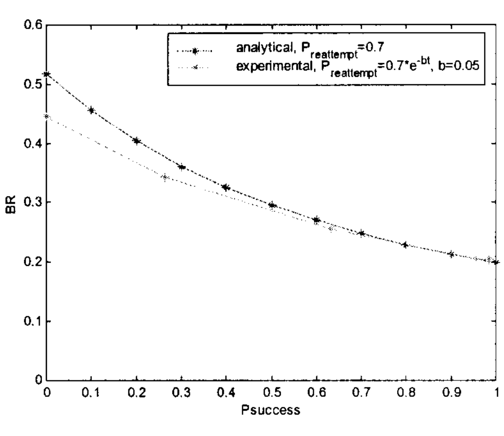


Figure 2.35 Analytical BR obtained for $P_{reattempt}=0.7$ (see Table 2.8) vs. simulation obtained BR in Experiment 1

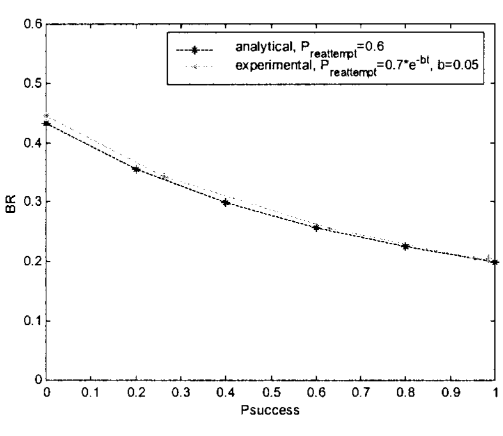


Figure 2.36 Analytical BR obtained for $P_{reattempt}=0.6$ vs. simulation obtained BR in Experiment 1

2.6.3.2 Experiment 2: $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where $b=0.1$

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	$P_{reattempt}=0.7\cdot e^{-bt}$, **b=0.1**
MU mean reattempt time [time units]	$\lambda^{*}=0.5$
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.16 Simulation parameters 2.6.3.2

# of unicast channels	# of users served thr. broadcast	# of users served thr. unicast	# of give-up users	BR	UR	$\bar{P}_{success}$
50	604	2362	0	0.2036	0.7963	0.9865
40	604	2360	2	0.2036	0.7954	0.9841
30	612	2321	33	0.2063	0.7825	0.9510
20	725	1863	377	0.2445	0.6283	0.6330
10	961	1004	1003	0.3237	0.3382	0.2621
0	1198	0	1766	0.4041	-	0.0

Table 2.17 Simulation obtained BR for $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=0.1

$P_{success}$	BR
1.0	0.1987
0.8	0.2221
0.6	0.2512
0.4	0.2880
0.2	0.3354
0.0	0.3976

Table 2.18 Theoretical BR obtained for $P_{reattempt}=0.55$

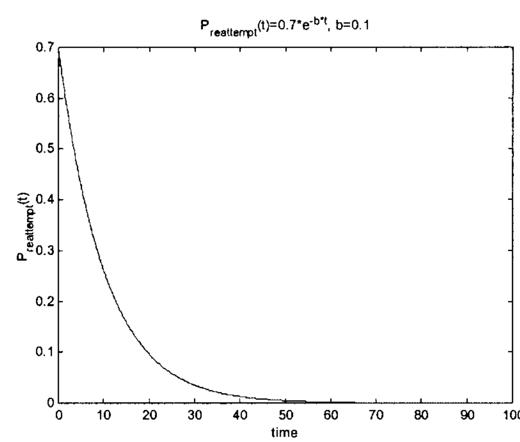


Figure 2.37 $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=0.1

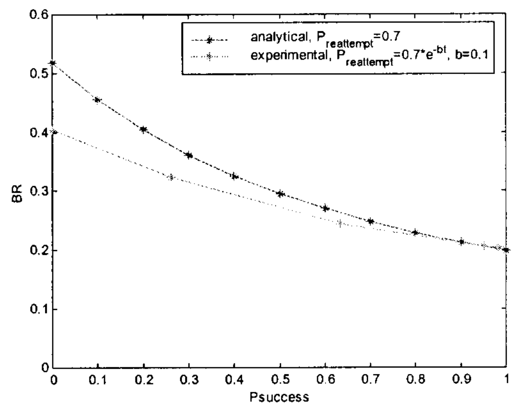


Figure 2.38 Analytical BR obtained for $P_{reattempt}$=0.7 (see Table 2.8) vs. simulation obtained BR in Experiment 2

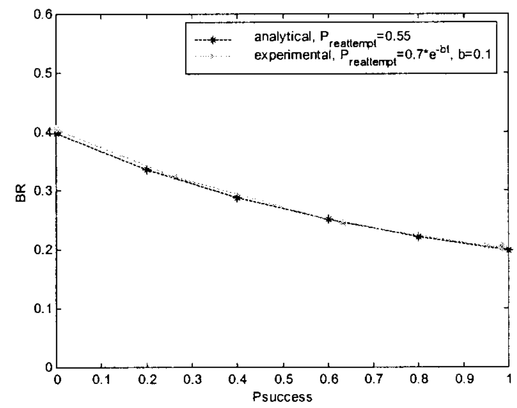


Figure 2.39 Analytical BR obtained for $P_{reattempt}$=0.55 vs. simulation obtained BR in Experiment 2

2.6.3.3 Experiment 3: $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where $b=0.5$

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l./p**
MU reattempt probability	**$P_{reattempt}=0.7\cdot e^{-bt}$, b=0.5**
MU mean reattempt time [time units]	**$\lambda^*=0.5$**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.19 Simulation parameters 2.6.3.3

# of unicast channels	# of users served thr. broadcast	# of users served thr. unicast	# of give-up users	BR	UR	$\bar{P}_{success}$
50	604	2362	0	0.2036	0.7963	0.9865
40	604	2360	2	0.2036	0.7956	0.9847
30	614	2311	41	0.2070	0.7791	0.9424
20	672	1835	462	0.2263	0.6180	0.6158
10	766	994	1207	0.2581	0.3350	0.2590
0	865	0	2101	0.2916	0.0	0.0

Table 2.20 Simulation obtained BR for $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=0.5

$P_{success}$	BR
1.0	0.1978
0.8	0.2130
0.6	0.2294
0.4	0.2483
0.2	0.2702
0.0	0.2958

Table 2.21 Theoretical BR obtained for $P_{reattempt}$=0.35

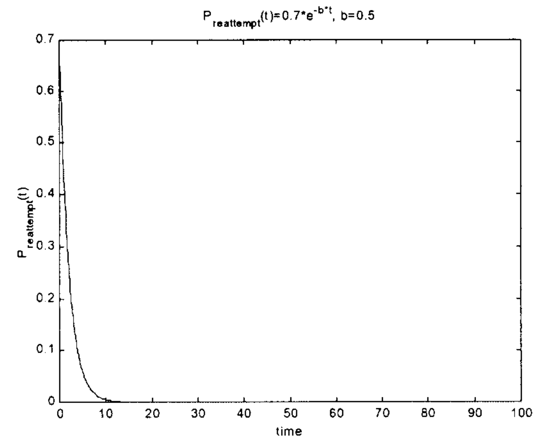


Figure 2.40 $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=0.5

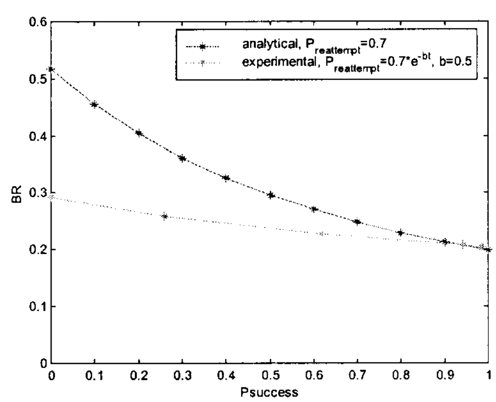


Figure 2.41 Analytical BR obtained for $P_{reattempt}$=0.7 (see Table 2.8) vs. simulation obtained BR in Experiment 3

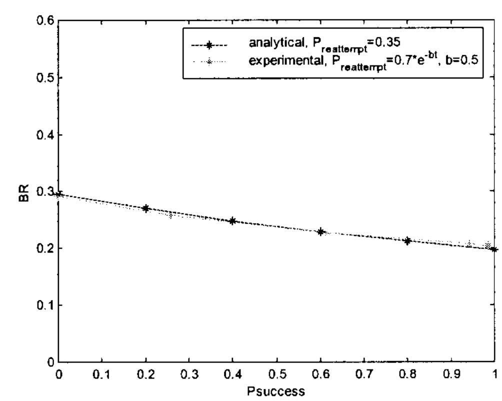


Figure 2.42 Analytical BR obtained for $P_{reattempt}$=0.35 vs. simulation obtained BR in Experiment 3

2.6.3.4 Experiment 4: $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=1.0

data-item request probabilities:	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]:	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]:	**s=l./p**
MU reattempt probability:	**$P_{reattempt}=0.7\cdot e^{-bt}$, b=1.0**
MU mean reattempt time [time units]:	**$\lambda^*=0.5$**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.22 Simulation parameters 2.6.3.1.4

# of unicast channels	# of users served thr. broadcast	# of users served thr. unicast	# of give-up users	BR	UR	$\bar{P}_{success}$
50	604	2362	0	0.2036	0.7963	0.9865
40	604	2360	2	0.2036	0.7956	0.9847
30	610	2306	50	0.2056	0.7774	0.9382
20	638	1830	497	0.2151	0.6172	0.6145
10	703	986	1276	0.2370	0.3325	0.2566
0	761	0	2204	0.2566	-	0.0

Table 2.23 Simulation obtained BR for $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=1.0

$P_{success}$	BR
1.0	0.1987
0.8	0.2087
0.6	0.2198
0.4	0.2319
0.2	0.2454
0.0	0.2604

Table 2.24 Theoretical BR obtained for $P_{reattempt}=0.15$

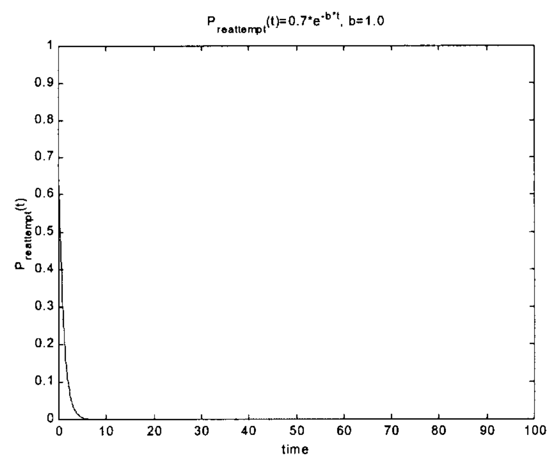


Figure 2.43 $P_{reattempt}(t)=0.7\cdot e^{-bt}$, where b=1.0

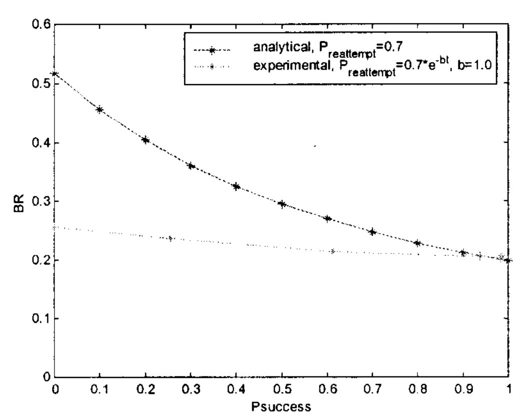


Figure 2.44 Analytical BR obtained for $P_{reattempt}=0.7$ (see Table 2.8) vs. simulation obtained BR in Experiment 4

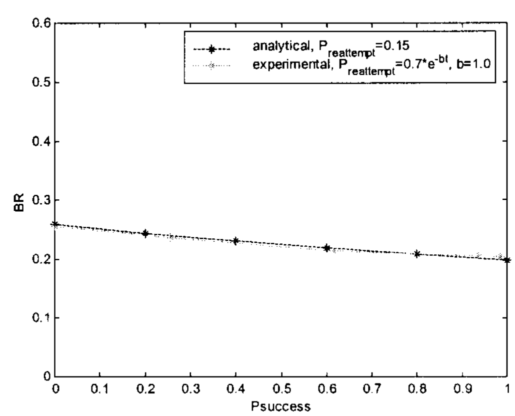


Figure 2.45 Analytical BR obtained for $P_{reattempt}=0.15$ vs. simulation obtained BR in Experiment 4

2.6.4 Robustness of Analytical BR to Non-Poissonian Arrivals

Statement 2.5 Our theoretical analysis of single-cell hybrid WDB systems has been based on the assumption of a Poisson user arrival process. Nevertheless, our analytical BR (see (2.34)) turns out to be a sufficiently accurate approximation of the actual BR, even in the cases of "on-off" user arrivals processes[20].

The simulation results presented in Sections 2.6.4.1.a) to 2.6.4.1.h) provide the proof for Statement 2.5. These results were collected under the same set of main simulation parameters (see Table 2.25, 2.27, 2.29, 2.31, 2.33, 2.35, 2.37, 2.39), however a different user arrival process was employed in each case. More specifically,

- in the first four simulation settings (Sections 2.6.4.1.a) to 2.6.4.1.d)), the user inter-arrival times followed a normal and uniform distribution, respectively, with the same, i.e. constant, mean value ($1/\lambda=0.1$), but different standard deviation[21];
- in the next two simulation settings (Sections 2.6.4.1.e) and 2.6.4.1.f)), a bursty (i.e. modulated) Poisson user arrival process was employed, with the mean user arrival rate altering between $\lambda_1=5$ and $\lambda_2=15$ every 30 or 60 [time units];
- in the last two simulation settings (Sections 2.6.4.1.g) and 2.6.4.1.h)), a bursty non-Poissoninan user arrival process was employed – here, in contrast to the previous two cases, the user inter-arrival times followed a normal, instead of an exponential, distribution.

(Figure 2.46 provides an illustration of a real-world situation where the user arrival process is expected to behave in "on-off" manner, due to the users' mobility.)

[20] We refer to a user arrival process as "on-off", or "modulated", if its average arrival rate is discretely changing between two values (low and high).

[21] A Poisson user arrival process requires that the user inter-arrival times follow an exponential distribution.

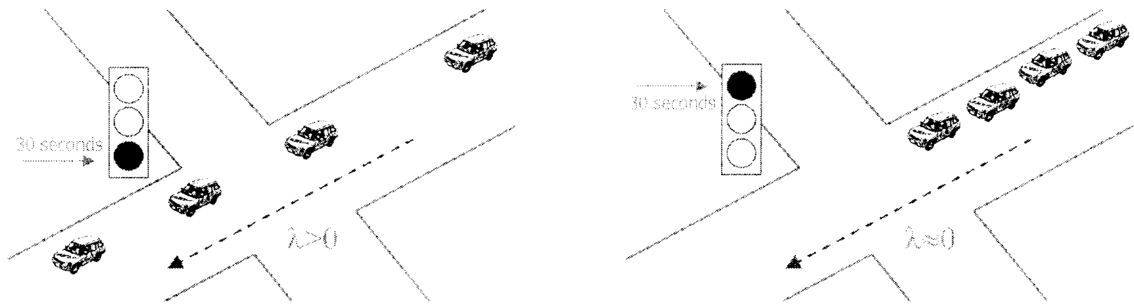


Figure 2.46 Bursty ("on-off") MU arrival process

In each of the eight simulation settings, by changing the number of available unicast channels, first the corresponding simulation-based BRs were observed (see Table 2.26, 2.28, 2.30, 2.32, 2.34, 2.36, 2.38, 2.40), and then the plots $BR_{simulation}=BR_{simulation}$(# available unicast channels | non-Poissonian user arrival process) were generated. Finally, these plots were compared against an equivalent plot obtained by simulating a single pure Poisson user arrival process[22] (see Figures 2.48, 2.50, 2.52, 2.54, 2.58, 2.62, 2.66, 2.70). Interestingly enough, in all eight cases the plots turned out to be very close, hence suggesting the following

$$BR_{simulation}(\text{Poisson user arrival process}) \approx BR_{simulation}(\text{non}-\text{Poissonian user arrival process}) \quad (2.84)$$

However, since we already know from Statement 2.3 that the analytical BR provides an accurate approximation of $BR_{simulation}$(Poisson user arrival process), then Statement 2.3 and (2.84) together imply (2.85), thereby providing the proof for Statement 2.4.

$$BR_{analytical}(\text{Poisson user arrival process}) \approx BR_{simulation}(\text{non}-\text{Poissonian user arrival process}) \quad (2.85)$$

The validity of Statement 2.4 is also verified through (2.86) to (2.93), which provide the value of relative mean error (RME) between the analytical BR and BR obtained through simulation. Clearly, based on these expressions, the RME is kept in the range of only a few percent, in all

[22] Please note, again, such a simulation, that employs a Poisson user arrival process, corresponds to the assumption of our theoretical analysis.

simulation cases, hence suggesting that the analytical BR indeed represents a fairly good approximation of the actual BR, regardless of the actual user arrival process.

Finally, we would like to point out that the exact analytical expression for BR, in the case of a modulated Poisson user arrival process, could be readily obtained by following our analysis from Section 2.3.2. In particular, the given expression could be obtained by simply modifying (2.30), so as to accommodate two possible, alternating values of λ, i.e. λ (i). Nevertheless, it would be very difficult, if possible at all, to obtain the exact analytical expression for BR, in the case of other non-Poissonian user arrival processes, as discussed in this section.

2.6.4.1 Robustness of Analytical BR – Various Experiments

2.6.4.1.a) Normal MU Inter-arrival Time Distribution (inter-arrival time $\in \mathcal{N}(0.1, 0.01)$)

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.25 Simulation parameters (2.6.4.1.a)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	$BR_{simulation}$	UR	$\bar{P}_{success}$	BR_{theory}
50	564	2417	0	0.1892	0.8108	1.0	0.1987
40	564	2417	0	0.1892	0.8108	1.0	0.1987
30	586	2372	23	0.1966	0.7957	0.9849	0.2007
20	757	1881	338	0.2544	0.6321	0.6395	0.2611
10	1114	1009	851	0.3746	0.3393	0.2632	0.3763
0	1531	0	1438	0.5157	0	0	0.5179

Table 2.26 Simulation-obtained values of BR for inter-arrival time $\in$ N (0.1, 0.01) (a)

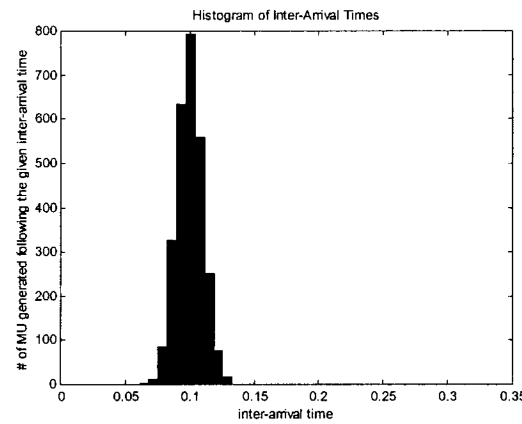


Figure 2.47 Histogram of MU inter-arrival times (a)

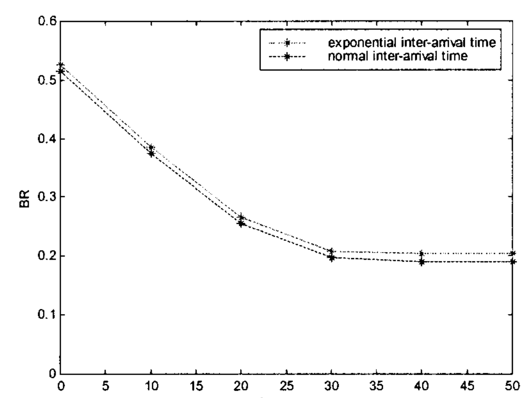


Figure 2.48 Poisson-based BR vs. BR obtained for inter-arrival times $\in$ N (0.1, 0.01) (a)
(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with λ=10)

$$\mathrm{RME}(\mathrm{BR}_{\mathrm{simulation}}, \mathrm{BR}_{\mathrm{theory}}) = \frac{1}{6}\sum_{k=1}^{6} \frac{|\mathrm{BR}_{\mathrm{theory}}(k) - \mathrm{BR}_{\mathrm{simulation}}(k)|}{\mathrm{BR}_{\mathrm{simulation}}(k)} = 2.61\,\% \tag{2.86}$$

2.6.4.1.b) Normal MU Inter-arrival Time Distribution (inter-arrival time ∈ $\mathcal{N}(0.1, 0.05)$)

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	$P_{reattempt}=0.7$
MU mean reattempt time [time units]	$\lambda^{*}=0.5$
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.27 Simulation parameters (2.6.4.1.b)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	$BR_{simulation}$	UR	$\bar{P}_{success}$	BR_{theory}
50	605	2321	0	0.2068	0.7932	0.9784	0.2016
40	605	2321	0	0.2068	0.7932	0.9784	0.2016
30	613	2299	14	0.2095	0.7857	0.9591	0.2043
20	760	1863	296	0.2604	0.6382	0.6499	0.2588
10	1116	1009	790	0.3828	0.3461	0.2699	0.3736
0	1582	0	1379	0.5343	0	0	0.5179

Table 2.28 Simulation-obtained values of BR for inter-arrival time ∈ N (0.1, 0.05) (b)

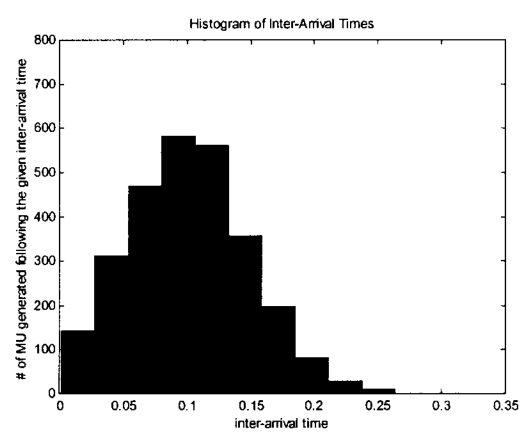


Figure 2.49 Histogram of MU inter-arrival times (b)

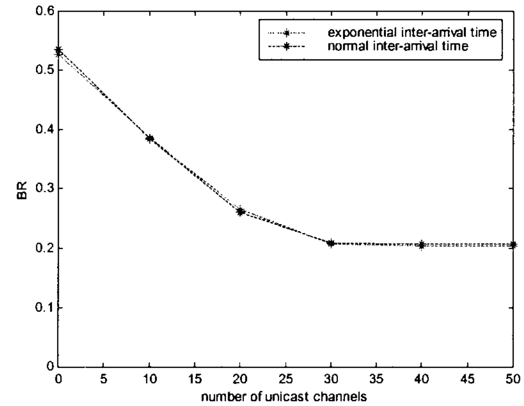


Figure 2.50 Poisson-based BR vs. BR obtained for inter-arrival times $\in$ N(0.1, 0.05) (b)

(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with λ=10)

$$\text{RME}(\text{BR}_{\text{simulation}}, \text{BR}_{\text{theory}}) = \frac{1}{6}\sum_{k=1}^{6}\frac{|\,\text{BR}_{\text{theory}}(k) - \text{BR}_{\text{simulation}}(k)\,|}{\text{BR}_{\text{simulation}}(k)} = 2.27\ \% \qquad (2.87)$$

2.6.4.1.c) Uniform MU Inter-arrival Time Distribution (inter-arrival time $\in$ [0.08, 0.12])

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.29 Simulation parameters (2.6.4.1.c)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	$BR_{simulation}$	UR	$\bar{P}_{success}$	BR_{theory}
50	556	2425	0	0.1865	0.8135	1.0	0.1987
40	556	2425	0	0.1865	0.8135	1.0	0.1987
30	572	2388	21	0.1919	0.8011	0.9992	0.1988
20	746	1890	343	0.2504	0.6344	0.6434	0.2603
10	1111	1015	849	0.3734	0.3412	0.2650	0.3756
0	1533	0	1435	0.5165	0	0	0.5179

Table 2.30 Simulation-obtained values of BR for inter-arrival time $\in$ uniform [0.08, 0.12] (c)

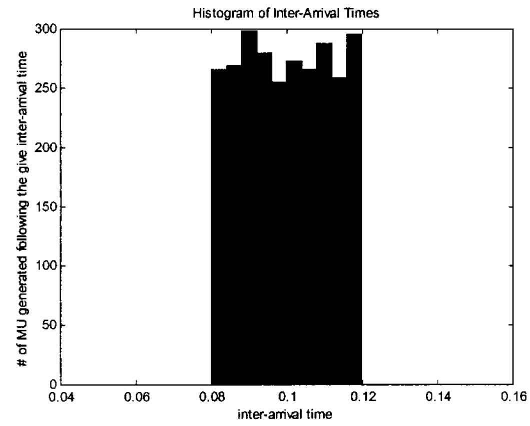


Figure 2.51 Histogram of MU inter-arrival times (c)

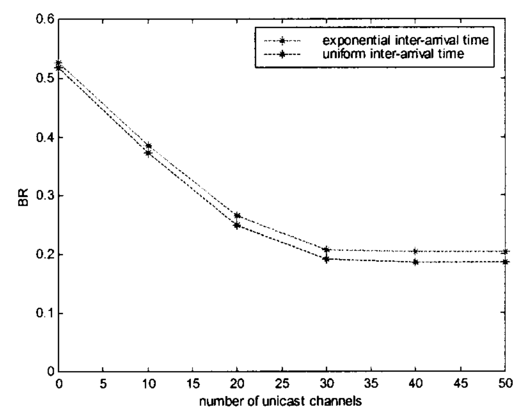


Figure 2.52 Poisson-based BR vs. BR obtained for inter-arrival times $\in$ uniform [0.08, 0.12] (c)

(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with λ=10)

$$\mathrm{RME}(\mathrm{BR}_{\text{simulation}}, \mathrm{BR}_{\text{theory}}) = \frac{1}{6}\sum_{k=1}^{6} \frac{|\,\mathrm{BR}_{\text{theory}}(k) - \mathrm{BR}_{\text{simulation}}(k)\,|}{\mathrm{BR}_{\text{simulation}}(k)} = 3.58\ \% \qquad (2.88)$$

2.6.4.1.d) Uniform MU Inter-arrival Time Distribution (inter-arrival time ∈ [0.05, 0.15])

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Table 2.31 Simulation parameters (2.6.4.1.d)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	$BR_{simulation}$	UR	$\bar{P}_{success}$	BR_{theory}
50	585	2396	0	0.1962	0.8038	1.0	0.1987
40	585	2396	0	0.1962	0.8038	1.0	0.1987
30	594	2365	22	0.1993	0.7934	0.9789	0.2015
20	765	1880	332	0.2570	0.6315	0.6384	0.2614
10	1141	1009	822	0.3839	0.3395	0.2634	0.3763
0	1556	0	1412	0.5243	0	0	0.5179

Table 2.32 Simulation-obtained values of BR for inter-arrival time ∈ uniform [0.05, 0.15] (d)

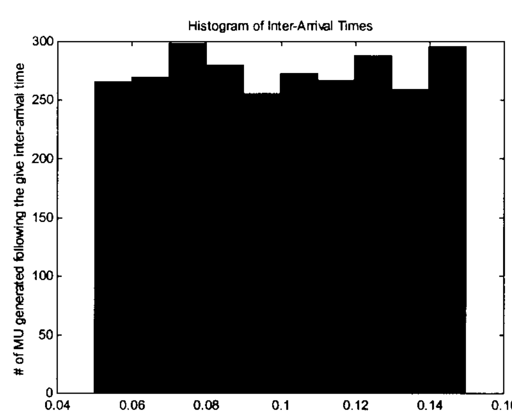


Figure 2.53 Histogram of MU inter-arrival times (d)

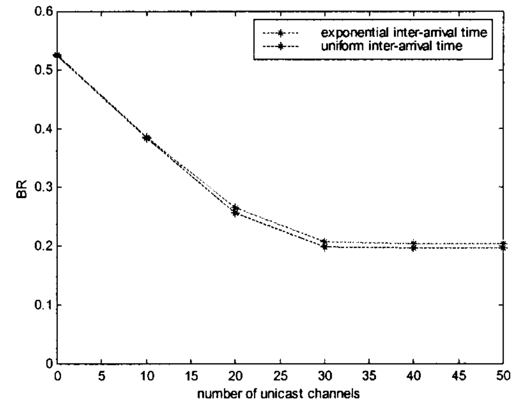


Figure 2.54 Poisson-based BR vs. BR obtained for inter-arrival times ∈ uniform [0.05, 0.15] (d)

(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with λ=10)

$$\mathrm{RME}(\mathrm{BR}_{\text{simulation}}, \mathrm{BR}_{\text{theory}}) = \frac{1}{6}\sum_{k=1}^{6}\frac{|\,\mathrm{BR}_{\text{theory}}(k) - \mathrm{BR}_{\text{simulation}}(k)\,|}{\mathrm{BR}_{\text{simulation}}(k)} = 1.43\ \% \qquad (2.89)$$

2.6.4.1.e) Poisson-Modulated MU Inter-arrival Time Distribution

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rates	**λ_1=5 [users per time unit], λ_2=15 [users per time unit]**
state occupancy time (see Figure 2.54)	**30 [time units]**
simulation time	**300 [time units]**

Table 2.33 Simulation parameters (2.6.4.1.e)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	BR	UR	$\bar{P}_{success}$	BR_{theory}
50	547	2333	0	0.1899	0.8101	1.0	0.1987
40	565	2290	25	0.1962	0.7951	0.9834	0.2009
30	644	2058	173	0.2240	0.7158	0.7976	0.2298
20	793	1621	452	0.2767	0.5642	0.5320	0.2871
10	1033	958	864	0.3618	0.3356	0.2596	0.3778
0	1448	0	1401	0.5082	0	0	0.5179

Table 2.34 Simulation obtained values of BR for Poisson-modulated arrival process - λ_1=5, λ_2=15 (e)

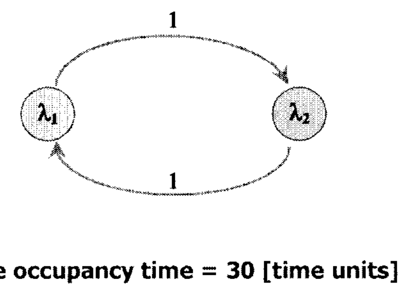


Figure 2.55 Change in MU arrival rates (e)

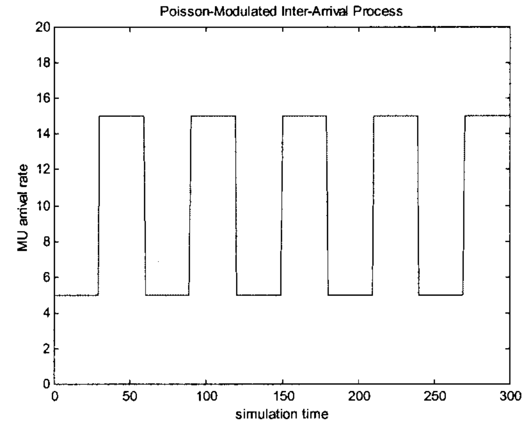


Figure 2.56 MU arrival rates (e)

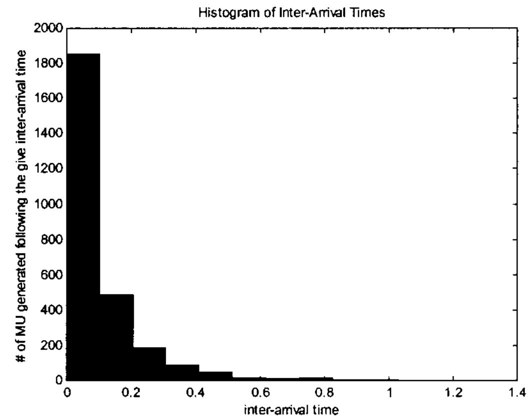


Figure 2.57 Histogram of MU inter-arrival times (e)

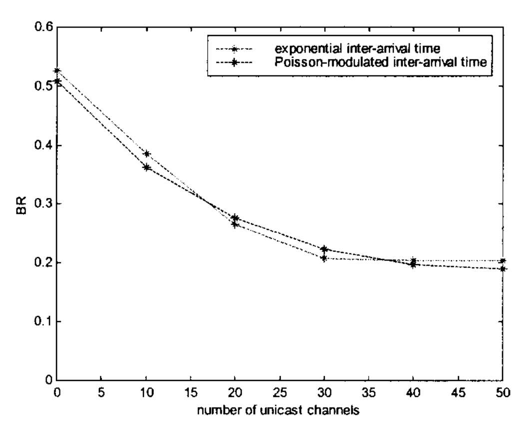


Figure 2.58 Poisson-based BR vs. BR obtained for Poisson-modulated inter-arrivals (e)
(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with λ=10)

$$\mathrm{RME}(\mathrm{BR}_{\mathrm{simulation}}, \mathrm{BR}_{\mathrm{theory}}) = \frac{1}{6}\sum_{k=1}^{6}\frac{|\,\mathrm{BR}_{\mathrm{theory}}(k) - \mathrm{BR}_{\mathrm{simulation}}(k)\,|}{\mathrm{BR}_{\mathrm{simulation}}(k)} = 3.28\ \% \tag{2.90}$$

2.6.4.1.f) Poisson-Modulated MU Inter-arrival Time Distribution

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rates	**λ_1=5 [users per time unit], λ_2=15 [users per time unit]**
state occupancy time (see Figure 2.58)	**60 [time units]**
simulation time	**300 [time units]**

Table 2.35 Simulation parameters (2.6.4.1.f)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	BR	UR	$\bar{P}_{success}$	BR_{theory}
50	497	2115	0	0.1902	0.8097	1.0	0.1987
40	518	2060	34	0.1983	0.7887	0.9668	0.2032
30	603	1844	165	0.2308	0.7060	0.7774	0.2334
20	715	1527	370	0.2737	0.5846	0.5626	0.2793
10	929	944	739	0.3556	0.3614	0.2852	0.3674
0	1336	0	1273	0.5120	0	0	0.5179

Table 2.36 Simulation obtained values of BR

for Poisson-modulated arrival process - λ_1=5, λ_2=15 (f)

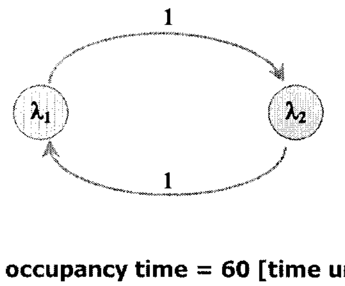


Figure 2.59 Change in MU arrival rates (f)

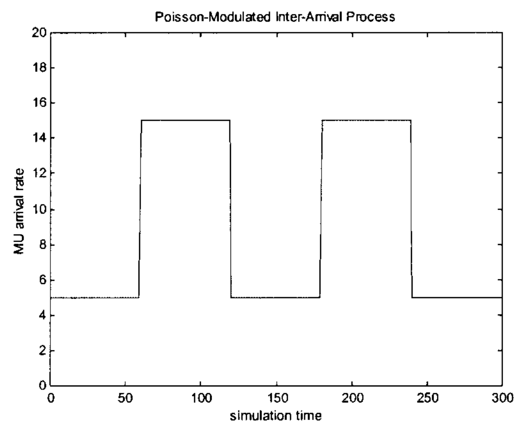


Figure 2.60 MU arrival rates (f)

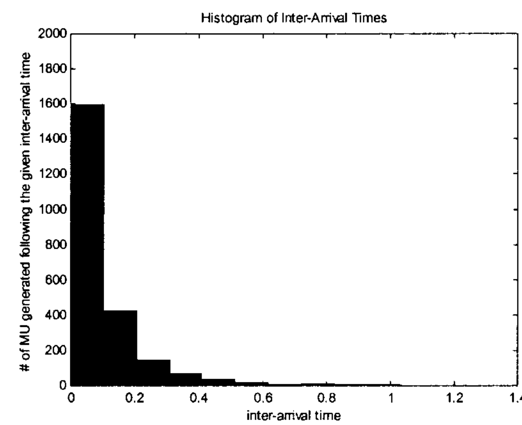


Figure 2.61 Histogram of MU inter-arrival times (f)

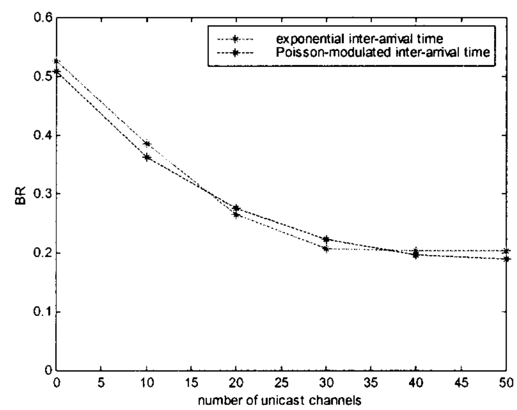


Figure 2.62 Poisson-based BR vs. BR obtained for Poisson-modulated inter-arrivals (f)
(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with λ=10)

$$\mathrm{RME}(\mathrm{BR}_{\mathrm{simulation}}, \mathrm{BR}_{\mathrm{theory}}) = \frac{1}{6}\sum_{k=1}^{6}\frac{|\mathrm{BR}_{\mathrm{theory}}(k) - \mathrm{BR}_{\mathrm{simulation}}(k)|}{\mathrm{BR}_{\mathrm{simulation}}(k)} = 2.43\ \% \qquad (2.91)$$

2.6.4.1.g) Normal-Modulated MU Inter-arrival Time Distribution

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	$\mathbf{P_{reattempt}=0.7}$
MU mean reattempt time [time units]	$\lambda^*=0.5$
MU arrival times (see Figure 2.64)	**time$_1$ =N(0.2,0.01), time$_2$ =N(0.0677,0.01)**
state occupancy time (see Figure 2.62)	**30 [time units]**
simulation time	**300 [time units]**

Table 2.37 Simulation parameters (2.6.4.1.g)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	BR	UR	$\bar{P}_{success}$	BR_{theory}
50	597	2335	0	0.2036	0.7967	0.9876	0.2003
40	604	2308	20	0.2060	0.7872	0.9629	0.2038
30	670	2105	151	0.2289	0.7194	0.8052	0.2285
20	853	1650	414	0.2924	0.5656	0.5341	0.2866
10	1098	978	838	0.3768	0.3356	0.2596	0.3778
0	1484	0	1419	0.5111	0	0	0.5179

Table 2.38 Simulation obtained values of BR

for normal-modulated arrival process (g)

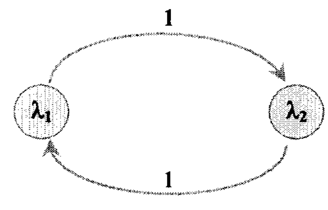
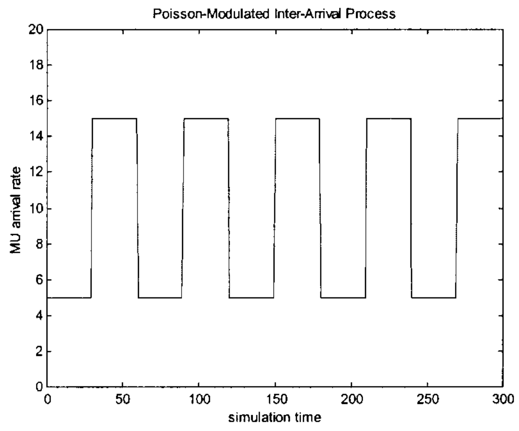


Figure 2.63 Change in MU arrival rates (g)



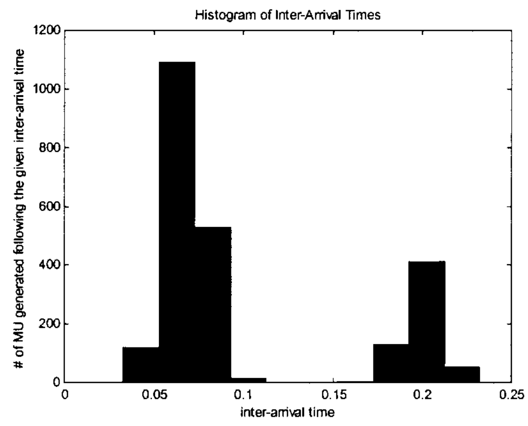


Figure 2.64 MU arrival rates (g)

Figure 2.65 Histogram of MU inter-arrival times (g)

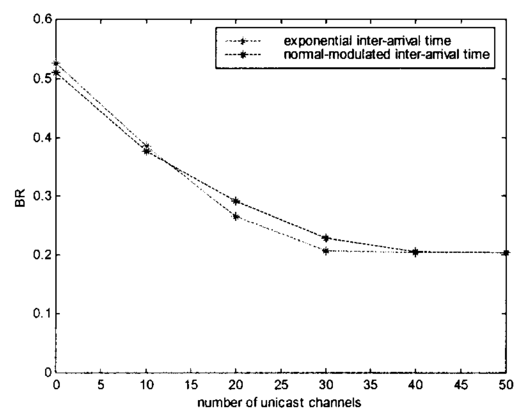


Figure 2.66 Poisson-based BR vs. BR obtained for normal-modulated inter-arrivals (g)

(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with λ=10)

$$\mathrm{RME}(\mathrm{BR}_{\mathrm{simulation}}, \mathrm{BR}_{\mathrm{theory}}) = \frac{1}{6}\sum_{k=1}^{6}\frac{|\,\mathrm{BR}_{\mathrm{theory}}(k) - \mathrm{BR}_{\mathrm{simulation}}(k)\,|}{\mathrm{BR}_{\mathrm{simulation}}(k)} = 1.07\ \% \qquad (2.92)$$

2.6.4.1.h) *Normal-Modulated MU Inter-arrival Time Distribution*

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	$P_{reattempt}$**=0.7**
MU mean reattempt time [time units]	λ^***=0.5**
MU arrival times (see Figure 2.68)	$time_1$ **=N(0.2,0.01),** $time_2$ **=N(0.0677,0.01)**
state occupancy time (see Figure 2.66)	**60 [time units]**
simulation time	**300 [time units]**

Table 2.39 Simulation parameters (2.6.4.1.h)

# of unicast channels	# of users served through broadcast	# of users served through unicast	# of give-up users	BR	UR	$\bar{P}_{success}$	BR_{theory}
50	548	2113	1	0.2058	0.7938	0.9800	0.2014
40	558	2080	24	0.2096	0.7814	0.9482	0.2058
30	623	1888	151	0.2340	0.7092	0.7839	0.2323
20	751	1561	350	0.2821	0.5864	0.5654	0.2786
10	987	945	722	0.3718	0.3561	0.2798	0.3695
0	1378	0	1270	0.5203	0	0	0.5179

Table 2.40 Simulation obtained values of BR

for normal-modulated arrival process (h)

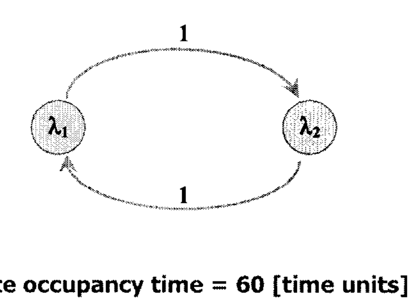


Figure 2.67 Change in MU arrival rates (h)

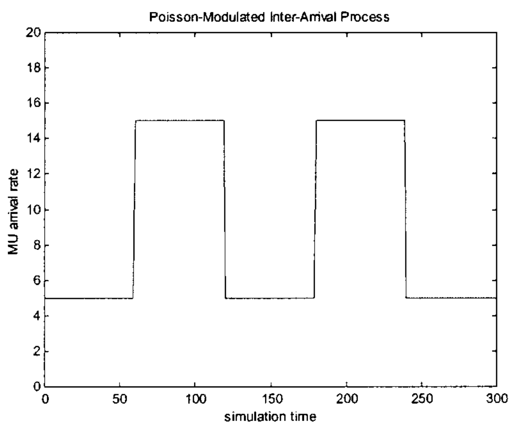


Figure 2.68 MU arrival rates (h)

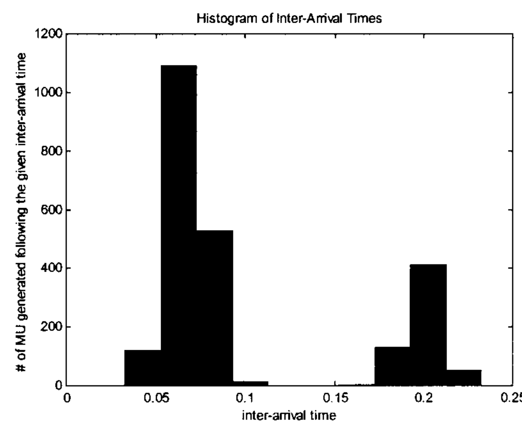


Figure 2.69 Histogram of MU inter-arrival times (h)

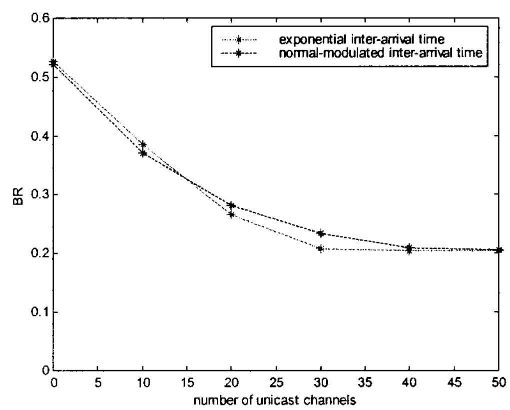


Figure 2.70 Poisson-based BR vs. BR obtained for normal-modulated inter-arrivals (h)

(NOTE: Poisson-based BR is obtained employing a single Poisson arrival process, with $\lambda=10$)

$$\mathrm{RME}(\mathrm{BR}_{\mathrm{simulation}}, \mathrm{BR}_{\mathrm{theory}}) = \frac{1}{6}\sum_{k=1}^{6}\frac{|\mathrm{BR}_{\mathrm{theory}}(k) - \mathrm{BR}_{\mathrm{simulation}}(k)|}{\mathrm{BR}_{\mathrm{simulation}}(k)} = 1.17\ \% \qquad (2.93)$$

2.6.5 Robustness of Analytical BR to Various 'Single-User Inter-Reattempt Time' Distributions

Statement 2.6 Our theoretical analysis of single-cell hybrid WDB systems has been based on the assumption that 'single-user inter-reattempt time' ($\Delta\tau$) is an exponentially distributed random variable with mean $1/\lambda^*$ (see Figure 2.70 and Section 2.2.2). Nevertheless, our analytical BR (see (2.34)) turns out to be a sufficiently accurate approximation of the actual BR, even in the following scenarios:

1) $\Delta\tau$ follows a normal distribution with mean $1/\lambda^*$, in the case of every user
2) $\Delta\tau$ follows a uniform distribution with mean $1/\lambda^*$, in the case of every user
3) a given user is associated with $\Delta\tau_k$ with probability $1/k$ (k=2 or 3), where $\Delta\tau_k$ follows an exponential distribution with mean μ_k and the overall mean of all $\Delta\tau_k$ is $1/\lambda^*$.[23] That is

$$E[\Delta\tau] = E\left[\sum_{i=1}^{k}\frac{1}{k}\Delta\tau_i\right] = \frac{1}{k}E\left[\sum_{i=1}^{k}\Delta\tau_i\right] = \frac{1}{k}\cdot\mu_k = \frac{1}{\lambda^*}$$

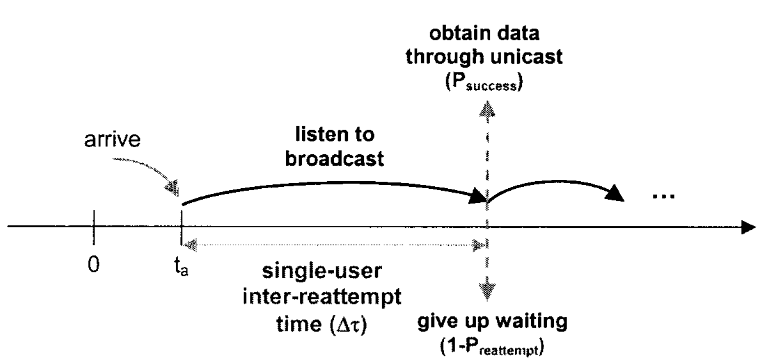


Figure 2.71 Single-user inter-reattempt time

[23] Here we assume it is possible that different users choose to re-issue their requests in different pace, thus exhibiting different levels of re-trial (im)patience.

The simulation results presented in Sections 2.6.5.1 to 2.6.5.2 provide the proof for Statement 2.6. These results were collected under three different sets of system-related simulation parameters (as mentioned in the previous sections), and in each of the three settings the following five scenarios concerning single-user inter-reattempt time were considered (see Table 2.41, 2.42, and 2.43):

scenario (1) the inter-reattempt time associated with each user is of the same exponentially distributed pdf (as assumed in the theoretical discussion of Section 2.3);

scenario (2) the inter-reattempt time associated with each user is of the same normally distributed pdf;

scenario (3) the inter-reattempt time associated with each user is of the same uniformly distributed pdf;

scenario (4) each user is associated with one of two possible exponential inter-reattempt time distributions with probability 1/2;

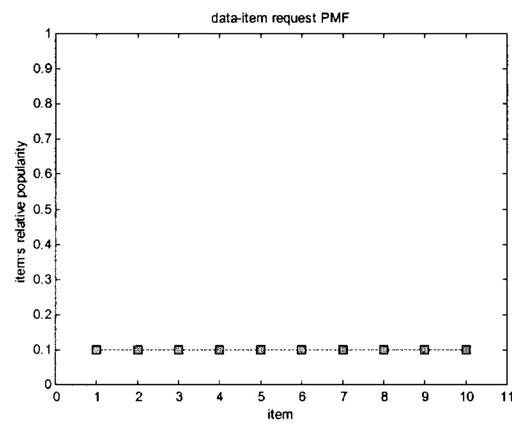
scenario (5) each user is associated with one of three possible exponential inter-reattempt time distributions with probability 1/3.

In each of the given scenarios, by changing the number of available unicast channels, the corresponding simulation based BRs were observed. Consequently, these BRs were compared against BR as observed in scenario (1), which was considered equivalent to the analytical BR (see Sections 2.6.1 to 2.6.4). As Figures 2.72 to 2.77 indicate, in each of the three scenarios the obtained curves BR=BR(# of unicast channels) were very close to each other, regardless of the underlying single-user inter-reattempt time distribution. Hence, Figures 2.72 to 2.77 provide the proof for our claim in Statement 2.6.

2.6.5.1 Various Single-User Inter-Reattempt Time Distributions: Experiment 1

data-item request probabilities	**p=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Simulation parameters (1)



Data-item request probabilities (1)

# of unicast channels	**$BR_{simulation}$ for exp. distributed** single-user inter-reattempt time (mean=2)	**$BR_{simulation}$ for normally distributed** single-user inter-reattempt time (mean=2 variance=0.4)	**$BR_{simulation}$ for uniformly distributed** single-user inter-reattempt time (min=1.8 max=2.2)	**$BR_{simulation}$ for exp. distributed** single-user inter-reattempt time 2 levels of impatience ($mean_1$=0.6667*2 $mean_2$=1.3334*2)	**$BR_{simulation}$ for exp. distributed** single-user inter-reattempt time 3 levels of impatience ($mean_1$=0.5*2 $mean_2$=1.0*2 $mean_3$=1.5*2)
50	0.2036	0.1946	0.1951	0.1911	0.1943
40	0.2036	0.1946	0.1951	0.1911	0.1943
30	0.2073	0.2016	0.2005	0.1971	0.2010
20	0.2656	0.2633	0.2689	0.2560	0.2528
10	0.3854	0.3845	0.3927	0.3646	0.3626
0	0.5262	0.5432	0.5413	0.4912	0.4835

Table 2.41 Simulation obtained BRs in Experiment 1

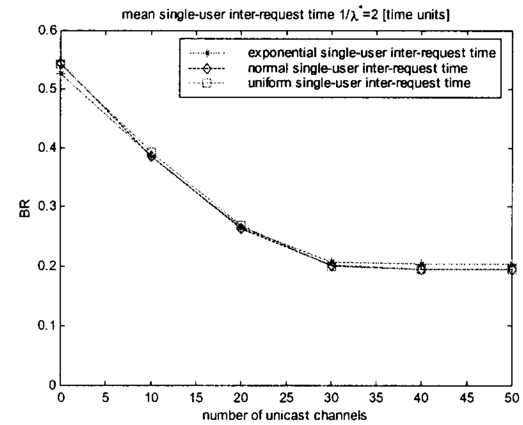


Figure 2.72 BR in case of exp. distributed vs. normally distributed vs. uniformly distributed single-user inter-reattempt time – Experiment 1

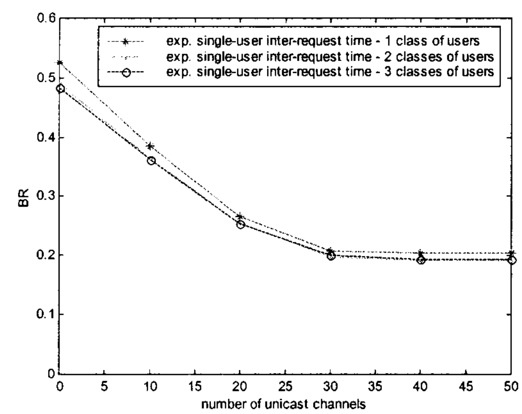
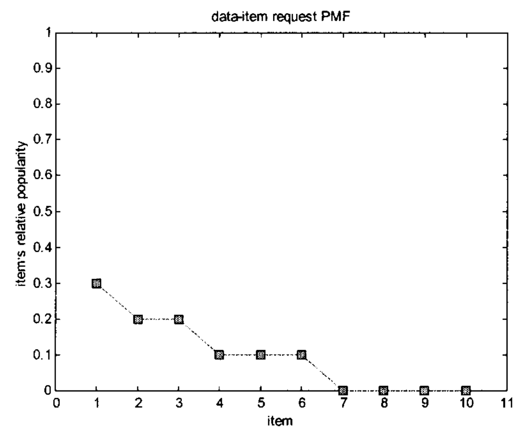


Figure 2.73 BR in case of exp. distributed single-user inter-reattempt time in case of 1, 2 or 3 levels of user impatience – Experiment 1

2.6.5.2 Various Single-User Inter-Reattempt Time Distributions: Experiment 2

data-item request probabilities	**p=[0.3, 0.2, 0.2, 0.1, 0.1, 0.1, 0.0, 0.0, 0.0, 0.0]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l./p=[3.33 5.0 5.0 10.0 10.0 10.0 Inf Inf Inf Inf]**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5 [users per time unit]**
mean MU arrival rate	**λ=10 [user per time unit]**
simulation time	**300 [time units]**

Simulation parameters (2)



Data-item request probabilities (2)

# of unicast channels	$BR_{simulation}$ for **exp. distributed** single-user inter-reattempt time (mean=2)	$BR_{simulation}$ for **normally distributed** single-user inter-reattempt time (mean=2 variance=0.4)	$BR_{simulation}$ for **uniformly distributed** single-user inter-reattempt time (min=1.8 max=2.2)	$BR_{simulation}$ for **exp. distributed** single-user inter-reattempt time 2 levels of impatience ($mean_1$=0.6667*2 $mean_2$=1.3334*2)	$BR_{simulation}$ for **exp. distributed** single-user inter-reattempt time 3 levels of impatience ($mean_1$=0.5*2 $mean_2$=1.0*2 $mean_3$=1.5*2)
50	0.3461	0.3930	0.4107	0.3330	0.3345
40	0.3461	0.3930	0.4107	0.3330	0.3345
30	0.3477	0.3946	0.4117	0.3363	0.3372
20	0.3795	0.4220	0.4360	0.3764	0.3728
10	0.5020	0.5476	0.5614	0.4889	0.4833
0	0.6564	0.7193	0.7356	0.6402	0.6313

Table 2.42 Simulation obtained BRs in Experiment 2

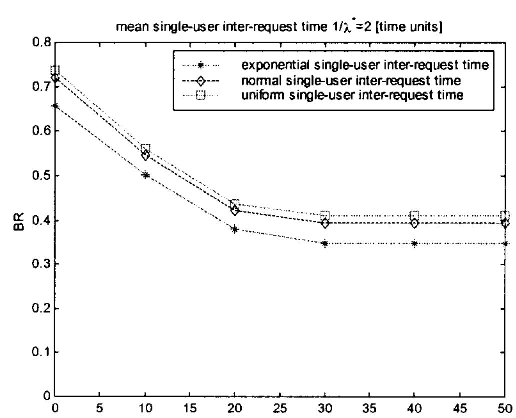


Figure 2.74 BR in case of exp. distributed vs. normally distributed vs. uniformly distributed single-user inter-reattempt time – Experiment 2

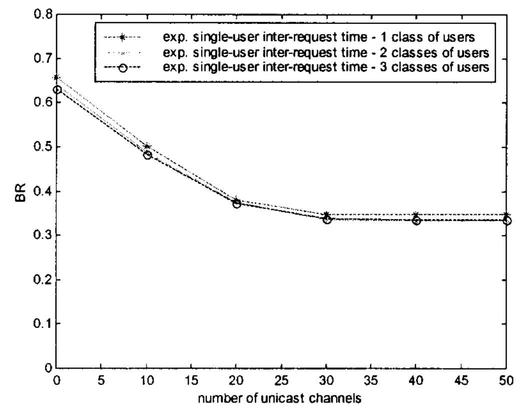
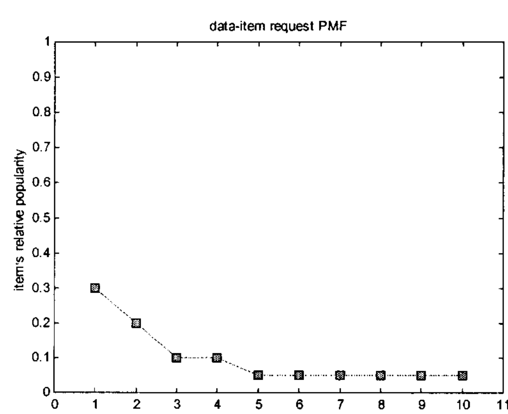


Figure 2.75 BR in case of exp. distributed single-user inter-reattempt time in case of 1, 2 or 3 levels of user impatience – Experiment 2

2.6.5.3 Various Single-User Inter-Reattempt Time Distributions: Experiment 3

data-item request probabilities	**p=[0.3, 0.2, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05]**
data-item lengths [time units]	**l=[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**
data-item broadcast periods [time units]	**s=l/p=[3.33 5.0 10.0 10.0 20.0 20.0 20.0 20.0 20.0 20.0]**
MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**

Simulation parameters (3)



Data-item request probabilities (3)

# of unicast channels	$BR_{simulation}$ for exp. distributed single-user inter-reattempt time (mean=2)	$BR_{simulation}$ for normally distributed single-user inter-reattempt time (mean=2 variance=0.4)	$BR_{simulation}$ for uniformly distributed single-user inter-reattempt time (min=1.8 max=2.2)	$BR_{simulation}$ for exp. distributed single-user inter-reattempt time 2 levels of impatience ($mean_1$=0.6667*2 $mean_2$=1.3334*2)	$BR_{simulation}$ for exp. distributed single-user inter-reattempt time 3 levels of impatience ($mean_1$=0.5*2 $mean_2$=1.0*2 $mean_3$=1.5*2)
50	0.2905	0.3175	0.3333	0.2662	0.2621
40	0.2905	0.3175	0.3333	0.2662	0.2621
30	0.2922	0.3218	0.3350	0.2669	0.2671
20	0.3344	0.3655	0.3720	0.3192	0.3144
10	0.4369	0.4759	0.4878	0.4290	0.4223
0	0.5736	0.6115	0.6255	0.5528	0.5479

Table 2.43 Simulation obtained BRs in Experiment 3

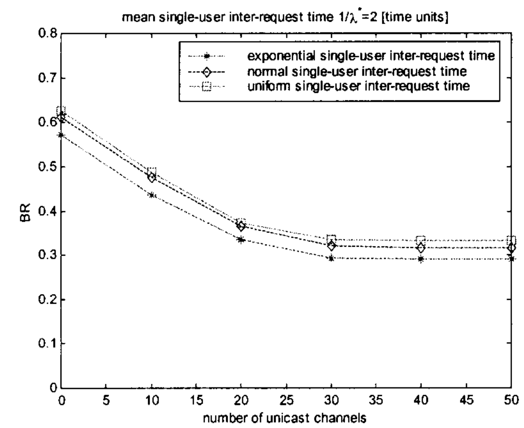


Figure 2.76 BR in case of exp. distributed vs. normally distributed vs. uniformly distributed single-user inter-reattempt time – Experiment 3

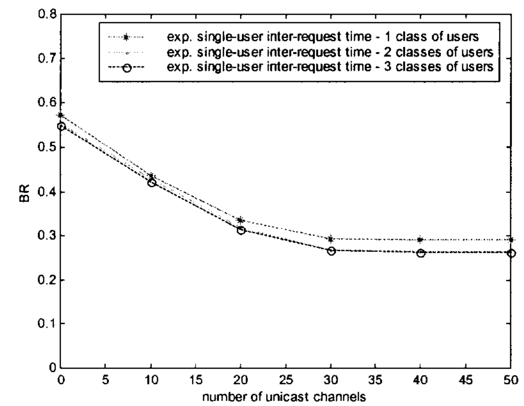


Figure 2.77 BR in case of exp. distributed single-user inter-reattempt time in case of 1, 2 or 3 levels of user impatience – Experiment 3

2.7 Analytical Sensitivity Analysis

At the end of this chapter, we present a series of analytical results concerning the gradient based sensitivity analysis of the most significant system performance measure BR (see Section 2.3), with respect to three main system and user-behaviour parameters: $P_{success}$, $P_{reattempt}$, and λ^*. In particular, we obtain the exact analytical expression for the partial derivatives of BR with respect to $P_{success}$, $P_{reattempt}$, and λ^*. Based on these expressions, we observe how the variation in either of the three parameters affects the actual value of BR. Subsequently, this enables us to make some general conclusions regarding the sensitivity of an approximation involving our analytical expression for BR, to possibly inaccurate knowledge of $P_{success}$, $P_{reattempt}$, and λ^*.

2.71 Sensitivity of BR with Respect to $P_{success}$

Based on (2.34), we obtain the partial derivative of BR with respect to $P_{success}$, as given in (2.94).

$$\begin{aligned}\frac{\partial BR}{\partial P_{success}} = & \frac{-P_{reattempt}}{\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)^2}\sum_{i=1}^{M}\frac{p(i)}{s(i)}\left(1-e^{-\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)s(i)}\right)+ \\ & +\frac{P_{reattempt}}{(1-P_{reattempt}(1-P_{success}))}\sum_{i=1}^{M}p(i)\cdot e^{-\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)s(i)}\end{aligned} \tag{2.94}$$

From (2.94), it is evident that the actual value of $\partial BR/\partial P_{success}$ at any given $P_{success}$ is dependent on the values of p(i), i=1,..,M, s(i), i=1,..,M, $P_{reattempt}$ and λ^*. Hence, we conclude that BR exhibits different forms of sensitivity with respect to $P_{success}$, for every different data-item request distribution (p(i), i=1,..,M), broadcast schedule (s(i), i=1,..,M), and parameters $P_{reattempt}$ and λ^*, respectively.

Nevertheless, let us consider a special case, in which all data-items are equally popular (p(i)=1/M (∀i=1,..,M)), and broadcasted in identical intervals of time s(i)=s(j) (∀ i,j=1,..,M). Under such assumptions, (2.94) gets transformed to (2.95). Subsequently, from (2.95), the graphical representation of BR's sensitivity with respect to $P_{success}$, under varying values of s(i), λ^*, and $P_{reattempt}$, can be obtained, as illustrated in Figures 2.78 to 2.80.

$$\frac{\partial BR}{\partial P_{success}} = \frac{-P_{reattempt}}{\lambda^*\left(1-P_{reattempt}\left(1-P_{success}\right)\right)^2 s(i)}\left(1-e^{-\lambda^*\left(1-P_{reattempt}\left(1-P_{success}\right)\right)s(i)}\right)+ \\ +\frac{P_{reattempt}}{\left(1-P_{reattempt}\left(1-P_{success}\right)\right)}e^{-\lambda^*\left(1-P_{reattempt}\left(1-P_{success}\right)\right)s(i)} \qquad (2.95)$$

Based on Figures 2.87 to 2.80, we observe that $\partial BR/\partial P_{success}$ is negative, for all possible values of $P_{success}$, s(i), $P_{reattempt}$, and λ^*. Hence, we conclude that an increase in $P_{success}$ will result in a decrease in BR, under any given data-item request distribution (p(i), i=1,..,M), broadcast schedule (s(i), i=1,..,M), and parameters $P_{reattempt}$ and λ^*. We also observe that BR exhibits considerable sensitivity with respect to $P_{success}$ for larger values of s(i) (see Figure 2.78) and λ^* (see Figure 2.79), while at the same time it appears very insensitive to changes in $P_{success}$ for low values of $P_{reattempt}$ (see Figure 2.80).

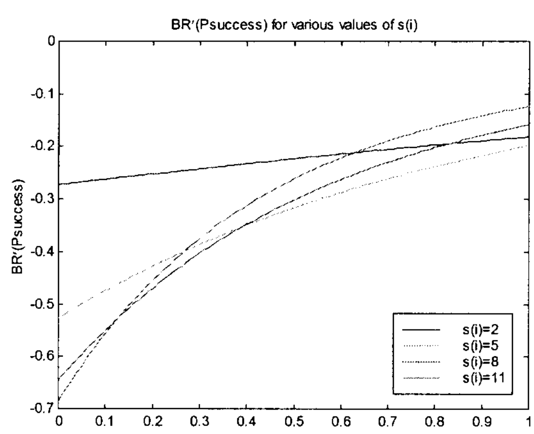


Figure 2.78 BR'($P_{success}$) in case of variable s(i), and $P_{reattempt}$=0.7, λ^*=0.5

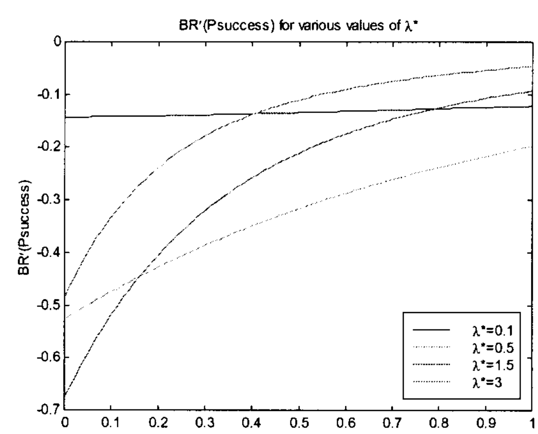


Figure 2.79 BR'($P_{success}$) in case of variable λ^*, and s(i)=5, $P_{reattempt}$=0.7

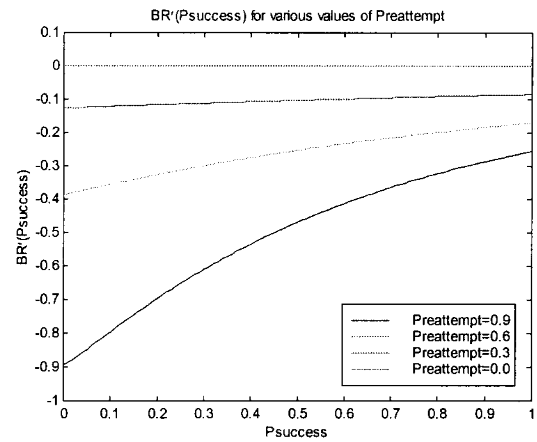


Figure 2.80 BR'($P_{success}$) in case of variable $P_{reattempt}$, and s(i)=5, λ^*=0.5

2.7.2 Sensitivity of BR with Respect to $P_{reattempt}$

The partial derivative of BR with respect to $P_{reattempt}$ is given in (2.96). Similar as in the preceding section, (2.96) suggests that the actual value of $\partial BR/\partial P_{reattempt}$, at any given $P_{reattempt}$, is also dependent on p(i), i=1,..,M, s(i), i=1,..,M, $P_{success}$ and λ^*. Nevertheless, we again consider the special case when p(i)=1/M ($\forall$i=1,..,M), and s(i)=s(j) ($\forall$ i,j=1,..,M) (see (2.97)), and subsequently obtain the graphs shown in Figures 2.81 to 2.83.

$$\frac{\partial BR}{\partial P_{reattempt}} = \frac{(1-P_{success})}{\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)^2}\sum_{i=1}^{M}\frac{p(i)}{s(i)}\left(1-e^{-\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)s(i)}\right) - \frac{(1-P_{success})}{(1-P_{reattempt}(1-P_{success}))}\sum_{i=1}^{M}p(i)\cdot e^{-\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)s(i)} \quad (2.96)$$

$$\frac{\partial BR}{\partial P_{reattempt}} = \frac{(1-P_{success})}{\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)^2 s(i)}\left(1-e^{-\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)s(i)}\right) - \frac{(1-P_{success})}{(1-P_{reattempt}(1-P_{success}))}e^{-\lambda^*\left(1-P_{reattempt}(1-P_{success})\right)s(i)} \quad (2.97)$$

Based on Figures 2.81 to 2.83 we observe that $\partial BR/\partial P_{reattempt}$ is positive, for all possible values of $P_{reattempt}$, s(i), $P_{success}$, and λ^*. Thus, BR turns out to be a monotonically increasing function of $P_{reattempt}$, under any given data-item request distribution (p(i), i=1,..,M), broadcast schedule (s(i), i=1,..,M), and parameters $P_{success}$ and λ^*. We also observe that BR appears least sensitive to changes in $P_{reattempt}$ under large values of $P_{success}$ (see Figure 2.82).

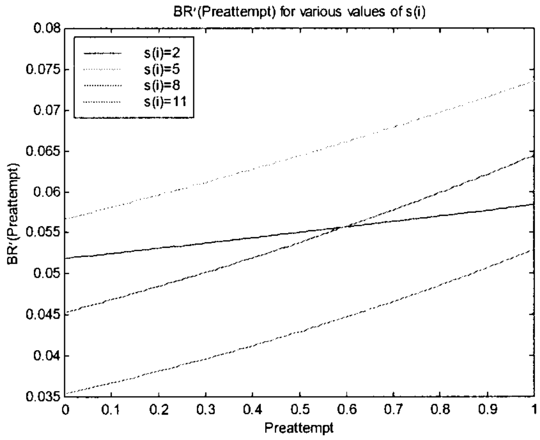


Figure 2.81 BR'($P_{reattempt}$) in case of variable s(i), and $P_{success}$=0.8, λ^*=0.5

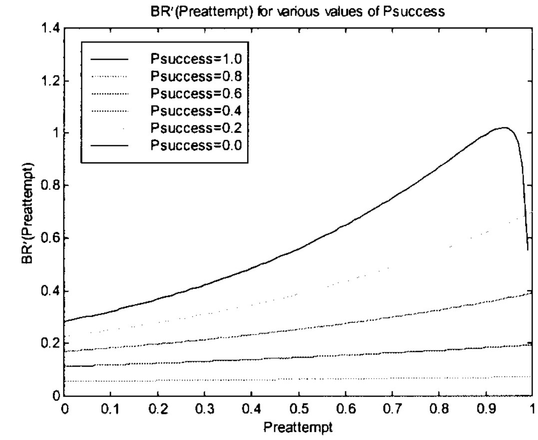


Figure 2.82 BR'($P_{reattempt}$) in case of variable $P_{success}$, and s(i)=5, λ^*=0.5

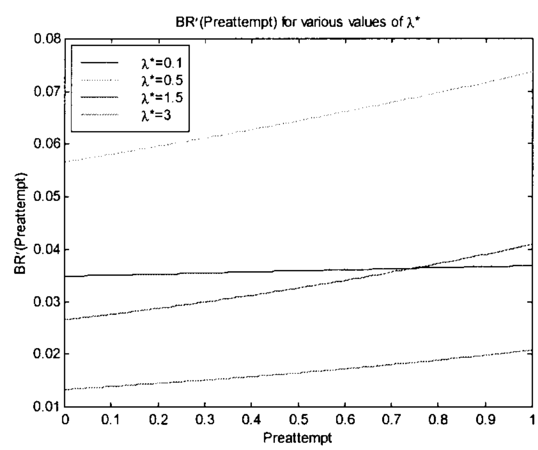


Figure 2.83 BR'($P_{reattempt}$) in case of variable λ^*, and $P_{success}$=0.8, s(i)=5

2.7.3 Sensitivity of BR with Respect to λ^*

The partial derivative of BR with respect to λ^* is given in (2.98). Similar as in the two earlier cases, (2.98) suggests that the value of $\partial BR/\partial\lambda^*$, at any given λ^*, also depends on the values of other system and user behaviour parameters. We again consider the special case when p(i)=1/M ($\forall$i=1,..,M), and s(i)=s(j) ($\forall$ i,j=1,..,M) (see (2.99)), and subsequently obtain the graphs shown in Figures 2.84 to 2.86. Considering that $\partial BR/\partial\lambda^*$ is negative, in all three cases presented in Figures 2.84 to 2.86, we conclude that BR is a monotonically increasing function of λ^*, under any given data-item request distribution (p(i), i=1,..,M), broadcast schedule (s(i), i=1,..,M), and parameters $P_{success}$ and $P_{reattempt}$. Moreover, based on these figures, we conclude that BR is considerably more sensitive to variations in λ^* in the lower than in the higher range of this variable.

$$\frac{\partial BR}{\partial\lambda^*}=\frac{-1}{(1-a)(\lambda^*)^2}\sum_{i=1}^{M}\frac{p(i)}{s(i)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)+\frac{1}{\lambda^*}\sum_{i=1}^{M}p(i)\cdot e^{-\lambda^*(1-a)s(i)} \tag{2.98}$$

$$\frac{\partial BR}{\partial\lambda^*}=\frac{-1}{(1-a)(\lambda^*)^2 s(i)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)+\frac{1}{\lambda^*}e^{-\lambda^*(1-a)s(i)} \tag{2.99}$$

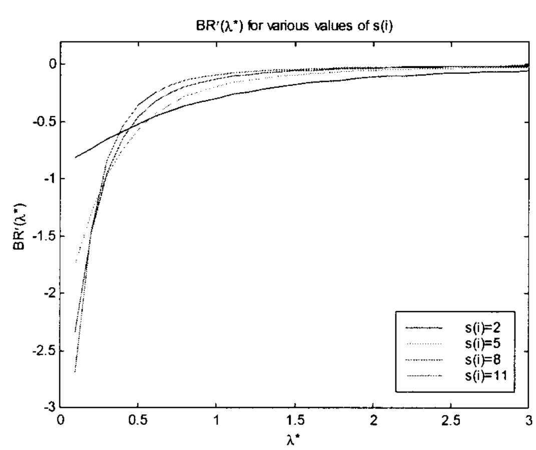


Figure 2.84 BR'(λ^*) in case of variable s(i), and $P_{reattempt}$=0.7, $P_{success}$=0.8

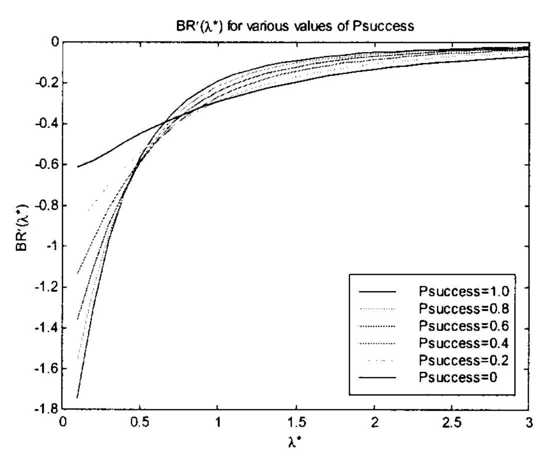


Figure 2.85 BR'(λ^*) in case of variable $P_{success}$, and $P_{reattempt}$=0.7, s(i)=5

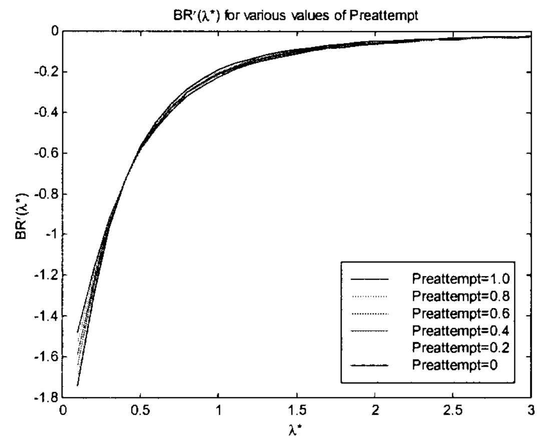


Figure 2.86 BR'(λ^*) in case of variable $P_{reattempt}$, and $P_{success}$=0.8, s(i)=5

2.8 Summary of Main Contributions

In this chapter, we have presented a general theoretical framework for the analysis of single-cell hybrid WDB systems in the presence of users retrials. Through a series of simulation results, we have proven the validity and robustness of the given framework, thus demonstrating its overall usefulness in the analysis of a wide range of real-world WDB system scenarios.

Chapter 3

Optimal Broadcast Scheduling Scheme in Single-Cell Hybrid WDB Systems

Overview

The maximization of BR and the minimization of UR, LR, WT, RP, and WP appear as the most natural objectives of a hybrid single-cell WDB based system. In the preceding chapter we have shown that the broadcast scheduling scheme (S) capable of accomplishing one of the given objectives, inherently accomplishes all the other ones, and therefore can be considered optimal from the system's point of view. In this chapter, we extend on the discussion from Chapter 2 and show the following:

Section 3.1[1] First, we prove that the search for the actual value of $S_{optimal}$ is a rather complex, non-linear double inequality-constrained optimization problem, without a tractable closed-form solution. However, by using the method of slack variable, we show that the given problem can be transformed into a much simpler, single equality-constrained optimization problem. Consequently, by demonstrating the (strict) convexity

[1] The work presented in this section extends on the work presented in [*Jiang_1*].

of both, the problem's feasible region and its main cost function, we prove the existence of a single $S_{optimal}$ in any particular WDB based system.

Section 3.2 By exploiting some specific properties of the main cost function, we show that the given optimization problem (stated in Section 3.1) can be further simplified, and an approximate closed from expression for $S_{optimal}$ can be obtained. Based on the given closed-form expression, we propose a CPU conserving algorithm for fast estimation of $S_{optimal}$. Subsequently, through a series of experimental results, we verify that the proposed algorithm requires minimum computation, while providing performance almost identical to $S_{optimal}$ obtained through numerical estimation.

3.1 Optimal Broadcast Schedule - Implicit-form Solution

Definition 3.1 The optimal broadcast scheduling scheme[2] of a generalized WDB system ($S_{optimal}=(s(1)^*,s(2)^*,.., s(M)^*)$) corresponds to the point in R^M which maximizes BR and minimizes UR, LR, WT, RP, and WP.[3]

According to Definition 3.1, the search for $S_{optimal}$ appears to be a multi-objective optimization problem. However, it has been shown that the minimization of UR, LR, WT, RP, and WP are inherently accomplished through the maximization of BR (see Sections 2.3.2 through 2.3.7). Hence, $S_{optimal}$ can actually be found by solving a much simpler, single-objective optimization problem, as defined in (3.1).

$$S_{optimal} = \{S^* \in R^M \mid BR(S^*) = \max BR(S), \text{ where } S = (s(1), s(2),.., s(M)) \in R^M\} \qquad (3.1)$$

However, the physical nature of the problem imposes two additional constraints to the definition of $S_{optimal}$, i.e. to (3.1). First, every $s(i)^*$, i=1,..,M, from $S_{optimal}$ (see Definition 3.1) must be greater than or equal to l(i)[4], where l(i)>0. Namely, s(i) represents the length of the inter-broadcast interval associated with item(i) (see Section 2.2). Thus, if a single broadcasting of item(i) takes l(i) [time units], then item(i) can be broadcasted at most every s(i)≥l(i) [time units]. Second, the broadcast channel is assumed to be of a finite capacity. Accordingly, if item(i) occupies a fraction of the channel's capacity that equals to l(i)/s(i) (i=1,..,M), then the sum of all l(i)/s(i), i=1,..,M, should be less than or equal to one.

[2] Please note, we use the terms 'broadcast schedule' and 'broadcast scheduling scheme' interchangeably.
[3] The motivation for Definition 3.1 can be found in Section 2.3.8.
[4] Note, l(i)=*l*(i)/r [time units], where *l*(i) is the length of item(i) [bits], while r is the datarate of the broadcast channel [bits/time unit]. Thus, l(i) represents the length of item(i) in time units.

Thus, due to the imposed constraints, the problem of finding $S_{optimal}$ turns into an inequality constrained single-objective optimization problem [*Bertsekas*], as stated below:

$$\text{maximize} \quad BR(s(1), s(2), .., s(M)) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^*(1-a)s(i)}) \tag{3.2.a}$$

$$\text{subject to} \quad C(s(1), s(2), .., s(M)) = 1 - \sum_{i=1}^{M} \frac{l(i)}{s(i)} \geq 0 \tag{3.2.b}$$

$$\text{and} \quad g_i(s(1), s(2), .., s(M)) = s(i) - l(i) \geq 0, \quad i = 1, .., M \tag{3.2.c}$$

where $l(i) \geq 0, \quad i = 1, .., M$.

Theorem 3.1 A non-trivial solution to (3.2), $S_{optimal}=(s(1)^*, s(2)^*, .., s(M)^*)$, lies on the boundary of (3.2.b) and in the interior of (3.2.c).

Put another way, a non-trivial solution to (3.2) can be obtained as a solution to a simpler, single equality-constrained optimization problem, as stated below

$$\text{maximize} \quad BR(s(1), s(2), .., s(M)) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^*(1-a)s(i)}) \tag{3.3.a}$$

$$\text{subject to} \quad C(s(1), s(2), .., s(M)) = 1 - \sum_{i=1}^{M} \frac{l(i)}{s(i)} = 0 \tag{3.3.b}$$

Proof of Theorem 3.1

According to the method of slack variable [*Wismer*], the solution to (3.2) can be obtained as a solution to the following equality constrained single objective optimization problem

$$\text{maximize} \quad BR(s(1),..,s(M)) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{s(i)} (1 - e^{-\lambda^*(1-a)s(i)}) \tag{3.4.a}$$

$$\text{subject to} \quad C'(s(1),..,s(M),k) = 1 - \sum_{i=1}^{M} \frac{l(i)}{s(i)} - k^2 = 0 \tag{3.4.b}$$

$$\text{and} \quad g'_i\,(s(1),..,s(M),\theta_i) = s(i) - l(i) - \theta_i^{\,2} = 0, \quad i = 1,..,M \tag{3.4.c}$$

where $l_i \geq 0$ (i=1,..,M), and k and θ_i (i=1,..,M) are unknown real-valued parameters, i.e. the so-called slack variables. Furthermore, the Lagrangian function corresponding to (3.4) is of the following form

$$L(\mathbf{s}, K, k, \boldsymbol{\lambda}, \boldsymbol{\theta}) = BR(\mathbf{s}) + KC'(\mathbf{s}, k) + \sum_{i=1}^{M} \lambda_i g'_i\,(\mathbf{s}, \theta_i) \tag{3.5}$$

where $\mathbf{s} \equiv (s(1),..,s(M))$, $\boldsymbol{\lambda} \equiv (\lambda_1,..,\lambda_M)$ and $\theta \equiv (\theta_1,..,\theta_M)$. Note, K and λ_i (i=1,..,M) in (3.5) represent unknown real-valued parameters, the so-called Lagrange Multipliers.

From (3.5), the necessary conditions concerning the optimal point corresponding to (3.2) are obtained, as given in (3.6.a) through (3.6.e).

$$\frac{\partial L(\mathbf{s}, K, k, \boldsymbol{\lambda}, \boldsymbol{\theta})}{\partial s(i)} = \frac{\partial BR(\mathbf{s})}{\partial s(i)} - K \frac{l(i)}{s(i)^2} + \lambda(i) = 0 \tag{3.6.a}$$

$$\frac{\partial L(\mathbf{s}, K, k, \boldsymbol{\lambda}, \boldsymbol{\theta})}{\partial K}, = 1 - \sum_{i=1}^{M} \frac{l(i)}{s(i)} - k^2 = 0 \tag{3.6.b}$$

$$\frac{\partial L(\mathbf{s},K,k,\boldsymbol{\lambda},\boldsymbol{\theta}),}{\partial k} = -2Kk = 0 \qquad (3.6.c)$$

$$\frac{\partial L(\mathbf{s},K,k,\boldsymbol{\lambda},\boldsymbol{\theta}),}{\partial \lambda_i} = s(i) - l(i) - \theta_i^{\,2} = 0, \quad i = 1,..,M \qquad (3.6.d)$$

$$\frac{\partial L(\mathbf{s},K,k,\boldsymbol{\lambda},\boldsymbol{\theta}),}{\partial \theta_i} = -2\lambda_i\theta_i = 0, \quad i = 1,..,M \qquad (3.6.e)$$

Special Case (1) Let us assume a trivial case: $\exists j \in \{1,..,M\}$ such that s(j)=l(j). Under this assumption, constraint (3.6.d), for i=j, implies $\theta_j=0$, while subsequently (3.6.b) gets transformed to

$$\sum_{\substack{i=1\\ i\neq j}}^{M} \frac{l(i)}{s(i)} = -k^2 \qquad (3.7)$$

Since $l(i)\geq 0$, $\forall i=1,..,M$, it can be easily verified that (3.7) holds if either $s(i)\rightarrow\infty$, $\forall i\in\{1,..M\}\backslash\{j\}$, (and consequently k=0), or some s(i)<0. Nevertheless, s(i)<0 is not a part of the constrained set; moreover it contradicts (3.6.d), since s(i)<0 would imply $\theta_i^2<0$. Hence, $S_{optimal}=\{(s(1),..,s(i),..,s(M))|\ s(i)=l(i)\ (i=j),\ s(i)=\infty\ (i\neq j)\}$, i.e. $S_{optimal}=(\infty,..,l(j),..,\infty)$ remains as the only valid solution to (3.4). In practical terms, such a solution implies that a single item (item(j)) needs to be broadcasted continually, while all the other items should be completely excluded from the broadcast schedule.

Special Case (2) Now, let us assume a more realistic case s(i)>l(i) for $\forall i=1,..,M$. Under this assumption, constraint (3.6.d) implies $\theta_i\neq 0$ (i=1,..,M). Consequently, from (3.6.e), $\lambda_i=0$ (i=1,..,M) is obtained, which suggests that the set of inequality constraints (3.2.c) can be neglected. At the same time, $\lambda_i=0$ (i=1,..,M) in (3.6.a)[5] implies $K\neq 0$, while $K\neq 0$ in (3.6.c) implies k=0.

---

[5] Please note that, it can be easily verified that $\partial BR(\mathbf{s})/\partial s_i\neq 0$ for $\forall \mathbf{s}\in R^M$.

Based on the above, we conclude that under assumption s(i)>l(i), i=1,..,M, it turns out that, in (3.4.b) k=0 while in (3.5) λ_i=0 (i=1,..,M), thus the optimal point for (3.2) can actually be obtained as a solution to the single equality–constrained optimization problem as stated in (3.3).

End of Proof

Theorem 3.2 In any given generalized WDB system, there exists a single, i.e. unique, optimal broadcast scheduling scheme ($S_{optimal}$) (defined by (3.2)).

Proof of Theorem 3.2

By employing the following substitution

$$\frac{l(i)}{s(i)} = z(i) \tag{3.8}$$

and recalling l(i)≥0 (i=1,..,M), (3.2) becomes

$$\text{maximize} \quad BR(z(1), z(2), .., z(M)) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)z(i)}{l(i)} \left(1 - e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}}\right) \tag{3.9.a}$$

$$\text{subject to} \quad g_0(z(1), z(2), .., z(M)) = \sum_{i=1}^{M} z(i) \le 1 \tag{3.9.b}$$

$$\text{and} \quad g_i(z(1), z(2), .., z(M)) = z(i) \le 1, \quad i = 1, .., M \tag{3.9.c}$$

$$\text{and} \quad g_i(z(1), z(2), .., z(M)) = z(i) \ge 0, \quad i = 1, .., M \tag{3.9.d}$$

Please note that (3.9) represents a problem equivalent to (3.2). Namely, once a solution that satisfies (3.9) is found, then through a simple inverse mapping based on (3.8) a solution to (3.2) can be found as well.

In (3.9), each of the inequality constraints ((3.9.b), (3.9.c) and (3.9.d)) represents a halfspace in R^M (see [*Boyd*]). Hence, the feasible region of the given optimization problem corresponds to the intersection of the three halfspaces. By recalling that, in general, halfspaces are special cases of convex sets, and that the intersection of convex sets is also convex [*Boyd*], we conclude that the feasible region corresponding to (3.9) is a convex set itself.

Now, let us represent (3.9.a) in the following form

$$BR(z(1), z(2), .., z(M)) = \sum_{i=1}^{M} br_i(z(i)) \tag{3.10}$$

where

$$br_i(z(i)) = \frac{1}{\lambda^*(1-a)} \frac{p(i)z(i)}{l(i)} (1 - e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}}) \tag{3.11}$$

(Note, even for a=1 or λ^*=0, $br_i(z(i))$ (i=1,..,M) are of finite values and it is possible to find a solution to (3.9). Namely, it can be easily proven that $\lim_{a\to 1} br_i(z(i)) = p(i)$ and $\lim_{\lambda^*\to 0} br_i(z(i)) = p(i)$.)

The first and second derivative tests on (3.11), as shown in (3.12) and (3.13) respectively, prove that $br_i(z(i))$, i=1,..,M, are strictly monotonically increasing and strictly concave functions on the feasible region of (3.9).

$$\frac{\partial br_i(z(i))}{\partial z(i)} = \frac{p(i)}{\lambda^*(1-a)l(i)} \left(1 - e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}} - \lambda^*(1-a)\frac{l(i)}{z(i)} e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}} \right) = \tag{3.12}$$

$$= \frac{p(i)}{\lambda^*(1-a)l(i)} \left(1 - \frac{1+\lambda^*(1-a)\frac{l(i)}{z(i)}}{e^{\lambda^*(1-a)\frac{l(i)}{z(i)}}} \right) = \frac{p(i)}{\lambda^*(1-a)l(i)} \left(1 - \frac{1+\lambda^*(1-a)\frac{l(i)}{z(i)}}{1+\lambda^*(1-a)\frac{l(i)}{z(i)} + \frac{1}{2!}\left(\lambda^*(1-a)\frac{l(i)}{z(i)}\right)^2 + \ldots} \right) > 0$$

$$\frac{\partial^2 br_i(z(i))}{\partial z(i)^2} = \tag{3.13}$$

$$= \frac{p(i)}{\lambda^*(1-a)l(i)}\left(-\lambda^*(1-a)\frac{l(i)e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}}}{(z(i))^2} + \lambda^*(1-a)\frac{l(i)e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}}}{(z(i))^2} - \left(\lambda^*(1-a)\right)^2\frac{(l(i))^2 e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}}}{(z(i))^3}\right) =$$

$$= \frac{p(i)}{\lambda^*(1-a)l(i)}\left(-\left(\lambda^*(1-a)\right)^2\frac{(l(i))^2}{(z(i))^3}e^{-\lambda^*(1-a)\frac{l(i)}{z(i)}}\right) < 0$$

According to (3.10), the cost function of (3.9) (i.e BR(z(1),..,z(M))) corresponds to a sum of strictly monotonically increasing and strictly concave functions ($br_i(z(i))$) on the feasible region defined by (3.9.b) through (3.9.d). Hence, BR(z(1),..,z(M)) is a strictly monotonically increasing and strictly concave function itself (see [*Boyd*]).

Based on the above discussion, (3.9) turns out to be a problem of finding the maximum of a strictly concave function over a convex feasible region. In general, such a problem is known to have a single, i.e. unique solution [*Boyd*]. From the uniqueness of the solution to (3.9), the uniqueness of the solution to (3.2) can be also claimed. Thus, we conclude that in any given generalized WDB system there exists a unique optimal broadcast scheduling scheme ($S_{optimal}$) that satisfies (3.2).

End of Proof

Remark 3.1 The problem of finding a real-domain closed form expression for $S_{optimal}$=(s(1),..,s(M)), that would satisfy (3.2), is unfeasible.

According to Theorem 3.1, $S \in R^M$ that maximizes BR (3.2.a), under constraints (3.2.b), can be obtained as a solution to the set of differential equations given in (3.14), i.e. (3.15).

$$\nabla BR(s(1), s(2), .., s(M)) + K \cdot \nabla C(s(1), s(2), .., s(M)) = 0 \tag{3.14}$$

$$\frac{\partial BR(s(1),s(2),..,s(M))}{\partial s(i)} + K \frac{\partial C(s(1),s(2),..,s(M))}{\partial s(i)} = 0, \quad i = 1,..,M \tag{3.15}$$

(In (3.14), i.e. (3.15), K is an unknown scalar, i.e. the so-called Lagrange Multiplier.) By employing the actual expressions for BR and C, i.e. (3.2.a) and (3.2.b) respectively, (3.15) gets transformed to

$$\frac{\partial}{\partial s(i)}\left(\frac{p(i)}{s(i)\lambda^*(1-a)}(1 - e^{-\lambda^*(1-a)s(i)}) \right) + K \frac{\partial}{\partial s(i)}\left(\frac{l(i)}{s(i)} \right) = 0, \quad i = 1,..,M \tag{3.16}$$

After finding the left-hand side derivatives, (3.16) becomes

$$\frac{p(i)}{\lambda^*(1-a)} \frac{\lambda^*(1-a)e^{-\lambda^*(1-a)s(i)} \cdot s(i) - 1 + e^{-\lambda^*(1-a)s(i)}}{s(i)^2} - K \frac{l(i)}{s(i)^2} = 0, \quad i = 1,..,M \tag{3.17}$$

Finally,

$$\frac{p(i)}{l(i)}\left(s(i)\lambda^*(1-a)e^{-\lambda^*(1-a)s(i)} + e^{-\lambda^*(1-a)s(i)} - 1 \right) = K\lambda^*(1-a) = \text{const}, \quad i = 1,..,M \tag{3.18}$$

Clearly, (3.18) cannot be transformed any further, so as to obtain a real-domain closed form expression for s_i (i=1,..,M) that defines $S_{optimal}$. Still, by analysing (3.18) it is possible to obtain an insight into a few significant properties of $S_{optimal}$, as stated in Theorem 3.3.

Theorem 3.3 Under $S_{optimal}$ broadcast scheduling scheme, the broadcast frequency of item(i) (1/s(i)) is directly proportional to its popularity (p(i)) and inversely proportional to its length (l(i)), i=1,..,M.

Proof of Theorem 3.3

Let us employ the following notation

$$F(s(i)) = s(i)\lambda^*(1-a)e^{-\lambda^*(1-a)s(i)} + e^{-\lambda^*(1-a)s(i)} - 1, \quad i = 1,..,M \tag{3.19}$$

Subsequently,

$$\frac{\partial F(s(i))}{\partial s(i)} = -s(i)\left(\lambda^*(1-a)\right)^2 e^{-\lambda^*(1-a)s(i)}, \quad i = 1,..,M \tag{3.20}$$

According to (3.20), F(s(i)) is a strictly monotonically decreasing function in the domain of our interest, i.e. $\partial F(s(i))/\partial s(i)<0$ for s(i)>0, i=1,..,M.

Now, based on (3.19), (3.18) can be rewritten as

$$\frac{p(i)}{l(i)} F(s(i)) = K\lambda^*(1-a) = \text{const}, \quad i = 1,..,M \tag{3.21}$$

(3.19), (3.20) and (3.21) together imply the following: if two items, e.g. item(i) and item(j) ($i,j \in \{1,..,M\}$, $i \neq j$) are of the same length (l(i)=l(j)=l) but different popularity ($p(i)\neq p(j)$), then under the optimal broadcast scheduling scheme the more popular item gets to be broadcasted more often. More precisely,

$$\text{if } p(i) \geq p(j), \text{ then } \frac{p(i)}{1}F(s(i)) = \frac{p(j)}{1}F(s(j)) = \text{const} \Leftrightarrow |F(s(i))| \leq |F(s(j))|, \text{ i.e. } s(i) \leq s(j) \quad (3.22)$$

(For better understanding of (3.22), please recall that F(s(i)) is monotonically decreasing, but $F(s(i))<0$ for $s(i)\in[0,\infty)$, i=1,..,M. Also, note that the broadcasting frequency of item(i) is obtained simply as $1/s(i)$.)

Similarly, if two items are of the same popularity (p(i)=p(j)=p), but different length ($l(i)\neq l(j)$), then under the optimal broadcast scheduling scheme the shorter item gets to be broadcasted more often (see (3.23)).

$$\text{if } l(i) \geq l(j), \text{ then } \frac{p}{l(i)}F(s(i)) = \frac{p}{l(j)}F(s(j)) = \text{const} \Leftrightarrow |F(s(i))| \geq |F(s(j))|, \text{ i.e. } s(i) \geq s(j) \quad (3.23)$$

End of Proof

3.2 Optimal Broadcast Schedule – Approximate Closed-form Solution

3.2.1 Motivation

In the preceding section, we have presented the procedure for obtaining the implicit-form expression concerning the optimal broadcast scheduling scheme ($S_{optimal}$) in a generalized single-cell WDB system (Theorem 3.1). Also, we have proven the uniqueness of $S_{optimal}$ (Theorem 3.2) and identified its most significant properties (Theorem 3.3).

However, the practical realization of $S_{optimal}$ requires that the exact numerical values of s(i), i=1,..,M are known. Clearly, due to the implicit-form nature of the obtained solution, the numerical values of $S_{optimal}=(s(1)^*,s(2)^*,..,s(M)^*)$ cannot be found directly from (3.18). Therefore, the utilization of $S_{optimal}$ requires that either of the following estimation techniques be employed:

(a) estimate the numerical values of $S_{optimal}=(s(1)^*,s(2)^*,..,s(M)^*)$ algorithmically *on-line*, as suggested in [*Jiang_1*];

(b) estimate the numerical values of $S_{optimal}=(s(1)^*,s(2)^*,..,s(M)^*)$ *off-line*, by employing some well-known non-linear programming algorithm (see [*Belegundu*]);

(c) based on (3.18), first obtain an approximate closed-form expression for $s(i)^*$, i=1,..,M; then, by using the given expressions, obtain their numerical values.

In general, it can be expected that approaches a) and b) result in a less significant estimation error compared to c), as they attempt to obtain the numerical value of $s(i)^*$, i=1,..,M, directly from (3.18). However, from the computational and analytical point of view, (a) and (b) exhibit two major drawbacks over (c).

1) Both (a) and (b) are computationally rather expensive. Namely, in (a), whenever the server needs to determine which item is to be transmitted next, it must re-calculate a 'weight' for every item (based on a function derived from (3.18)) and return the one with the maximum weight value [*Jiang_1*]. In (b), computations occur less frequently. In particular, re-estimation of $s(i)^*$, i=1,..,M should take place only when the popularity (p(i)) of either item changes. However, each re-estimation may consist of up to several thousand iteration steps. The exact number of iterations steps in (b) depends on the actual number of items (i.e. the dimensionality of the solution space), the requested estimation accuracy, and the value of the initial estimation. Approach (c) is computationally the least expensive, as it requires that the re-estimations of $s(i)^*$, i=1,..,M occur as infrequently as in (b), but each re-estimation consists of just several iteration step. (Namely, every $s(i)^*$ (i=1,..,M) is obtained directly from the corresponding closed-form expression.)

2) (a) and (b) produce numerical results only. Thus, they are unable to provide a better insight into the nature of $S_{optimal}$ in a single-cell WDB environment, nor they can facilitate the theoretical analysis of progressively more complex (e.g. multi-cell) WDB based systems. With (c), this drawback is avoided.

In the following section, we attempt to propose an approximate closed-form solution to (3.2), i.e. (3.3), and thereby establish the theoretical ground for approach (c).

3.2.2 Theoretical Discussion

Let us employ the following substitution

$$x(i) = s(i)\lambda^*(1-a), \qquad i = 1,..,M \tag{3.24}$$

where variable s(i) is being replaced with a new variable $x(i)=s(i)\lambda^*(1\text{-}a)$, i=1,..,M, and $\lambda^*(1\text{-}a)$ is assumed constant. Based on (3.24), the cost function (3.3.a) can be rewritten as follows.

$$BR(x(1), x(2),.., x(M)) = \sum_{i=1}^{M} \frac{p(i)}{x(i)}(1 - e^{-x(i)}) \qquad (3.25)$$

The gradient function of (3.25) is given in (3.26),

$$\frac{\partial BR(x(1),x(2),..,x(M))}{\partial x(i)} = p(i)\frac{x(i)e^{-x(i)} + e^{-x(i)} - 1}{x(i)^2} = p(i)B(x(i)), \qquad i = 1,..,M \qquad (3.26)$$

where

$$B(x(i)) = \frac{x(i)e^{-x(i)} + e^{-x(i)} - 1}{x(i)^2}, \qquad i = 1,..,M \qquad (3.27)$$

By employing (3.24), the implicit form solution to (3.3) (i.e. (3.25)) can also be rewritten as presented in (3.28).

$$\frac{p(i)}{l(i)}F(x(i)) = K\lambda^{*}(1\text{-}a) = \text{const}, \qquad i = 1,..,M \qquad (3.28)$$

Here, F(x(i)) corresponds to (3.19), hence

$$F(x(i)) = x(i)e^{-x(i)} + e^{-x(i)} - 1, \qquad x(i) = s(i)\lambda^{*}(1 - a), \qquad i = 1,..,M \qquad (3.29)$$

(The graphical representation of (3.29) is provided in Figure 3.1.) Based on Figure 3.1, it appears justifiable to replace F(x(i)) in (3.28) with its linearized two-segment approximation H(x(i)), as given in (3.30). For the actual interpretation of H(x(i)) that we employ throughout this work, see Definitions 3.2 and 3.3.

$$H(x(i)) = \begin{cases} k \cdot x(i), & x(i) < \chi \quad (k < 0) \\ -1, & x(i) \geq \chi \end{cases} \qquad (3.30)$$

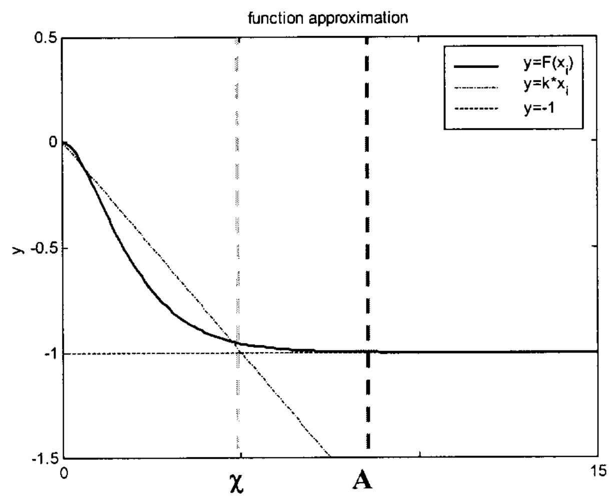


Figure 3.1 F(x(i)) and its linearized approximation H(x(i)) (k=-0.1948, $\chi \approx 5.1355$)

Definition 3.2 We define the first-segment of H(x(i)) (see (3.30)) to be the minimum squared error (MSE) linear approximation of F(x(i)) on the interval [0,A), given [A,+∞] corresponds to the set of points where F(x(i)) can be sufficiently approximated with −1 (see Figure 3.1). [6]

Definition 3.3 For any given $x(i) \in R^+$ we say that F(x(i)) can be sufficiently approximated with −1 provided the difference between F(x(i)) and -1 falls below 1%.[7]

Lemma 3.1 Based on Definitions 3.2 and 3.3, the actual value of parameter k in (3.30) turns out to be **k≈-0.1948**, while the actual value of knee point (χ) in (3.30) turns out to be **$\chi \approx 5.1335$**.

[6] Note, -1 in fact represents the second-segment of H(x(i)).

[7] Note, Definition 3.7 represents an implicit definition of segment [A,+∞].

The proof Lemma 3.1 can be found in Appendix 6.

Now, let us assume that for an arbitrary number of items ($M_1 \le M$) their x(i)-s are such that the first segment approximation applies, i.e. $x(i) \in [0,\chi)$. If we annotate the set of indexes of these M_1 items with $\mathcal{M}_1$, then the following holds

$$F(x(i)) \approx kx(i), \quad \forall i \in \mathcal{M}_1 \tag{3.31}$$

Accordingly, for the remaining $M_2=(M-M_1)$ items the corresponding x(i)-s fall into interval $[\chi,\infty)$, i.e. the second segment approximation of $F(x_i)$ applies. Thus, similarly to the previous case, if we annotate the set of indexes of these M_2 items with $\mathcal{M}_2$, then

$$F(x(i)) \approx -1, \quad \forall i \in \mathcal{M}_2 \tag{3.32}$$

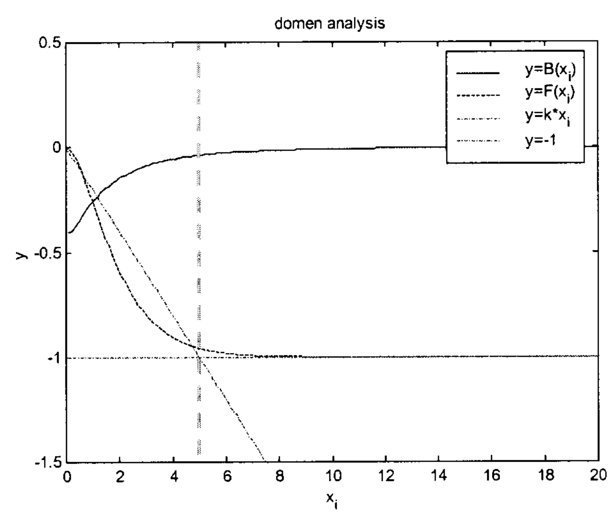


Figure 3.2 F(x(i)), H(x(i)) and B(x(i))

However, based on Figure 3.2, for every item(i) that requires the second segment approximation (3.32), i.e. $x(i) \in [\chi,\infty)$, the corresponding B(x(i)) (see (3.26) and (3.27)) is close to zero regardless of the exact value for x(i). More precisely,

$$B(x(i)) \approx 0, \quad \forall x(i) \in [\chi, \infty) \tag{3.33}$$

Thus, from (3.26) and (3.31), the following implies

$$\frac{\partial BR(x(1),..,x(i-1),x(i),x(i+1),..,x(M))}{\partial x(i)} \approx 0, \quad \forall i \in \mathcal{M}_2 \tag{3.34}$$

The above expression (together with Figure 3.2) suggests that the value of the cost function (BR) can be assumed constant for all x(i), $i \in \mathcal{M}_2$, i.e. BR is constant on the interval $x(i) \in [\chi, \infty)$, $i \in \mathcal{M}_2$. Hence, from (3.34), approximation (3.35) is obtained.

$$BR(x(1),..,x(i-1),x(i),x(i+1),..,x(M)) \approx BR(x(1),..,x(i-1),x(i)=\infty,x(i+1),..,x(M)), \quad \forall i \in \mathcal{M}_2 \tag{3.35}$$

Subsequently, based on (3.36)

$$\lim_{x(i)\to\infty} \frac{p(i)}{x(i)}(1-e^{-x(i)}) = 0 \tag{3.36}$$

(3.35) can be extended into (3.37).

$$BR(x(1),..,x(i),..,x(M)) \approx \sum_{\substack{j=1 \\ j\neq i}}^{M} \frac{p(j)}{x(j)}(1-e^{-x(j)}) + \lim_{x(i)\to\infty} \frac{p(i)}{x(i)}(1-e^{-x(i)}) = \sum_{\substack{j=1 \\ j\neq i}}^{M} \frac{p(j)}{x(j)}(1-e^{-x(j)}), \quad \forall i \in \mathcal{M}_2 \tag{3.37}$$

Finally, (3.37) implies the following

$$BR(x(1),..,x(i-1),x(i),x(i+1),..,x(M)) \approx BR(x(1),..,x(i-1),x(i+1),..,x(M)), \quad \forall i \in \mathcal{M}_2 \tag{3.38}$$

In practical terms, (3.38) suggests that if x(i) associated with item(i) is such that the corresponding F(x(i)) (see (3.29)) falls within the second segment approximation (3.32), i.e. $i \in \mathcal{M}_2$, then the broadcasting of the given item appears insignificant from the cost function point of view (see (3.36) and (3.37)). Put another way, (3.38) proves that if under the optimal broadcast scheduling scheme ($S_{optimal}$) item(i) gets to be broadcasted every s(i) time units, where s(i) corresponds to $x(i)=s(i)\lambda^*(1\text{-}a)$ and $x(i)\in[\chi,\infty)$, then the actual broadcasting of this item provides negligible contribution to the overall BR, thus it can be entirely omitted.

The conclusion made in the above paragraph is of major importance, as it allows for a substantial simplification in the definition of our optimization problem (3.2), i.e. (3.3), and the corresponding solution (3.18), i.e. (3.28). Namely, based on (3.38), (3.3) gets transformed to

$$\text{maximize} \quad BR(s(1), s(2),.., s(M)) \approx BR(s(i) \mid i \in \mathcal{M}_1) = \frac{1}{\lambda^*(1\text{-}a)}\sum_{i\in\mathcal{M}_1}\frac{p(i)}{s(i)}(1-e^{-\lambda^*(1\text{-}a)s(i)}) \qquad (3.39.a)$$

$$\text{subject to} \quad C(s(1),s(2),..,s(M)) = \sum_{i\in\mathcal{M}_1}\frac{l(i)}{s(i)} - 1 = 0 \qquad (3.39.b)$$

Accordingly, by employing (3.30) instead of (3.29) in (3.28), the implicit-form solution to (3.39) becomes

$$\frac{p(i)}{l(i)}F(x(i)) \approx \frac{p(i)}{l(i)}\cdot kx(i) = \frac{p(i)}{l(i)}ks(i)\lambda^*(1-a) = K\lambda^*(1\text{-}a) = \text{const}, \quad i \in \mathcal{M}_1 \qquad (3.40)$$

Now, (3.40) can be easily transformed into the explicit-form solution, as given in (3.41).

$$s(i) = \frac{K}{k}\cdot\frac{l(i)}{p(i)}, \quad i \in \mathcal{M}_1 \qquad (3.41)$$

The exact value of parameter K (i.e. the Lagrange multiplier) in (3.41) is unknown. However, by employing (3.41) back in (3.39.b) (as presented in (3.42)), the closed-form expression for K (23.43) is obtained.

$$C(s(1),s(2),..,s(M)) = \sum_{i\in\mathcal{M}_1}\frac{k}{K}p(i) - 1 = 0 \qquad (3.42)$$

$$K = k\sum_{i\in\mathcal{M}_1} p(i) \tag{3.43}$$

Thus, by replacing K in (3.41) with the right-hand side expression of (3.43), (3.41) becomes

$$s(i) = \frac{l(i)}{p(i)}\sum_{i\in\mathcal{M}_1} p(i) = \frac{l(i)}{\dfrac{p(i)}{\sum_{i\in\mathcal{M}_1} p(i)}}, \qquad i \in \mathcal{M}_1 \tag{3.44}$$

Please note that the denominator of (3.44) represents a re-normalized probability of item(i) being requested ($i\in\mathcal{M}_1$), where the re-normalization spans over a reduced set of items that are shown to be worth of broadcasting with respect to BR, i.e. over $\mathcal{M}_1$. By annotating these re-normalized probabilities with $\overline{p}(i)$ ($i\in\mathcal{M}_1$), (3.44) can be rewritten as presented in (3.45).

$$s(i) = \frac{l(i)}{\overline{p}(i)}, \qquad i \in \mathcal{M}_1 \tag{3.45}$$

Thus, (3.45) represents our final approximate closed-form solution to the problem concerning the optimal broadcast scheduling scheme in a generalized WDB system, as defined in (3.2).

3.2.3 Algorithm for Estimation of $S_{optimal}$ Based on Approximate Closed-form Solution (FBE Algorithm)

Let us annotate the set of inter-broadcast intervals obtained by employing (3.44), i.e. (3.45), with $\tilde{S}_{optimal}$. Based on the discussion presented in Section 3.2.2, broadcasting according to $\tilde{S}_{optimal}$ is expected to result in BR nearly as large as it would be obtained if broadcasting were done according to the actual $S_{optimal}$. In particular, the following is expected to hold

$$BR(S_{optimal}) \approx BR(\tilde{S}_{optimal}) \tag{3.46}$$

Undoubtedly, the major advantage of utilizing $\tilde{S}_{optimal}$, over some other possible approximate solutions to (3.2), lies in the computational simplicity of (3.45). (For more details regarding other possible solutions, see Section 3.2.1.) Nevertheless, it should be noted that the actual computation of $\tilde{S}_{optimal}$ goes slightly beyond a simple division of l(i) and $\overline{p}(i)$, $i \in \mathcal{M}_1$, (see (3.45)). In order to obtain $\tilde{S}_{optimal}$ from (3.45), the exact knowledge of $\mathcal{M}_1$ is required. However, according to (3.31), $\mathcal{M}_1$ cannot be precisely identified unless the actual values of s(i), i=1,..,M, of $S_{optimal}$, i.e. $\tilde{S}_{optimal}$, are known. Thus, a technique aimed at calculating $\tilde{S}_{optimal}$, based on (3.45), should also be capable of identifying $\mathcal{M}_1$.

In Figure 3.3 we propose an iterative algorithm for fast estimation of $\tilde{S}_{optimal}$ (FBE Algorithm). The algorithm begins by assuming that all items correspond to the 'first-segment' approximation (3.31), that is $\mathcal{M}_1=\{1,2,..,M\}$. However, if some of s(i) (obtained by employing (3.45)) turn out to be greater than $\chi/\lambda^*(1\text{-}a)$ (see Figure 3.2), this suggests that the initial assumption was incorrect, and the corresponding items should be excluded from the broadcast schedule (their indexes should be excluded from $\mathcal{M}_1$). The exclusion of items should start with the one that corresponds to the largest s(i) that is greater than $\chi/\lambda^*(1\text{-}a)$, and it should be done on one 'one by one' basis. As soon as any given item gets excluded from the broadcast schedule, a change in $\sum_{i\in\mathcal{M}_1} p(i)$ occurs. (In particular, the value of the given summation decreases.) Accordingly, based on (3.45), the following is obtained

$$s(i)[k+1] = \frac{l(i)}{p(i)} \cdot \sum_{i\in\mathcal{M}_1[k+1]} p(i) \;\; \le \;\; s(i)[k] = \frac{l(i)}{p(i)} \cdot \sum_{i\in\mathcal{M}_1[k]} p(i) \tag{3.47}$$

Step 1 Relative popularity (p(i)) and length (l(i)) of every item is known.

Assume that all items require 'first-segment' approximation (3.31), i.e. $\mathcal{M}_1=\{1,2,..,M\}$.

Step 2 For $\forall\ i\in\mathcal{M}_1$ calculate[8]

$$\bar{p}(i)=\frac{p(i)}{\sum_{i\in\mathcal{M}_1}p(i)} \quad \text{and} \quad s(i)=\frac{l(i)}{\bar{p}(i)}$$

Step 3 $\mathcal{E}=\varnothing$.

Find the largest s(i), $i\in\mathcal{M}_1$. (Let us annotate the corresponding index with k).

Check whether s(k) satisfies the following inequality.

$$s(k)\le\frac{\chi}{\lambda^*(1-a)}\approx\frac{5}{\lambda^*(1-a)}$$

If the inequality is not satisfied, then $\mathcal{E}=\{k\}$.[9]

If more than one s(i) is found in Step 3, proceed with the one that correspond to the smallest p(i).

Step 4 If $\mathcal{E}\neq\varnothing$ then $\mathcal{M}_1=\mathcal{M}_1/\mathcal{E}$.[10] Go to Step 2.

Otherwise proceed with Step 5.

Step 5 $\tilde{S}_{optimal}=\{s(i)\,|\,i\in\mathcal{M}_1\}$. End.

Figure 3.3 Algorithm for estimation of $S_{optimal}$

(*F*ast *B*roadcast Scheduling Scheme *E*stimation *A*lgorithm – *FBE Algorithm*)

---

[8] Note, in the first round of computation $\mathcal{M}_1$ contains all the indexes, thus $\sum_{i\in\mathcal{M}_1}p(i)=1$, and accordingly $\bar{p}(i)=p(i)$.

[9] If the given inequality is not satisfied, this suggests $x(k)=s(k)\lambda^*(1\text{-}a)>\chi$, thus the assumption that the 'first-segment' approximation (3.31) applies, is incorrect.

[10] In this step the largest s_i that violates the inequality from Step 3 gets excluded from $\tilde{S}_{optimal}$.

(In expression (3.47) s(i)[k] and $\mathcal{M}_l[k]$ are used to annotate the values of s(i) and $\mathcal{M}_l$ at the end of the kth iteration, i.e. after a total number of k items have been excluded from the broadcast channel.) Thus, as it can be observed from (3.47), if item(i) ($i\in\mathcal{M}_l$) was initially associated with $s(i)[0]>\chi/\lambda^*(1-a)$ (when all items were included in the broadcast schedule, i.e. $\mathcal{M}_l=\{1,2,..,M\}$), then in the very next iteration it may end up with $s(i)[1]\leq\chi/\lambda^*(1-a)< s(i)[0]$ thereby satisfying the inequality constraint of Step 3 in Figure 3.3. Hence, as stated earlier, it is crucial that the exclusion of items be performed on 'one-by-one basis', starting from the one that corresponds to the largest estimated s(i) greater than $\chi/\lambda^*(1-a)$.

3.2.4 Experimental Performance Evaluation of FBE Algorithm

In this section, through a series of experimental results, we compare the actual values and performance of $S_{optimal}$ (obtained by solving (3.2) with Matlab - see Appendix 7 for a detailed explanation) and $\tilde{S}_{optimal}$ (obtained by employing our FBE algorithm as proposed in Figure 3.3). In Experiment 1.a) through 1.c) the same non-uniform distribution of p(i) and uniform distribution of l(i) (i=1,..,10) are used (see Table 3.1 through 3.3), while the value of $\lambda^*(1\text{-}a)$[11] is varied so as to simulated different system conditions. In Experiment 2.a) through 2.c) the value of $\lambda^*(1\text{-}a)$ is varied as well, while in contrast to the preceding set of experiments, both p(i) and l(i) (i=1,..,10) are chosen to be of non-uniform distribution (see Table 3.4 through 3.6). In Experiment 3.a) through 3.b), p(i) and l(i) were chosen to be of non-uniform distribution again, but the overall collection size was increased to 20 (i.e. i=1,..,20).

Experiment 1.a) $H=\lambda^*(1\text{-}a)=0.05 \Rightarrow \chi/\lambda^*(1\text{-}a)\approx 5/0.05=100$

In this experiment, the value of parameter $H=\lambda^*(1\text{-}a)$ is set to 0.05. For the given value of H, as it can be seen from Table 3.1, all s(i)-s obtained already in the initial evaluation of $\tilde{S}_{optimal}$, i.e. $\tilde{S}_{optimal}[0]$, satisfy the constraint $s(i)\leq\chi/\lambda^*(1\text{-}a)$. Thus, in this case, according to our algorithm proposed in Figure 3.3, $\tilde{S}_{optimal}[0]$ is assumed an adequate representation of the true $S_{optimal}$. Table 3.1 and Figure 3.4 verify that $\tilde{S}_{optimal}[0]$ is very close to the actual $S_{optimal}$, as estimated using Matlab (see Appendix 7). Moreover, under the assumed H, p(i) and l(i) (i=1,..,10), the obtained values of broadcast service ratio (BR) at $\tilde{S}_{optimal}[0]$ and $S_{optimal}$, i.e. $BR(\tilde{S}_{optimal}[0])$ and $BR(S_{optimal})$, indicate that these two broadcast scheduling schemes can be expected to result in very close WDB system performance.

[11] In further discussion we annotate $\lambda^*(1\text{-}a)$ with H.

item	1	2	3	4	5	6	7	8	9	10
p	0.3	0.2	0.1	0.1	0.1	0.05	0.05	0.05	0.04	0.01
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	4.8383	6.0456	8.9616	8.9616	8.9619	13.6527	13.6527	13.6527	15.7758	99.7431
$\tilde{S}_{optimal}[0]$	3.3333	5.0000	10.0000	10.0000	10.0000	20.0000	20.0000	20.0000	25.000	100.00

Table 3.1 Experiment 1.a) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.8192$

(number of evaluations (iterations) = 732)

$BR(\tilde{S}_{optimal}[0]) = 0.8090$

(number of iterations = 0)

error: $\dfrac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 1.25\ [\%]$

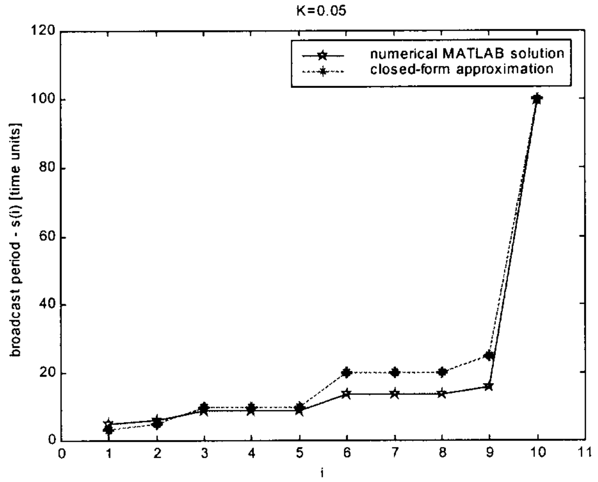


Figure 3.4 Experiment 1.a) - Numerical MATLAB solution and closed-form approximation

Experiment 1.b) $H=\lambda^*(1\text{-}a)=0.1 \Rightarrow \chi/\lambda^*(1\text{-}a)\approx 5/0.1=50$

In Experiment 1.b), the value of parameter H is increased to 0.1. Now, as a consequence of the given change in system conditions, it turns out that s(10), obtained in the initial estimation of $\tilde{S}_{optimal}$, violates the constraint $s(i)\leq\chi/\lambda^*(1\text{-}a)$. Hence, according to the FBE algorithm, the corresponding item needs to be excluded from the broadcast schedule. After the exclusion of item(10) and re-estimation of $\tilde{S}_{optimal}$, all s(i) of the new $\tilde{S}_{optimal}[1]$ (see Table 3.2) satisfy the above constraint. Thus, in this case, $\tilde{S}_{optimal}[1]$ is an adequate approximation of the true $S_{optimal}$. Again, Table 3.2 and Figure 3.5 verify that $\tilde{S}_{optimal}[1]$ is very close to $S_{optimal}$, as estimated using Matlab, while the obtained values of $BR(\tilde{S}_{optimal}[1])$ and $BR(S_{optimal})$ indicate that these two broadcast scheduling schemes can be expected to result in almost equally good WDB system performance, under the assumed H, p(i) and l(i) (i=1,..,10).

item	1	2	3	4	5	6	7	8	9	10
p	0.3	0.2	0.1	0.1	0.1	0.05	0.05	0.05	0.04	0.01
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	4.4250	5.6337	8.7932	8.7848	8.7940	14.8883	14.9617	14.9304	18.4321	4.8543e+5
$\tilde{S}_{optimal}[1]$	3.3000	4.9500	9.9000	9.9000	9.9000	19.8000	19.8000	19.8000	24.7500	Inf

Table 3.2 Experiment 1.b) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.6859$

(number of (evaluations) iterations = 2255)

$BR(\tilde{S}_{optimal}[1]) = 0.6838$

(number of iterations = 1)

error: $\dfrac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.31\,[\%]$

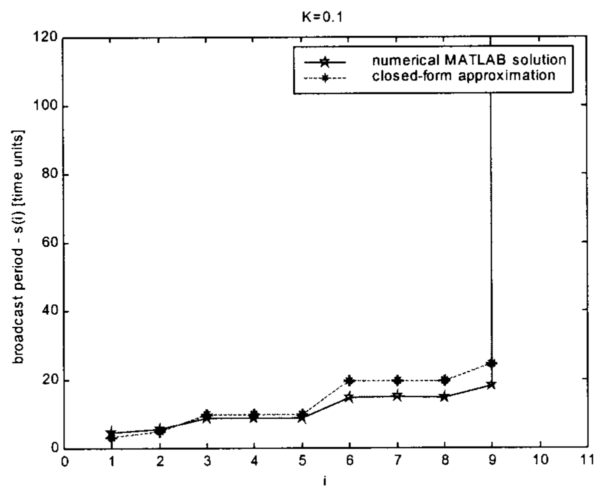


Figure 3.5 Experiment 1.b) - Numerical MATLAB solution and closed-form approximation

Experiment 1.c) $H=\lambda^*(1-a)=0.3 \;\Rightarrow\; \chi/\lambda^*(1-a)\approx 5/0.3=16.6667$

In Experiment 1.c), H is increased to 0.3. Consequently, it turns out that more than one s(i) obtained in the initial estimation of $\tilde{S}_{optimal}$ violates the constraint $s(i)\leq\chi/\lambda^*(1-a)$. Through a gradual removal of items from the broadcast schedule, according to the algorithm from Figure 3.3, finally all s(i) of $\tilde{S}_{optimal}[5]$ (see Table 3.3) are shown to satisfy the above constraint. Thus, in this case, $\tilde{S}_{optimal}[5]$ represents an adequate approximation of the true $S_{optimal}$.

Again, Table 3.3 and Figure 3.6 verify the similarity between $\tilde{S}_{optimal}[5]$ and $S_{optimal}$, as estimated using Matlab. At the same time, the values of $BR(\tilde{S}_{optimal}[5])$ and $BR(S_{optimal})$ indicate that these two broadcast scheduling schemes result in almost identical WDB system performance, under the assumed H, p(i) and l(i) (i=1,..,10).

item	1	2	3	4	5	6	7	8	9	10
p	0.3	0.2	0.1	0.1	0.1	0.05	0.05	0.05	0.04	0.01
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	2.9174	3.9110	7.4716	7.4714	7.4717	7.9927e+5	7.9927e+5	7.9927e+5	6.7253e+5	7.5821e+4
$\tilde{S}_{optimal}[5]$	2.6667	4.0000	8.0000	8.0000	8.0000	Inf	Inf	Inf	Inf	Inf

Table 3.3 Experiment 1.c) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.4373$

(number of (evaluations) iterations = 1032)

$BR(\tilde{S}_{optimal}[5]) = 0.4366$

(number of iterations = 5)

error: $$\frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.16\,[\%]$$

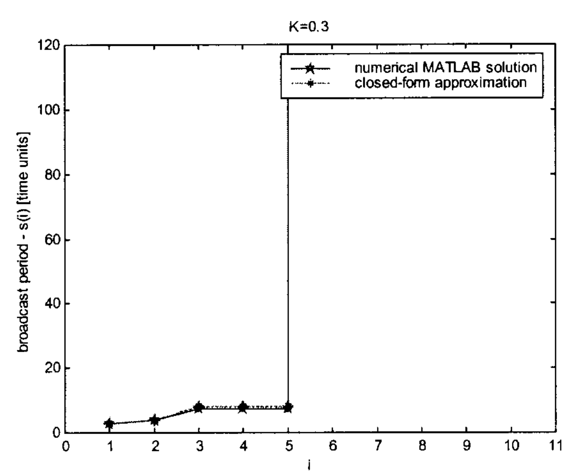


Figure 3.6 Experiment 1.c) - Numerical MATLAB solution and closed-form approximation

Experiment 2.a) $H=\lambda^*(1-a)=0.05 \Rightarrow \chi/\lambda^*(1-a)\approx 5/0.05=100$

In Experiment 2.a) to 2.c) the distribution of item lengths (l(i)) is changed into a non-uniform distribution (see Table 3.4 to 3.6).

Similar to Experiment 1.a), in Experiment 2.a) the value of parameter H is set to 0.05. Again, for the given H, all s(i) obtained already in the initial evaluation of $\tilde{S}_{optimal}$ satisfy the constraint $s(i)\leq\chi/\lambda^*(1-a)$ (see Table 3.4). Thus, $\tilde{S}_{optimal}[0]$ is assumed an adequate representation of the true $S_{optimal}$. Table 3.4 and Figure 3.7 show that $\tilde{S}_{optimal}[0]$ and $S_{optimal}$, as estimated using Matlab, are almost identical. Consequently, the obtained values of $BR(\tilde{S}_{optimal}[0])$ and $BR(S_{optimal})$ verify that these two broadcast scheduling schemes result in equally good WDB system performance.

item	1	2	3	4	5	6	7	8	9	10
p	0.3	0.2	0.1	0.1	0.1	0.05	0.05	0.05	0.04	0.01
l [time]	10	8	6	4	2	1	1	1	1	1
$S_{optimal}$	33.3332	40.00	60.00	40.00	20.0001	20.0001	20.0001	20.0001	25.00	100.00
$\tilde{S}_{optimal}[0]$	33.3333	40.00	60.00	40.00	20.0000	20.0000	20.0000	20.0000	25.00	100.00

Table 3.4 Experiment 2.a) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.4902$

(number of (evaluations) iterations = 12)

$BR(\tilde{S}_{optimal}[0]) = 0.4902$

(number of iterations = 0)

error: $$\frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.0\ [\%]$$

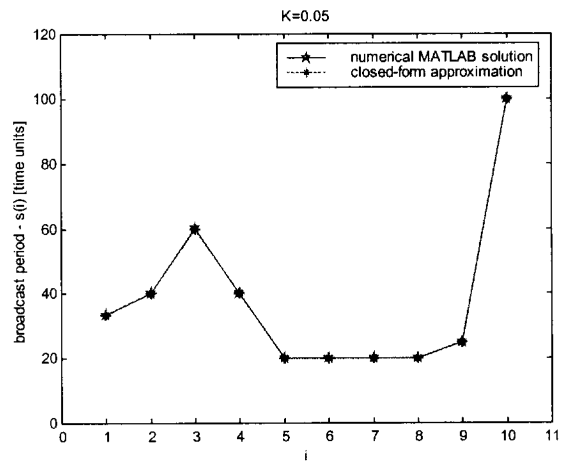


Figure 3.7 Experiment 2.a) - Numerical MATLAB solution and closed-form approximation

Experiment 2.b) $H=\lambda^*(1-a)=0.1 \Rightarrow \chi/\lambda^*(1-a)\approx 5/0.1=50$

In Experiment 2.b), H is increased to 0.1. In this case, upon the initial evaluation of $\tilde{S}_{optimal}$, it turns out that s(3) and s(10) are greater than $\chi/\lambda^*(1-a)$. However, after the removal of these two items from the broadcast schedule, all s(i) of the newly obtained $\tilde{S}_{optimal}[2]$ are shown to satisfy the constraint $s(i)\leq\chi/\lambda^*(1-a)$ (see Table 3.5). Table 3.5 and Figure 3.8 verify the similarity between $\tilde{S}_{optimal}[2]$ and $S_{optimal}$, while the obtained values of $BR(\tilde{S}_{optimal}[2])$ and $BR(S_{optimal})$ prove that these two broadcast scheduling schemes result in very close WDB system performance.

item	1	2	3	4	5	6	7	8	9	10
p	0.3	0.2	0.1	0.1	0.1	0.05	0.05	0.05	0.04	0.01
l [time]	10	8	6	4	2	1	1	1	1	1
$S_{optimal}$	28.3949	43.5969	3.3851e+3	43.5793	15.6191	15.6538	15.7248	15.7051	19.5216	2.9182e+3
$\tilde{S}_{optimal}[2]$	29.6667	35.6000	Inf	35.6000	17.8000	17.8000	17.8000	17.8000	22.2500	Inf

Table 3.5 Experiment 2.b) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.3116$

(number of (evaluations) iterations = 2137)

$BR(\tilde{S}_{optimal}[2]) = 0.3106$

(number of iterations = 2)

$$\textbf{error:} \quad \frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.32\ [\%]$$

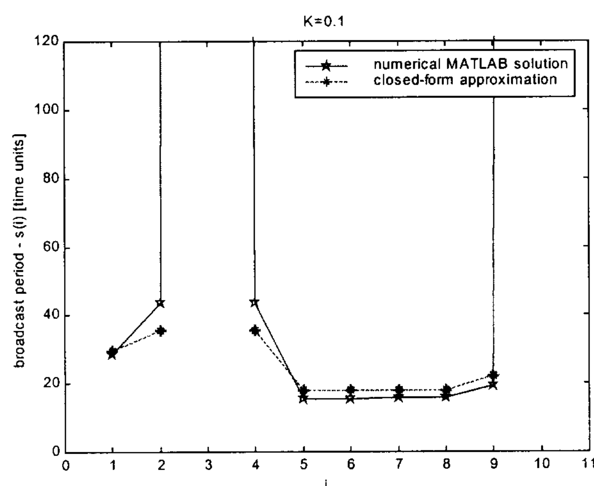


Figure 3.8 Experiment 2.b) - Numerical MATLAB solution and closed-form approximation

Experiment 2.c) $\mathbf{H=\lambda^*(1\text{-}a)=0.15 \;\Rightarrow\; \chi/\lambda^*(1\text{-}a)\approx 5/0.15=33.3333}$

In Experiment 2.c) H is set to 0.15. For the given value of H, more than one s(i) obtained in the initial evaluation of $\tilde{S}_{optimal}$ is greater than $\chi/\lambda^*(1\text{-}a)$. However, through a gradual removal of the given items from the broadcast schedule (see FAB algorithm proposed in Figure 3.3), finally all s(i) of $\tilde{S}_{optimal}[4]$ (see Table 3.6) are shown to satisfy the constraint $s(i) \leq \chi/\lambda^*(1\text{-}a)$. The presented results, including Table 3.6 and Figure 3.9, verify the similarity between both, $\widetilde{S}_{optimal}[4]$ and $S_{optimal}$ (as estimated using Matlab), and WDB system performances they provide.

item	1	2	3	4	5	6	7	8	9	10
p	0.3	0.2	0.1	0.1	0.1	0.05	0.05	0.05	0.04	0.01
l [time]	10	8	6	4	2	1	1	1	1	1
$S_{optimal}$	21.4991	1.5580e+3	1.0066e+3	947.3735	11.1719	11.1709	11.1709	11.1709	14.0898	1.0017e+3
$\widetilde{S}_{optimal}[4]$	19.6667	Inf	Inf	Inf	11.8000	11.8000	11.8000	11.8000	14.7500	Inf

Table 3.6 Experiment 2.c) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.2295$

(number of (evaluations) iterations = 876)

$BR(\widetilde{S}_{optimal}[4]) = 0.2297$

(number of iterations = 4)

error: $$\frac{BR(S_{optimal}) - BR(\widetilde{S}_{optimal}[0])}{BR(S_{optimal})} \approx 0\ [\%]$$

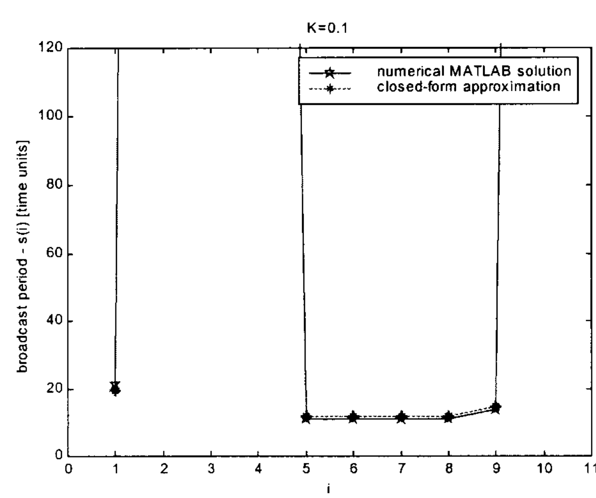


Figure 3.9 Experiment 2.c) - Numerical MATLAB solution and closed-form approximation

Experiment 3.a) $H=\lambda^*(1\text{-}a)=0.02 \Rightarrow \chi/\lambda^*(1\text{-}a)\approx 5/0.02=250$

In Experiment 3.a) through 3.c), the data-item collection size is increased to 20.

In Experiment 3.a), in particular, H is set to 0.02. Upon initial evaluation of $\tilde{S}_{optimal}$, according to the FBE algorithm, it turns out that more than one value of s(i) is greater than $\chi/\lambda^*(1\text{-}a)$. Through a gradual removal of such items from the broadcast schedule, finally all s(i) of $\tilde{S}_{optimal}$[3] are shown to satisfy the constraint $s(i)\leq\chi/\lambda^*(1\text{-}a)$. The results presented in Table 3.10 and Figure 3.10 prove that the two broadcast scheduling schemes, $S_{optimal}$ and $\tilde{S}_{optimal}$[3], are rather similar and they provide nearly identical WDB system performance.

item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
p	0.18	0.14	0.1	0.09	0.08	0.07	0.06	0.05	0.04	0.03
l [time]	5	4	3	2	1	5	4	3	2	1
$S_{optimal}$	29.2924	30.2016	30.8932	25.9418	18.1694	55.3114	52.6310	49.0463	43.9411	35.1276
$\tilde{S}_{optimal}$[7]	26.9444	27.7143	29.1000	21.5556	12.1250	69.2857	64.6667	58.2000	48.5000	32.3333

item	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
p	0.03	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
l [time]	5	4	3	2	1	5	4	3	2	1
$S_{optimal}$	163.0395	199.0485	146.5956	92.6891	43.6376	500.8378	400.7976	300.5356	199.5236	96.2749
$\tilde{S}_{optimal}$[7]	161.6667	194.0000	145.5000	97.0000	48.5000	Inf	Inf	Inf	194.0000	97.0000

Table 3.7 Experiment 3.a) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.6662$

(number of (evaluations) iterations = 2001)

$BR(\tilde{S}_{optimal}[4]) = 0.6644$

(number of iterations = 3)

error: $\dfrac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} \approx 0.27\ [\%]$

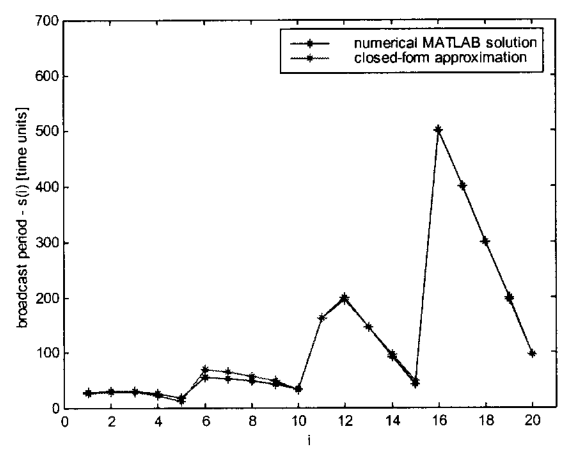


Figure 3.10 Experiment 3.a) - Numerical MATLAB solution and closed-form approximation

Experiment 3.b) $H=\lambda^*(1\text{-}a)=0.05 \Rightarrow \chi/\lambda^*(1\text{-}a)\approx 5/0.05=100$

In Experiment 3.b), H is increased to 0.05. Upon initial evaluation of $\tilde{S}_{optimal}$, according to the FBE algorithm, several s(i) turned out to be greater than $\chi/\lambda^*(1\text{-}a)$. Through a gradual removal such items, finally all s(i) of $\tilde{S}_{optimal}[7]$ are shown to satisfy the constraint $s(i)\leq\chi/\lambda^*(1\text{-}a)$. The obtained results prove that the two broadcast scheduling schemes, $S_{optimal}$ and $\tilde{S}_{optimal}[7]$, although being somewhat different (see Table 3.11 and Figure 3.11), still result in nearly identical WDB system performance.

item	1	2	3	4	5	6	7	8	9	10
p	0.18	0.14	0.1	0.09	0.08	0.07	0.06	0.05	0.04	0.03
l [time]	5	4	3	2	1	5	4	3	2	1
$S_{optimal}$	23.15	24.04	24.06	20.19	15.67	149.29	112.28	67.51	41.78	28.67
$\widetilde{S}_{optimal}[7]$	24.72	25.43	26.69	19.77	11.12	63.58	59.36	53.44	44.45	29.63

item	11	12	13	14	15	16	17	18	19	20
p	0.03	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
l [time]	5	4	3	2	1	5	4	3	2	1
$S_{optimal}$	250.43	254.20	207.68	149.24	38.87	515.25	418.19	322.31	227.72	125.38
$\widetilde{S}_{optimal}[7]$	Inf	Inf	Inf	88.90	44.45	Inf	Inf	Inf	Inf	88.90

Table 3.8 Experiment 3.b) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.4467$

(number of (evaluations) iterations = 2009)

$BR(\widetilde{S}_{optimal}[4]) = 0.4463$

(number of iterations = 7)

$$\textbf{error:}\quad \frac{BR(S_{optimal}) - BR(\widetilde{S}_{optimal}[0])}{BR(S_{optimal})} \approx 0.09\,[\%]$$

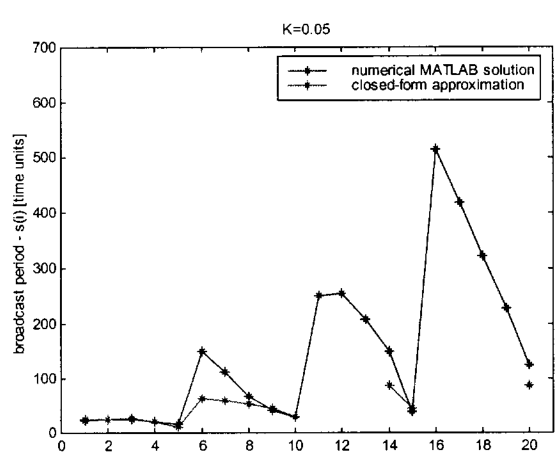


Figure 3.11 Experiment 3.b) - Numerical MATLAB solution and closed-form approximation

Experiment 3.c) $H=\lambda^*(1\text{-}a)=0.1 \Rightarrow \chi/\lambda^*(1\text{-}a)\approx 5/0.1=50$

In Experiment 3.c), H is increased to 0.1. Eventually, all s(i) obtained in the ninth round of procedure based on the FBE algorithm ($\tilde{S}_{optimal}[9]$) are shown to satisfy $s(i)\leq\chi/\lambda^*(1\text{-}a)$. (See Table 3.12 and Figure 3.12.) As in the previous cases, the performance of $\tilde{S}_{optimal}[9]$ turns out to be almost identical to the performance of $S_{optimal}$ obtained through numerical estimation.

item	1	2	3	4	5	6	7	8	9	10
p	0.18	0.14	0.1	0.09	0.08	0.07	0.06	0.05	0.04	0.03
l [time]	5	4	3	2	1	5	4	3	2	1
$S_{optimal}$	27.7778	28.5714	30.0000	22.2222	12.4999	71.4287	66.6668	60.0001	50.0001	33.3334
$\tilde{S}_{optimal}[9]$	23.8922	24.5696	25.7883	19.1137	10.7475	61.4631	57.3412	51.6684	42.9422	28.6706

item	11	12	13	14	15	16	17	18	19	20
p	0.03	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
l [time]	5	4	3	2	1	5	4	3	2	1
$S_{optimal}$	166.6667	200.00	150.00	100.00	50.00	500.00	400.00	300.00	200.00	100.00
$\tilde{S}_{optimal}[9]$	Inf	Inf	Inf	Inf	42.9422	Inf	Inf	Inf	Inf	Inf

Table 3.9 Experiment 3.c) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.2765$

(number of (evaluations) iterations = 22)

$BR(\tilde{S}_{optimal}[4]) = 0.2911$

(number of iterations = 9)

error: $$\frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} \approx 0\ [\%]$$

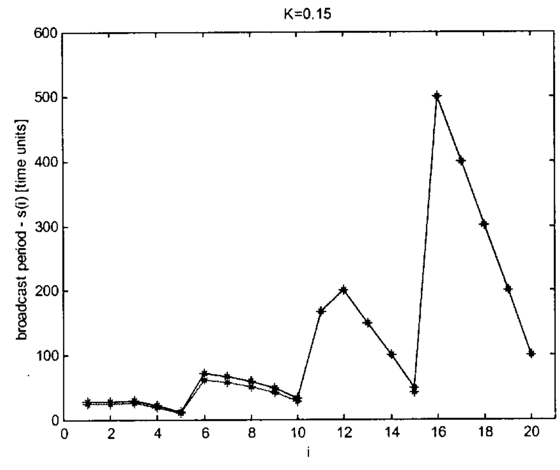


Figure 3.12 Experiment 3.c) - Numerical MATLAB solution and closed-form approximation

3.2.5 Simulation-Based Evaluation of FBE Algorithm

With the experiment presented in this section we study the effectiveness of the FBE algorithm in a real-world setting. The actual results are obtained employing our Java-based simulation software, as already mentioned in Section 2.6. A detailed overview of the software can be found in Chapter 7.

More specifically, in this experiment we simulate the performance of a single-cell WDB system (see Table 3.10), under the following broadcast scheduling policies (see Table 3.11):

- $S_{optimal}$ obtained through numerical estimation[12];
- $\tilde{s}_{optimal}[5]$ - an approximation of $S_{optimal}$ obtained using the FEB algorithm;
- five other broadcast scheduling schemes ($\tilde{s}_{optimal}[0], \tilde{s}_{optimal}[1], \tilde{s}_{optimal}[2], \tilde{s}_{optimal}[3]$, and $\tilde{s}_{optimal}[4]$), each one corresponding to an intermediate step of the FBE algorithm leading towards $\tilde{s}_{optimal}[5]$.

Based on the obtained results (see Table 3.11 and Figures 3.13 to 3.15), a gradual removal of the items that are not worth of broadcasting, according to the FEB algorithm, is clearly reflected in the system's performance. In particular, each successive broadcast scheduling scheme ($\tilde{s}_{optimal}[0]$ to $\tilde{s}_{optimal}[4]$) provides an improvement in the system's broadcast service ratio (BR), broadcast loss ratio (LR) and users' waiting time (WT).[13] Finally, the performance of $\tilde{s}_{optimal}[5]$ is shown to be as effective as the performance of $S_{optimal}$.

---

[12] $S_{optimal}$ is a Matlab-obtained numerical solution to (3.3), under the system and user related parameters as in Table 3.10. The estimation procedure requires 1032 iterations steps, assuming $\tilde{s}_{optimal}[0]$ as the initial estimation point.

[13] There is a slight deviation in the observed trend for BR and LR at $\tilde{s}_{optimal}[1]$, most likely caused by the randomness associated with many of the simulation parameters as well as very close values of BR, i.e. LR, at $\tilde{s}_{optimal}[1]$ and $\tilde{s}_{optimal}[0]$.

MU reattempt probability	**$P_{reattempt}$=0.7**
MU mean reattempt time [time units]	**λ^*=0.5**
Mean MU arrival rate	**λ=10 [users per time unit]**
simulation time	**300 [time units]**
number of unicast channels in the system	**20**
$P_{success}$	**0.4233** (obtained from the observed value of UR)
H	**0.3** (obtained as H=λ^***(1-$P_{reattempt}$(1-$P_{success}$))**)

Table 3.10 System and user related simulation parameters

item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**			
p	0.3	0.2	0.1	0.1	0.1	0.05	0.05	0.05	0.04	0.01			
l [time]	1	1	1	1	1	1	1	1	1	1	**BR**	**LR**	**WT**
$S_{optimal}$	2.90	3.90	7.45	7.45	7.45	7.9e+5	7.9e+5	7.9e+5	6.7e+5	7.5e+4	0.4014	0.0300	1.3496
$\tilde{S}_{optimal}[0]$	3.3	5.0	10.0	10.0	10.0	20.0	20.0	20.0	25.0	100.0	0.3370	0.0603	1.6125
$\tilde{S}_{optimal}[1]$	3.30	4.95	9.90	9.90	9.90	19.80	19.80	19.80	24.75	Inf	0.3318	0.0669	1.6021
$\tilde{S}_{optimal}[2]$	3.16	4.75	9.50	9.50	9.50	19.00	19.00	19.00	Inf	Inf	0.3584	0.0503	1.5294
$\tilde{S}_{optimal}[3]$	3.00	4.50	8.99	8.99	8.99	18.01	18.01	Inf	Inf	Inf	0.3726	0.0483	1.4533
$\tilde{S}_{optimal}[4]$	2.83	4.24	8.49	8.49	8.49	17.01	Inf	Inf	Inf	Inf	0.3855	0.0439	1.3868
$\tilde{S}_{optimal}[5]$	2.6	4.0	8.0	8.0	8.0	Inf	Inf	Inf	Inf	Inf	0.4086	0.0303	1.2947

Table 3.11 System performance under different broadcast scheduling policies

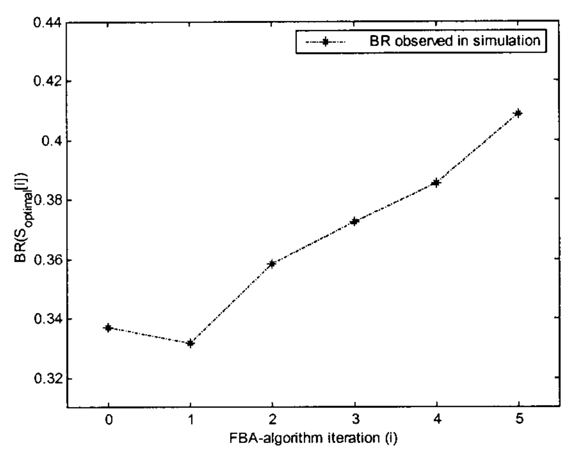


Figure 3.13 BR at $\tilde{S}_{optimal}[i]$, i=0,1,..,5

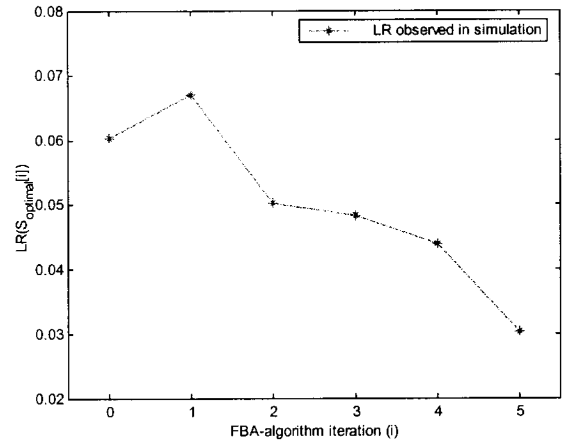


Figure 3.14 LR at $\tilde{s}_{optimal}[i]$, i=0,1,..,5

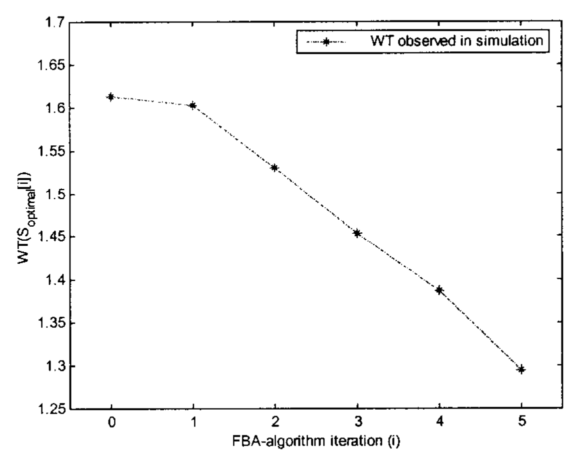


Figure 3.15 WT at $\tilde{s}_{optimal}[i]$, i=0,1,..,5

3.3 Summary of Main Contributions

In this chapter, we have shown that the search for the optimal broadcast scheduling scheme ($S_{optimal}$), in a single-cell hybrid WDB system with user retrials, is a complex constrained optimization problem. However, we have also demonstrated that the given problem can be considerably simplified. Subsequently, we have proposed an algorithm for fast estimation (i.e. approximation) of $S_{optimal}$. The presented experimental results confirm that the proposed algorithm requires minimum computation, while resulting in performance almost identical to the one based on the true $S_{optimal}$.

Although the proposed algorithm is quite general and could be applied in a wide range of hybrid WDB systems, it appears particularly important in view of the so-called networks of pervasive devices (see Chapter 1). Namely, networks of pervasive devices are envisioned as highly dynamic systems, with thousands of users simultaneously attempting information retrieval, using low-cost low-power devices. Thus, the ability to perform a rapid and timely estimation of $S_{optimal}$ seems crucial in such environments, as broadcasting according to $S_{optimal}$ may be the most effective, if not the only way of dealing with excessive load, while not compromising on the provided throughput/QoS/GoS, or affecting the end-devices' power and CPU consumption.

Chapter 4

Single-Cell Hybrid WDB Systems With Variable Mobilities

Overview

Our preceding analysis of single-cell hybrid WDB systems has been based on the assumption of a homogeneous user population, i.e. user population that follows the same or similar behavioural and/or mobility patterns. (Please note, this assumption is commonly found in the previous research works on WDB.) However, in this chapter we extend the earlier discussion to the case of single-cell hybrid WDB systems with a heterogeneous user population. In particular,

Section 4.1 First, we introduce the concept of 'dwell-time random variable' and explain its significance in characterizing users' mobility. In addition, we outline the main difference between the dwell-time random variable models suited for the analysis of macro- and micro- cellular systems.

Section 4.2 Next, we argue that with a minor modification, our realistic-impatient MU behaviour model (see Section 2.2.2) can be used in representing diverse mobility patterns, and we show that such a modified model results in a dwell-time random variable that is proven suitable for the analysis of micro-cellular systems with variable user mobilities. Subsequently, by extending on the discussion from Chapters 2 and 3, we propose a modified Fast Broadcast Scheduling Scheme Estimation (FBE) algorithm for the optimal broadcast scheduling scheme estimation in WDB systems with variable user mobilities. Examples of the obtained experimental results suggest that the proposed algorithm requires minimum computation, while providing performance very close to $S_{optimal}$ found through numerical estimation.

Section 4.3 Finally, we show that in the presence of variable user mobilities, the estimation of data-item request probabilities becomes an increasingly complex problem, as it requires that the actual knowledge of various user mobility patterns be taken into consideration.

4.1 Introduction

4.1.1 Dwell Time and Mobility Modeling

Most wireless cellular systems are expected to deliver a variety of services to many types of users/platforms having a wide range of mobility characteristics (for an illustration see Figure 4.1). The essential feature in characterizing a user's/platform's mobility is the so-called *dwell time*, also known as *cell residence time* or *cell occupancy time* [*Orlik*], [*Fang*], [*Ortigoza-Guerrero*] (see Definition 4.1).

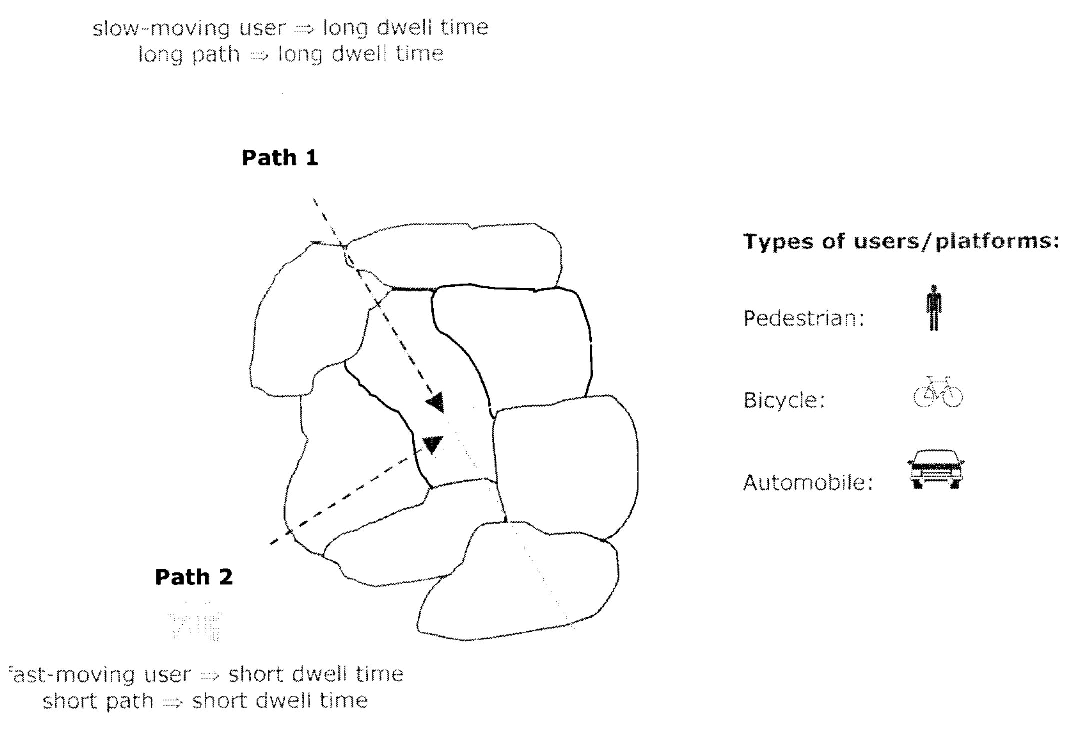


Figure 4.1 Cellular system with variable user mobilities

Definition 4.1 The dwell time of a platform/user is as a random variable that corresponds to the time that the platform/user spends in a cell. In other words, it represents the difference between the time when the platform/user leaves the cell and the time when it enters the cell.

In general, a platform's dwell time depends on many factors, including: speed, path, transmitting power, signal propagation, and interference (Figure 4.1.).

In the previous research works on teletraffic aspects of mobile cellular systems (see [*Orlik*] [*Fang*]), the coefficient of variation (κ) of a random variable (r.v.) has been proposed as an important factor in evaluating the r.v.'s flexibility as a mobility, i.e. dwell time, model (see Definition 4.2).

Definition 4.2 The coefficient of variation (κ) of a random variable X is the ratio of its standard deviation to its mean. Put another way, κ represents the random variable's dispersion around the mean relative the mean (see 4.1).

$$\kappa^2(X) = \frac{\mathrm{var}(X)}{E[X]} \tag{4.1}$$

(1) macro-cellular systems + uniform mobility ⇒ dwell time models with $\kappa \leq 1$

As observed in [*Orlik*], the mobility models that result in a dwell time r.v. with $\kappa \leq 1$ seem to be reasonable only in the case of macro-cellular systems with users of homogeneous mobility characteristics. Namely, in macro-cellular systems, cell sizes are relatively large compared to their shape and possible path variations. Hence, if in such a system most users follow the same mobility patterns, then it is also to expect that most of the users experience the same or similar cell residence times. A typical example of random variables with $\kappa \leq 1$ is the well-known negative exponential r.v.

(2) *micro-cellular systems + variable mobility* $\Rightarrow$ *dwell time models with* $\kappa \geq 1$

In contrast to macro-cellular systems, micro-cellular systems have smaller cells, which lead to smaller mean dwell times and greater dwell time variability. Furthermore, the transmitting power in micro-cellular systems is reduced, hence the obstacles such as building, trees, hills, etc. have a greater effect on cell sizes and shapes. In other words, cells in micro-cellular systems tend to be less regular in shape and more variable throughout the coverage area (for an illustration see Figure 4.1). As a result of the large variation in cell sizes and shapes, together with the shorter mean dwell time, micro-cellular systems clearly require the utilization of dwell-time r.v. models with coefficients of variation that are greater than or equal to one ($\kappa \geq 1$).

In [*Orlik*], two different dwell time models suitable for the analysis of micro-cellular systems with users of variable mobility (i.e. dwell time models with $\kappa \geq 1$), have been proposed and discussed. These are: 1) hyper-exponential pdf, and 2) sum of hyper-exponentials. In this work, we focus on the first model only. In particular, in Section 4.1.2, we present the hyper exponential pdf and justify its utilization as a dwell time r.v. model in micro-cellular systems. Subsequently, in Section 4.2, by employing this model, we extend our analysis from Chapters 2 and 3 to the case of a single-cell hybrid WDB system with variable mobilities.

4.1.2 Hyper-Exponential PDF

K-stage hyper-exponential pdf is a weighted sum (i.e. convex combination) of K exponential functions (see (4.2)).

$$f_X(t) = \sum_{k=1}^{K} q(k) \cdot \mu(k) \cdot e^{-\mu(k)t} \tag{4.2}$$

(Note, in (4.2), q(k)≥0, ∀k=1,..,K, and $\sum_{k=1}^{K} q(k) = 1$.) The mean and variance of the K-stage hyper-exponential pdf are given in (4.3) and (4.4), respectively.

$$E[X] = \sum_{k=1}^{K} \frac{q(k)}{\mu(k)} \tag{4.3}$$

$$\sigma^2 = \left[\sum_{k=1}^{K} \frac{2 \cdot q(k)}{[\mu(k)]^2}\right] - (E[X])^2 \tag{4.4}$$

Based on (4.3) and (4.4), the corresponding coefficient of variation (κ) can be obtained, as shown in (4.5) [1].

$$\kappa = \sqrt{\frac{\sigma^2}{(E[X])^2}} = \sqrt{\frac{\left[\sum_{k=1}^{K} \frac{2 \cdot q(k)}{[\mu(k)]^2}\right] - (E[X])^2}{(E[X])^2}} \geq 1 \tag{4.5}$$

Based on (4.5), it is clear that the K-stage hyper-exponential pdf satisfies the main requirement for being a suitable dwell time r.v. model in the case of micro-cellular systems with variable mobilities, as its κ≥1 at all times (see the discussion of Section 4.1.1).

It should be observed, though, that in addition to satisfying the given requirement, K-stage hyper-exponential pdf appears as an appealing dwell time r.v. model from the practical point of view as well. Namely, one should be able to easily relate this pdf with the cell residence time of a platform moving through a micro-cellular system. E.g. each time the platform enters a new cell, it chooses a stage from the K stages available for its dwell time in the new cell. (The actual

[1] Please note, the exact proof of the inequality in (4.5) can be found in [Orlik]. Also, note that with appropriate choices of q(k) and μ(k), k=1,..,K, the coefficient of variation in (4.4) can be set at any given value greater than one.

choice of the stage depends on the platform's speed, path, transmitting power, etc.) The stage k (k=1,..,K) is chosen with probability q(k). Upon completion if this stage, the platform will enter another cell, where it will again choose a new stage [*Orlik*].

4.2 Optimal Broadcast Schedule in WDB Systems with Variable Mobilities

4.2.1 Motivation – Dwell Time in Single-Cell Hybrid WDB Systems with User Retrials

The user behaviour model that we employ in Chapters 2 and 3 has been characterized with 2 main parameters[2] (see Section 2.2.1 and Figure 2.4):

(1) reattempt arrival rate (λ^*) [attempt/time unit]

This parameter corresponds to the frequency of request resubmission due to failed MU's attempts to seize a point-to-point connection. From the practical point of view, λ^* can be interpreted as the pace at which human-users reissue their requests, at the application layer, during the times of heavy network load and congestion.

In general, one might argue that in reality more than one value of λ^* exists. However, we believe it is reasonable to assume that most humans reissue their requests at the same or similar pace, when facing a congested network.

[2] The third parameter in Figure 2.4 ($P_{success}$) is not directly related to MUs' behaviour. $P_{success}$ describes the quality of the system's performance. For example, the larger the number of unicast connections that the system can simultaneously support, the larger $P_{success}$.

(2) probability of request resubmission ($P_{reattempt}$)

This parameter corresponds to the probability that a MU remains in the system and issues a new request after a previously unsuccessful request submission. In contrast to λ^*, $P_{reattempt}$ is not necessarily related to any of the protocols employed by the MUs' end-devices. Hence, in some cases, it is possible that $P_{reattempt}$ does not take the same value for all users residing in a cell.

Nevertheless, in order to obtain a better insight into the nature of $P_{reattempt}$ and determine when it might be appropriate to assume more than one value of this parameter, let us observe a MU's behaviour in an environment with no external stimuli, i.e. in a system with no unicast or broadcast service. (Please note, if there is no unicast service, then $P_{success}$=0, and accordingly a=$P_{reattempt}$.) Hence, in such a system, based on (2.53), probability density function $p(t_d \mid t_a = \tau_1, \text{item} = i)$ becomes

$$p(t_d \mid t_a = \tau_1, \text{item} = i)\Big|_{P_{success}=0} = \lambda^*(1 - P_{reattempt})e^{-\lambda^*(1-P_{reattempt})(t_d-\tau_1)} \tag{4.6}$$

Under the given assumption of no unicast or broadcast service being present in the system, (4.6) can be interpreted as the probability density function of the overall time that a MU spends in a particular cell waiting for a requested item, , upon his arrival at time $t_a=\tau_1$. Put another way, (4.5) corresponds to the users' *dwell-time*, as defined in Section 4.1.1, where $t_{dwell\text{-}time}=(t_d-t_a)$ [time units] and $t_a=\tau_1$. Now, by using the new notation, (4.6) can be rewritten as

$$p(t_{dwell\text{-}time}) = \lambda^*(1 - P_{reattempt}) \cdot e^{-\lambda^*(1-P_{reattempt})t_{dwell\text{-}time}} \tag{4.7}$$

and, subsequently, the following important conclusions can be made:

dwell-time property 1 The actual form of (4.7) suggests that in the case of MU behaviour model as assumed in the earlier chapters, the corresponding $t_{dwell\text{-}time}$ turns out to be an exponentially distributed random variable with the mean value of $1/\lambda^*(1-P_{reattempt})$ (see (4.8)).

$$E[t_{dwell\text{-}time}] = \frac{1}{\lambda^*(1-P_{reattempt})} \tag{4.8}$$

Please recall from the Section 4.1.2, the negative exponential pdf represents a valid dwell-time r.v. model in macro cellular systems of small cell-size variation and with uniform user-mobility.

dwell-time property 2 There is a direct relationship between $P_{reattempt}$ and MUs' dwell time (see (4.7) and (4.8)). Hence, if the average user dwell-time is known, then from (4.8) the actual value of the corresponding $P_{reattempt}$ can be easily found, and vice versa. Also note, a longer $E[t_{dwell\text{-}time}]$ would correspond to a larger $P_{reattempt}$, and the other way around.

dwell-time property 3 Based on both property 1 and 2, it is clear that the assumption of single-valued $P_{reattempt}$ (i.e. single negative exponential dwell time r.v. model) is well suited for the analysis of macro cellular hybrid WDB systems with homogeneous platforms and/or homogeneous user mobilities. However, the analysis of micro cellular WDB systems with heterogeneous platforms and variable mobilities would require that more than one value of $P_{reattempt}$ be assumed.

Please note, in addition to micro cellular WDB systems with variable mobilities, the examples of other real-world scenarios which would require that more than one $P_{reattemt}$ (i.e. dwell time) be assumed, are as follows:

- ***Users' impatience*** – It is known that MUs tend to withdraw from the waiting (i.e. data retrieval) procedure if it takes too long or requires too many resubmission attempts. However, the level of MUs' (im)patience is not always the same – it varies among users and generally depends on the significance of data being retrieved. Hence, in any system that provides the access to information of different significance levels, more than a single value of $P_{reattempt}$ may have to be assumed.

- ***Time sensitive applications*** - In the case of time-sensitive applications, the item requested by a MU should be retrieved within a specified time interval, otherwise the corresponding information becomes stale or progressively less relevant. Consequently, only resubmission attempts made within the specified time interval can be considered useful, while all the later resubmission attempts can be ignored, as they lead to the retrieval of outdated information. Thus, in a time-sensitive application environment the actual number of useful resubmission attempts made by a user depends not only on his patience and mobility, but also on the time-sensitivity of the data being retrieved. Accordingly, if a time-sensitive application environment provides access to information of different time-sensitivity levels, multiple $P_{reattempts}$ may have to be employed.

Lemma 4.1 The assumption of multiple-valued $P_{reattempt}$ results in a hyper-exponential dwell time r.v. model. Consequently, this assumption is well suited for the analysis of micro cellular hybrid WDB systems with variable user mobilities.

Proof of Lemma 4.1

Let us assume that there are K possible $P_{reattempt}$ values ($P_{reattempt}(k)$, k=1,..,K). Each user gets associated with $P_{reattempt}(k)$ with probability q(k). Based on (4.6), the dwell time r.v. model of a user that is associated with $P_{reattempt}(k)$ (i.e. $t_{dwell\text{-}time}(k)$) corresponds a negative exponential pdf (see (4.9)).

$$p(t_{dwell\text{-}time}(k)) = \mu(k) \cdot e^{-\mu(k) \cdot t_{dwell\text{-}time}(k)} \tag{4.9}$$

In (4.9), $\mu(k)=\lambda^*(1-P_{reattempt}(k))$. However, as the user becomes associated with $P_{reattempt}(k)$ only with probability q(k), the actual model of the user's dwell time turns out to be of the following form,

$$p(t_{dwell-time}) = \sum_{k=1}^{K} q(k) \cdot p(t_{dwell-time}(k)) = \sum_{k=1}^{K} q(k) \cdot \mu(k) \cdot e^{-\mu(k) \cdot t_{dwell-time}(k)} \quad (4.10)$$

(4.10), clearly, corresponds to the pdf of a hyper-exponential r.v. (see (4.2)), which is already proven to be an appropriate dwell time model in the analysis of micro-cell systems with heterogeneous platforms and variable mobilities (see Section 4.1.2).

End of Proof

4.2.2 Optimal Broadcast Schedule in WDB Systems with Variable Mobilities - Approximate Closed-form Solution

4.2.2.1 Theoretical Background

As indicated in the preceding section, the user behaviour model that we employ in Chapters 2 and 3 has been well suited for the analysis of macro-cell hybrid WDB systems with homogeneous platforms and uniform user mobilities. In this section, however, by assuming that the parameter $P_{reattempt}$ can take more than one value (i.e. by assuming a hyper-exponential dwell time r.v. model – see Lemma 4.1), we extend our earlier discussion to the case of micro-cell hybrid WDB system with multiple classes of users. (Here, 'class of users' is a rather general concept, and it refers to the group of users that share the same or similar path, mobility characteristics, data-retrieval (im)patience level, etc. – for more see Section 4.2.1.)

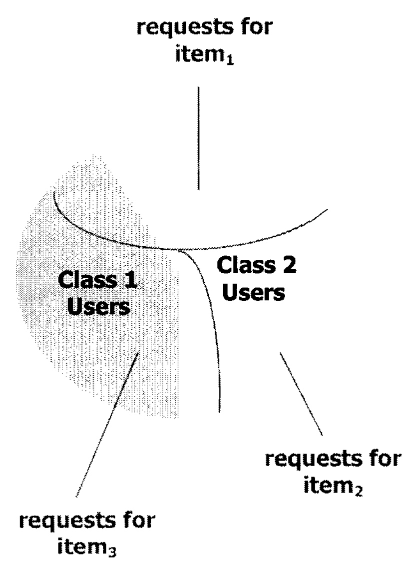


Figure 4.2 Request distribution over multiple classes of users

In particular, let us assume a micro-cell hybrid WDB system with K classes of users. Each class is characterized with its own $P_{reattempt}(k)$ (k=1,..,K) and, accordingly, each class gets associated with its own a(k), as defined in (4.11).

$$a(k) = (1 - P_{success})P_{reattempt}(k), \quad (k = 1,.., K) \tag{4.11}$$

Furthermore, let us assume that in the given system, every item(i) (i=1,..,M) can be requested by every class(k) MU (k=1,..,K), with certain probability. (For an illustration, see Figure 4.2) If we annotate the probability of item(i) being requested by a class(k) MU with q(i,k), as given in (4.12),

$$q(i,k) = p(\text{MU wants to obtain item(i)} \cap \text{MU belongs to class(k)}) \tag{4.12}$$

then (4.13) must hold.

$$\sum_{k=1}^{K} q(i,k) = p(i), \qquad i = 1,.., M \tag{4.13}$$

(Please recall, in (4.13) p(i) represents the probability of item(i) (i=1,..,M) being requested by any MU, regardless of his class. Also note that the above assumptions match the assumptions of Lemma 4.1, and hence result in a hyper-exponential dwell time r.v. model.)

Now, by re-employing the results from Section 2.3, in (4.14) we provide the expression for the probability that a class(k) MU (k=1,..,K), who is interested in obtaining item(i) and has arrived in the system at $t_a=\tau \in [0,s(i))$, eventually gets served through the broadcast channel.

$$P(t_d \geq s(i) \mid t_a = \tau, \text{item} = i, \text{class} = k, t_{br}(i) = s(i)) = e^{-\lambda^*(1-a_k)(s(i)-\tau)} \tag{4.14}$$

Accordingly, the expected number of class(k) users who obtain item(i) through the broadcast channel at time s(i) is

$$E[n_{BR}(0, s(i), item(i), s(i), class(k))] = \frac{\lambda q(i,k)}{\lambda^*(1-a(k))}(1 - e^{-\lambda^*(1-a(k))s(i)}) \quad (4.15)$$

(4.16) represents the expected number of MUs who obtain item(i) through the broadcast channel, at time s(i) irrespective of their class.

$$E[n_{BR}(0, s(i), item(i), s(i), class(k) \mid k = 1,.., K)] = \quad (4.16)$$
$$= \sum_{k=1}^{K} \frac{\lambda q(i,k)}{\lambda^*(1-a(k))}(1 - e^{-\lambda^*(1-a(k))s(i)})$$

Similarly, (4.17) represents the expected number of MUs served through the broadcast channel within the duration of an entire broadcast schedule, irrespective of their class or the item they want to retrieve.

$$E[n_{BR}(0, C, item(i), j \cdot s(i), classk(k) \mid j = 1,.., C/s(i); i = 1,.., M; k = 1,.., K)] = \quad (4.17)$$
$$= \sum_{i=1}^{M} E[n_{BR}(0, C, item(i), j \cdot s(i), classk(k) \mid j = 1,.., C/s(i); k = 1,.., K)] =$$
$$= C\lambda \sum_{i=1}^{M} \sum_{k=1}^{K} \frac{q(i,k)}{\lambda^*(1-a(k))s(i)}(1 - e^{-\lambda^*(1-a(k))s(i)})$$

Finally, the expression for the overall broadcast service ratio, in a system with K classes of users, is given in (4.18).

$$BR(s(1), s(2),.., s(M)) = \sum_{i=1}^{M} \sum_{k=1}^{K} \frac{q(i,k)}{\lambda^*(1-a(k))s(i)}(1 - e^{-\lambda^*(1-a(k))s(i)}) \quad (4.18)$$

(In a similar manner, the other results from Section 2.3 can be readily extended to the case of a WDB system with multiple classes of users. Also, it can be easily proven that the maximization of BR in such a system ensures the minimization of the other performance measures - UR, LR, WT, RP, and WP, exactly as it has been the case for a WDB system with a single class of users.)

Now that the expression for BR is known, let us find the broadcast schedule (S={s(1),s(2),..,s(M)}) that provides for the maximization of BR. In other words, let us find the solution to the following optimization problem.

$$\text{maximize} \quad BR(s(1),s(2),..,s(M)) = \sum_{i=1}^{M}\sum_{k=1}^{K}\frac{q(i,k)}{s(i)\lambda^{*}(1-a(k))}(1-e^{-\lambda^{*}(1-a(k))s(i)}) \qquad (4.19.a)$$

$$\text{subject to} \quad C(s(1),s(2),..,s(M)) = \sum_{i=1}^{M}\frac{l(i)}{s(i)} - 1 = 0 \qquad (4.19.b)$$

Based on the Lagrange Multiplier Theory, S={s(1),s(2),..,s(M)} that satisfies (4.19) can be obtained as a solution to the system of differential equations presented in (4.20), i.e. (4.21).

$$\frac{\partial BR(s(1),s(2),..,s(M))}{\partial s(i)} + K\frac{\partial C(s(1),s(2),..,s(M))}{\partial s(i)} = 0, \qquad i=1,..,M \qquad (4.20)$$

$$\sum_{k=1}^{K}\frac{q(i,k)}{\lambda^{*}(1-a(k))}\frac{\lambda^{*}(1-a(k))e^{-\lambda^{*}(1-a(k))s(i)}\cdot s(i)-1+e^{-\lambda^{*}(1-a(k))s(i)}}{s(i)^{2}} - K\frac{l_{i}}{s(i)^{2}} = 0, \qquad i=1,..,M \qquad (4.21)$$

Hence, from (4.21), we obtain the implicit form solution to (4.19), as given in (4.22).

$$\frac{1}{l(i)}\sum_{k=1}^{K}\frac{q(i,k)}{(1-a(k))}\left(s(i)\lambda^{*}(1-a(k))e^{-\lambda^{*}(1-a(k))s(i)} + e^{-\lambda^{*}(1-a(k))s(i)} - 1\right) = K\lambda^{*} = \text{const}, \qquad i=1,..,M \qquad (4.22)$$

Again, based on the analogy with the discussion from Section 3.2.2, the following approximation can be employed in (4.22).

$$\left(s(i)\lambda^{*}(1-a(k))e^{-\lambda^{*}(1-a(k))s(i)} + e^{-\lambda^{*}(1-a(k))s(i)} - 1\right) = F(s(i)\lambda^{*}(1-a(k))) \qquad (4.23)$$

where,

$$F(s(i)\lambda^*(1-a(k))) \approx \begin{cases} k \cdot s(i)\lambda^*(1-a(k)), & s(i)\lambda^*(1-a(k)) < \chi \\ -1, & s(i)\lambda^*(1-a(k)) \geq \chi \end{cases} \tag{4.24}$$

(Please note, $F(s_i\lambda^*(1-a_k))$ in (4.23), i.e. (4.24), is a generalized form of function F(x(i)) from (3.29). Namely, in contrast to the case studied in Chapter 3 (where we assumed a single $P_{reattempt}$, i.e. a single parameter a), here the function on the left-hand side of (4.24) depends on both s(i) and a(k).)

By following the discussion from Section 3.2.2 further (see (3.30)), let us assume that for an arbitrary number of items ($M_1 \leq M$) their s(i) are such that $s(i)\lambda^*(1-a(k)) \in [0,\chi)$ for at least one class of users (i.e. one k, k=1,..,K). If we use $\mathcal{M}_1$ to annotate the set of indexes of these M_1 items, and $\mathcal{N}_{i1}$ to annotate the set of indexes of those classes that for a given $i \in \mathcal{M}_1$ satisfy $s(i)\lambda^*(1-a(k)) \in [0,\chi)$, then (4.25) must hold.

$$F(s(i)\lambda^*(1-a(k))) \approx k \cdot s(i)\lambda^*(1-a(k)), \quad \forall i \in \mathcal{M}_1, \quad \forall k \in \mathcal{N}_{i1} \tag{4.25}$$

Similarly, if we use $\mathcal{N}_{i2}$ to annotate the set of indexes of those classes that for any given $i \in \mathcal{M}_1$ do not satisfy $s(i)\lambda^*(1-a(k)) \in [0,\chi)$, then (4.26) is obtained.

$$F(s(i)\lambda^*(1-a(k))) \approx -1, \quad \forall i \in \mathcal{M}_1, \quad \forall k \in \mathcal{N}_{i2} \tag{4.26}$$

For the remaining $M_2=(M-M_1)$ items the corresponding $s(i)\lambda^*(1-a(k))$ falls into interval $[\chi,\infty)$, for every class of users (i.e. every k, k=1,..,K). Thus, if we annotate the set of indexes of these M_2 items with $\mathcal{M}_2$, then

$$F(s(i)\lambda^*(1-a(k))) \approx -1, \quad \forall i \in \mathcal{M}_2, \quad \forall k = 1,..,K \tag{4.27}$$

Now, the cost function (BR) from (4.19) can also be presented in the form of (4.28)

$$BR(s(1), s(2),.., s(M)) = \sum_{i=1}^{M}\sum_{k=1}^{K} q(i,k) f(s(i)\lambda^{*}(1-a(k))) \tag{4.28}$$

where $f(s(i)\lambda^{*}(1\text{-}a(k)))$ stands for

$$f(s(i)\lambda^{*}(1-a(k))) = \frac{(1-e^{-s(i)\lambda^{*}(1-a(k))})}{s(i)\lambda^{*}(1-a(k))} \tag{4.29}$$

Accordingly

$$\frac{\partial f(s(i)\lambda^{*}(1-a(k)))}{\partial s(i)} = \lambda^{*}(1\text{-}a(k))B(s(i)\lambda^{*}(1\text{-}a(k))) \tag{4.30}$$

where

$$B(s(i)\lambda^{*}(1\text{-}a(k))) = \frac{s(i)\lambda^{*}(1\text{-}a(k))e^{-\lambda^{*}(1\text{-}a(k))s(i)} + e^{-\lambda^{*}(1\text{-}a(k))s(i)} - 1}{\left(s(i)\lambda^{*}(1\text{-}a(k))\right)^{2}} \tag{4.31}$$

Again, please note $B(s(i)\lambda^{*}(1\text{-}a(k)))$, in (4.31), is a generalized form of function $B(x(i))$ from (3.27). Based on the observation that $B(s(i)\lambda^{*}(1\text{-}a(k)))\approx 0$ (i.e. $\partial f(s(i)\lambda^{*}(1\text{-}a(k)))/\partial s(i)\approx 0$) for $\forall$ $s(i)\lambda^{*}(1\text{-}a(k))\in[\chi,\infty)$ (see Section 3.2.2, Figure 3.2 and expression (3.33)), the following can be easily proven.

$$f(s(i)\lambda^{*}(1-a(k))) \approx f(\infty) \to 0, \quad \forall\ s(i)\lambda^{*}(1-a(k)) \in [\chi,\infty) \tag{4.32}$$

Thus, as $s(i)\lambda^{*}(1\text{-}a(k))\notin[\chi,\infty)$ (i.e. $f(s(i)\lambda^{*}(1\text{-}a(k)))\neq 0$) only for $i\in\mathcal{M}_1$ and $k\in\mathcal{N}_{i1}$, an approximation of (4.28) is obtained, as given in (4.33).

$$BR(s(1), s(2),.., s(M)) \approx \sum_{i\in\mathcal{M}_1}\sum_{k\in\mathcal{N}_{i1}} q(i,k) f(s(i)\lambda^{*}(1-a(k))) \tag{4.33}$$

Accordingly, by employing (4.33) in (4.20), now (4.21) and (4.22) get transformed into (4.34) and (4.35) respectively.

$$\sum_{k\in\mathcal{N}_{i1}} \frac{q(i,k)}{\lambda^*(1-a(k))} \frac{\lambda^*(1-a(k))e^{-\lambda^*(1-a(k))s(i)}\cdot s(i)-1+e^{-\lambda^*(1-a(k))s(i)}}{s(i)^2} - K\frac{l(i)}{s(i)^2}=0,\quad i\in\mathcal{M}_1 \tag{4.34}$$

$$\frac{1}{l(i)}\sum_{k\in\mathcal{N}_{i1}} \frac{q(i,k)}{(1-a(k))}\left(s(i)\lambda^*(1-a(k))e^{-\lambda^*(1-a(k))s(i)}+e^{-\lambda^*(1-a(k))s(i)}-1\right)=K\lambda^*=\text{const},\quad i\in\mathcal{M}_1 \tag{4.35}$$

Based on (4.25), (4.35) becomes (4.36), i.e. (4.37).

$$\frac{1}{l(i)}\sum_{k\in\mathcal{N}_{i1}} \frac{q(i,k)}{(1-a(k))} k\cdot s(i)\lambda^*(1-a(k))=K\lambda^*=\text{const},\quad i\in\mathcal{M}_1 \tag{4.36}$$

$$\frac{k\cdot s(i)}{l(i)}\sum_{k\in\mathcal{N}_{i1}} q(i,k)=K=\text{const},\quad i\in\mathcal{M}_1 \tag{4.37}$$

From (4.37), we obtain an approximate explicit-form expression for s_i, $i\in\mathcal{M}_1$ (see (4.38)), although the Lagrange Multiplier constant (K) in this expression is still unknown.

$$s(i)=\frac{l(i)\cdot K}{k\cdot\sum_{k\in\mathcal{N}_{i1}} q(i,k)},\quad i\in\mathcal{M}_1 \tag{4.38}$$

The actual value of K, as given in (4.39), is found by employing (4.38) back in (4.19.b).

$$K=k\cdot\sum_{i\in\mathcal{M}_1}\sum_{k\in\mathcal{N}_{i1}} q(i,k) \tag{4.39}$$

Finally, by replacing K in (4.38) with the expression provided in (4.39), we obtain the approximate closed-form solution to (4.19), as presented in (4.40).

$$s(i) = \frac{\frac{l(i)}{\sum_{k \in \mathcal{N}_{i1}} q(i,k)}}{\sum_{i \in \mathcal{M}_1} \sum_{k \in \mathcal{N}_{i1}} q(i,k)}, \quad i \in \mathcal{M}_1 \tag{4.40}$$

If we annotate the set of inter-broadcast intervals obtained by employing (4.41) with $\widetilde{S}_{optimal}$, then based on the preceding discussion, broadcasting according to $\widetilde{S}_{optimal}$ is expected to result in BR nearly as large as it would be obtained if broadcasting was done according to the actual $S_{optimal}$. Thus, we can write

$$BR(S_{optimal}) \approx BR(\widetilde{S}_{optimal}) \tag{4.41}$$

4.2.2.2 *Modified FBE Algorithm – Algorithm for Estimation of $S_{optimal}$ in Hybrid Single-Cell WDB System with Multiple Classes of Users*

By simple inspection, it can be observed that (4.40) resembles the corresponding expression, concerning the estimation of inter-broadcast intervals, from Section 3.2.2, i.e. (3.44)[3]. However, it should be noted that (4.40) employs $\sum_{k \in \mathcal{N}_{i1}} q(i,k)$ in the place of the full p(i) ($i \in \mathcal{M}_1$). More specifically, when estimating s(i) (i=1,..,M) in a WDB system with multiple classes of users, the probability of item(i) being requested by a MU from either of the classes (p(i)) gets substituted with the probability of item(i) being requested by a MU only from those classes that eventually satisfy $s(i)\lambda^*(1\text{-}a(k)) \in [0,\chi)$ ($\sum_{k \in \mathcal{N}_{i1}} q(i,k)$).

The above observation regarding the correspondence of the two expressions ((3.44) and (4.40)) suggests that the earlier proposed FBE algorithm (which has been derived from (3.44)) can serve

[3] Please recall, (3.44) applies to single-cell hybrid WDB systems with a single class of users.

Step 1 For every item(i) its length l(i) and relative popularity with respect to each class of users q(i,k) are known, i=1,..,M, k=1,..,K. Assume that all items require 'first-segment' approximation (see (4.25)), i.e. $\mathcal{N}_{i1}=\{1,..,K\}$ for $\forall i \in \mathcal{M}_1=\{1,2,..,M\}$.

Step 2 For $\forall\ i \in \mathcal{M}_1$ calculate

$$s(i) = \frac{l(i)}{\dfrac{\sum_{k\in\mathcal{N}_{i1}} q(i,k)}{\sum_{i\in\mathcal{M}_1}\sum_{k\in\mathcal{N}_{i1}} q(i,k)}}$$

Step 3 $\mathcal{E}=\varnothing$, $\mathcal{F}=\varnothing$.

Among all $\{s(i) \mid i \in \mathcal{M}_1\}$ attempt to find the one that most considerably violates following constraint, for either class of users (i.e. for either a(k), $k \in \mathcal{N}_{i1}$).

$$s(i) \le \frac{\chi}{\lambda^*(1-a(k))} \approx \frac{5}{\lambda^*(1-a(k))}$$

Annotate the index of such item with u, and the index of the corresponding class with v. Set $\mathcal{E}=\{u\}$, and $\mathcal{F}=\{v\}$. If more than one s(i) is found in Step 3, proceed with the one that corresponds to the smallest $\sum_{k\in Ni1/\mathcal{F}} q(i,k)$[4].

Step 4 If $\mathcal{E}\neq\varnothing$ then proceed with Steps 4.a and 4.b. Otherwise proceed with Step 5.

Step 4.a $\mathcal{N}_{u1}=\mathcal{N}_{u1}/\mathcal{F}$

Step 4.b If $\mathcal{N}_{u1}=\{\}$ then $\mathcal{M}_1=\mathcal{M}_1/\mathcal{E}$. Go to Step 2.

Step 5 $\tilde{S}_{optimal} = \{s(i) \mid i \in \mathcal{M}_1\}$. End.

Figure 4.3 Modified FBE algorithm for single-cell hybrid WDB systems with multiple classes of users / variable user mobilities

[4] As a consequence of Steps 4.a, s(i) selected in Step 3 eventually gets increased (i.e the corresponding item gets broadcasted less frequently). Thus, among several s(i) that might meet Step 3 criterion, the algorithm proceeds with, i.e. increases, the one that is least significant (popular) with respect to the other classes of users.

as a basis for estimation of $\widetilde{S}_{optimal}$. However, clearly, the FBE algorithm needs to be modified so as to include a mechanism for identification, i.e. exclusion, of the q(i,k) terms that correspond to $s(i)\lambda^*(1-a(k))>\chi$ (see the discussion preceding the derivation of (4.32) and (4.33)).

In Figure 4.3 we propose a modified FBE algorithm for estimation of $\widetilde{S}_{optimal}$ in a single-cell hybrid WDB system with multiple classes of users. The algorithm begins by assuming that all items (i.e. all s(i)) correspond to the 'first-segment' approximation (4.25) with respect to all classes of MUs. In particular, $\mathcal{N}_{i1}=\{1,..,K\}$ for $\forall i\in\mathcal{M}_1=\{1,2,..,M\}$. However, if it turns out that some particular s(i) (obtained by employing (4.40)) violates the constraint $s(i)\lambda^*(1-a(k))\leq\chi$, this suggests that the initial assumption regarding the validity of the 'first-segment' approximation for the given pair (s(i),a(k)) was incorrect. Therefore, the corresponding q(i,k) needs to be excluded from $\mathcal{N}_{i1}$. The exclusion starts with q(i,k) that most significantly violates the constraint $s(i)\lambda^*(1-a(k))\leq\chi$, and it is always performed on 'one by one' basis, as already explained in Section 4.2.2.1.

4.2.3 Experimental Results

In this section, through a series of experiments, we compare the actual values and performance of $S_{optimal}$ (obtained by solving (4.19) with Matlab - see Appendix 7 for a more detailed explanation) and $\tilde{S}_{optimal}$ (obtained by employing our modified FBE algorithm as proposed in Figure 4.3).

In all experiments the presence of three different classes of MUs is assumed, each class being identified with its own $\lambda^*(1\text{-}a(k))$[5], k=1,..,3. Furthermore,

1) in Experiment 1.a) through 1.c) it is assumed that all the items are equally popular, while each individual item gets requested only by MUs of a single class. In particular, for every $i \in \{1,..,10\}$ there exists a single $k \in \{1,2,3\}$ such that q(i,k)=0.1, while the other q(i,j)=0, $j=\{1,2,3\}\backslash\{k\}$ (see Table 4.1 through 4.3);
2) in Experiment 2.a) through 2.c) it is assumed that each individual item gets requested by MUs from more than a single class. Still, all the items are assumed equally popular, i.e. $p(i)=\sum q(i,k)=0.1$, for every i={1,..,10} (see Table 4.4 through 4.6).;
3) Experiments 3.a) through 3.c) capture the most general case - the items are assumed to be of unequal popularities, i.e. $p(i)\neq p(j)$ for $i\neq j$ ($i,j \in \{1,..,10\}$), while each individual item gets requested by MUs of more than a single class.

Experiment 1.a)

The results presented in Experiment 1.a) show that, for the given values of H(k) and q(i,k), k=1,..,3, i=1,..,10, $S_{optimal}$ (obtained using Matlab) and $\tilde{S}_{optimal}$ (obtained in the initial evaluation of our modified FBE algorithm) are almost identical. Consequently, these two broadcast scheduling schemes result in equally good WDB system performance (see $BR(S_{optimal})$ and $BR(\tilde{S}_{optimal}(0))$).

[5] In further discussion we annotate $\lambda^*(1\text{-}a(k))$ with H(k).

$H(1)=\lambda^*(1-a(1))=0.1$ $\Rightarrow$ $\chi/\lambda^*(1-a(1))\approx 5/0.1=50$

$H(2)=\lambda^*(1-a(2))=0.2$ $\Rightarrow$ $\chi/\lambda^*(1-a(2))\approx 5/0.2=25$

$H(3)=\lambda^*(1-a(3))=0.3$ $\Rightarrow$ $\chi/\lambda^*(1-a(3))\approx 5/0.3=16.6667$

item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
q (class 1)	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q (class 2)	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0
q (class 3)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	10.0001	10.0001	10.0001	9.9998	9.9998	9.9998	9.9998	10.0001	10.0001	10.0001
$\tilde{S}_{optimal}[0]$	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000

Table 4.1 Experiment 1.a) - Numerical MATLAB solution and closed-form approximation

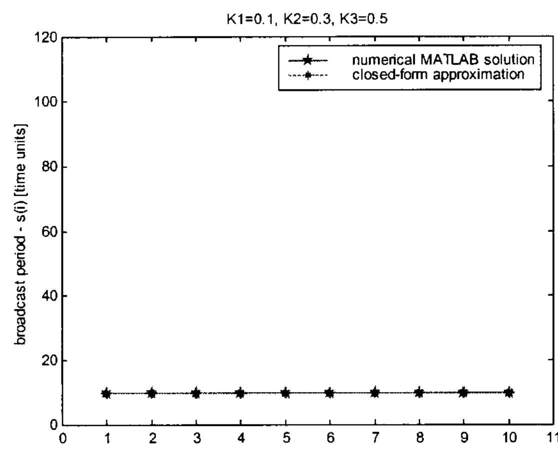


Figure 4.4 Experiment 1.a) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.4576$

(number of evaluations (iterations) = 12)

$BR(\tilde{S}_{optimal}[0]) = 0.4576$

(number of iterations = 0)

error: $\dfrac{BR(S_{optimal}) - BR(\widetilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.0\ [\%]$

Experiment 1.b)

In this experiment, the values of H(2) and H(3) are increased from 0.2 to 0.3, and from 0.3 to 0.5, respectively. As a consequence of the given changes, three s(i) obtained in the initial evaluation of $\widetilde{S}_{optimal}$ (s(8), s(9), and s(10)) are shown to violate the constraint $s(i)\lambda^*(1\text{-}a(k))\leq\chi$ with respect to class(3) (i.e. with respect to k=3). After the exclusion of the given items from the broadcast schedule, according to the rules of the modified FBE algorithm, finally all s(i) of $\widetilde{S}_{optimal}[3]$ are proven to comply with $s(i)\lambda^*(1\text{-}a(k))\leq\chi$, for $\forall k\in\{1,2,3\}$. The presented results (see Table 4.2 and Figure 4.5) verify the similarity between both, $S_{optimal}$ and $\widetilde{S}_{optimal}[3]$, and WDB system performances they provide.

$\mathbf{H(1)=\lambda^*(1\text{-}a(1))=0.1} \Rightarrow \chi/\lambda^*(1\text{-}a(1))\approx 5/0.1=50$

$\mathbf{H(2)=\lambda^*(1\text{-}a(2))=0.3} \Rightarrow \chi/\lambda^*(1\text{-}a(2))\approx 5/0.3=16.6667$

$\mathbf{H(3)=\lambda^*(1\text{-}a(3))=0.5} \Rightarrow \chi/\lambda^*(1\text{-}a(3))\approx 5/0.5=10$

item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
q (class 1)	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q (class 2)	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0
q (class 3)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1
l [time]	1	1	1	1	1	1	1	1	1	1
$\mathbf{S_{optimal}}$	8.2417	8.2417	8.2417	6.7390	6.7390	6.7390	6.7390	70.6964	70.6964	70.6964
$\widetilde{S}_{optimal}[3]$	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	Inf	Inf	Inf

Table 4.2 Experiment 1.b) - Numerical MATLAB solution and closed-form approximation

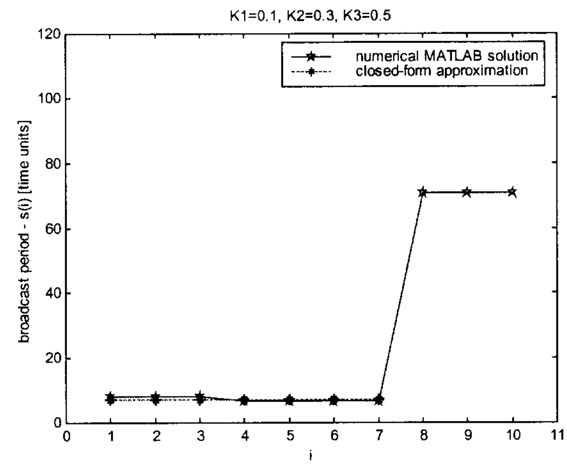


Figure 4.5 Experiment 1.b) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.3845$

(number of evaluations (iterations) = 392)

$BR(\tilde{S}_{optimal}[3]) = 0.3829$

(number of iterations = 3)

error: $\dfrac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.42\,[\%]$

Experiment 1.c)

In Experiment 3.3, the value of H(3) is further increased to 0.7. As a consequence, significantly larger s(8), s(9), and s(10) of $S_{optimal}$ are obtained, when compared to $S_{optimal}$ from Experiment 1.b). However, regardless of this change, the corresponding BR (i.e. $BR(S_{optimal})$) and the other results remain the same as in Experiment 1.b).

$\mathbf{H(1)=\lambda^*(1-a(1))=0.1} \Rightarrow \chi/\lambda^*(1-a(1))\approx 5/0.1=50$

$\mathbf{H(2)=\lambda^*(1-a(2))=0.3} \Rightarrow \chi/\lambda^*(1-a(2))\approx 5/0.3=16.6667$

$\mathbf{H(3)=\lambda^*(1-a(3))=0.7} \Rightarrow \chi/\lambda^*(1-a(3))\approx 5/0.7= 7.1429$

item	1	2	3	4	5	6	7	8	9	10
q (class 1)	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q (class 2)	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0
q (class 3)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	7.9901	7.9901	7.9900	6.4043	6.4055	6.4054	6.4043	3.3403e+005	3.3403e+005	3.3403e+005
$\tilde{S}_{optimal}[3]$	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	Inf	Inf	Inf

Table 4.3 Experiment 1.c) - Numerical MATLAB solution and closed-form approximation

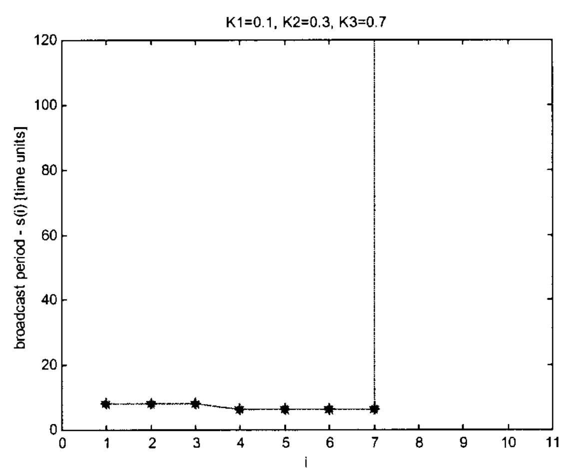


Figure 4.6 Experiment 1.c) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.3843$

(number of evaluations (iterations) = 720)

$BR(\tilde{S}_{optimal}[3]) = 0.3829$

(number of iterations = 3)

error: $$\frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.36\,[\%]$$

Experiment 2.a)

In this experiment, H(1), H(2), and H(3) are the same as in Experiment 1.a), while different values of q(i,k) are employed (see Table 4.1 and 4.4). Similar as in the case of Experiment 1.a), $\tilde{S}_{optimal}$ obtained already in the initial evaluation of the modified FBE algorithm turns out to be a perfect approximation of $S_{optimal}$. Accordingly, these two broadcast scheduling schemes are shown to provide equally good WDB system performance (see $BR(S_{optimal})$ and $BR(\tilde{S}_{optimal}[0])$). (Please recall, the distribution of q(i,k) makes the only difference between this and Experiment 1.a). Namely, in this experiment it is assumes that every item gets requested by MUs from more than a single class. This results in BR being smaller than BR of Experiment 1.a).)

$\mathbf{H(1)=\lambda^*(1\text{-}a(1))=0.1} \Rightarrow \chi/\lambda^*(1\text{-}a(1))\approx 5/0.1=50$

$\mathbf{H(2)=\lambda^*(1\text{-}a(2))=0.2} \Rightarrow \chi/\lambda^*(1\text{-}a(2))\approx 5/0.2=25$

$\mathbf{H(3)=\lambda^*(1\text{-}a(3))=0.3} \Rightarrow \chi/\lambda^*(1\text{-}a(3))\approx 5/0.3=16.6667$

item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
q (class 1)	0.04	0.05	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q (class 2)	0.03	0.05	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0
q (class 3)	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1
l [time]	1	1	1	1	1	1	1	1	1	1
$\mathbf{S_{optimal}}$	10.0001	10.0000	10.0002	9.9998	9.9998	9.9998	9.9998	10.0001	10.0001	10.0001
$\tilde{S}_{optimal}[0]$	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000

Table 4.4 Experiment 2.a) - Numerical MATLAB solution and closed-form approximation

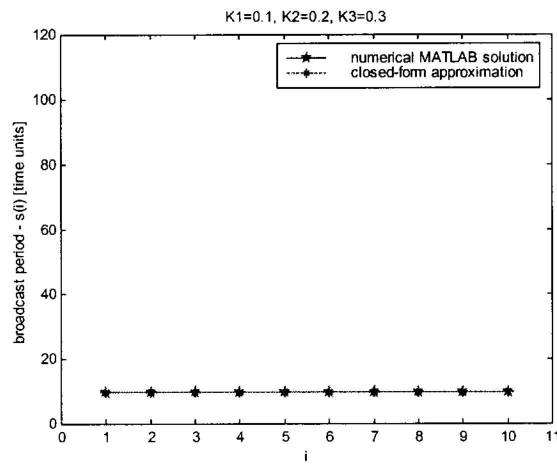


Figure 4.7 Experiment 2.a) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.4321$
(number of evaluations (iterations) = 12)

$BR(\widetilde{S}_{optimal}[0]) = 0.4321$
(number of iterations = 0)

error: $$\frac{BR(S_{optimal}) - BR(\widetilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.0\,[\%]$$

Experiment 2.b)

In this experiment, H(1), H(2), and H(3) are the same as in Experiment 1.b), while the values of q(i,k) (i=1,..,10, k=1,..,3) are as in Experiment 2.a). Similar as in Experiment 1.b), three s(i) obtained in the initial evaluation of $\widetilde{S}_{optimal}$ (s[8], s[9], and s[10]) are shown to violate the constraint $s(i)\lambda^*(1-a(k))\leq\chi$ with respect to class(3). After the exclusion of the given items from the broadcast schedule, finally all s(i) of $\tilde{S}_{optimal}[3]$ are proven to comply with $s(i)\lambda^*(1-a(k))\leq\chi$, for $\forall k\in\{1,2,3\}$. The presented results (see Table 4.5 and Figure 4.8) verify the similarity between both, $S_{optimal}$ and $\tilde{S}_{optimal}[3]$, and WDB system performances they provide.

$H(1)=\lambda^*(1-a(1))=0.1$ $\Rightarrow$ $\chi/\lambda^*(1-a(1))\approx 5/0.1=50$

$H(2)=\lambda^*(1-a(2))=0.3$ $\Rightarrow$ $\chi/\lambda^*(1-a(2))\approx 5/0.3=16.6667$

$H(3)=\lambda^*(1-a(3))=0.5$ $\Rightarrow$ $\chi/\lambda^*(1-a(3))\approx 5/0.5=10$

item	1	2	3	4	5	6	7	8	9	10
q (class 1)	0.04	0.05	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q (class 2)	0.03	0.05	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0
q (class 3)	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	7.9943	7.6295	8.2453	6.7399	6.7399	6.7399	6.7399	103.1804	103.1804	103.1804
$\tilde{S}_{optimal}[3]$	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	Inf	Inf	Inf

Table 4.5 Experiment 2.b) - Numerical MATLAB solution and closed-form approximation

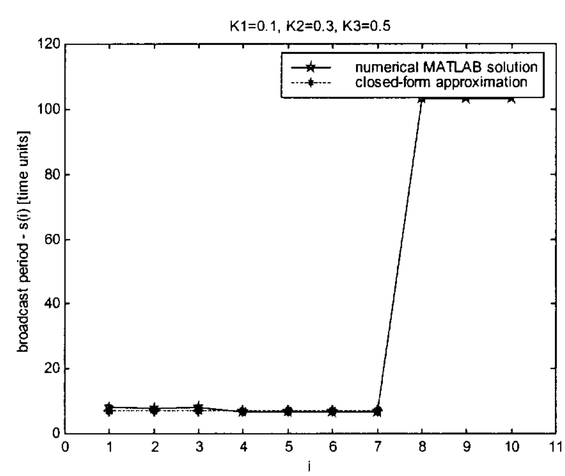


Figure 4.8 Experiment 2.b) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.3464$

(number of evaluations (iterations) = 766)

$BR(\tilde{S}_{optimal}[3]) = 0.3455$

(number of iterations = 3)

error: $\dfrac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.26\,[\%]$

Experiment 2.c)

In this experiment, H(1), H(2), and H(3) are the same as in Experiment 1.c), while the values of q(i,k) (i=1,..,10, k=1,..,3) are as in the preceding two experiments. The presented results indicate that finally all s(i) of $\tilde{S}_{optimal}(3)$ comply with $s(i)\lambda^*(1-a_k)\leq\chi$, for $\forall k\in\{1,2,3\}$. Again, Table 4.6 and Figure 4.9 verify the similarity between $S_{optimal}$ and $\tilde{S}_{optimal}[3]$, while the values of $BR(S_{optimal})$ and $BR(\tilde{S}_{optimal}[3])$ suggest that these two broadcast scheduling schemes result in almost equally good WDB system performance.

$\mathbf{H(1)=\lambda^*(1-a(1))=0.1} \Rightarrow \chi/\lambda^*(1-a(1))\approx 5/0.1=50$

$\mathbf{H(2)=\lambda^*(1-a(2))=0.3} \Rightarrow \chi/\lambda^*(1-a(2))\approx 5/0.3=16.6667$

$\mathbf{H(3)=\lambda^*(1-a(3))=0.7} \Rightarrow \chi/\lambda^*(1-a(3))\approx 5/0.7=7.1429$

item	1	2	3	4	5	6	7	8	9	10
q (class 1)	0.04	0.05	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q (class 2)	0.03	0.05	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0
q (class 3)	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	8.2710	7.4009	8.0536	6.4850	6.4850	6.4850	6.4850	1.0000e+003	1.0000e+003	1.0000e+003
$\tilde{S}_{optimal}[3]$	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	7.0000	Inf	Inf	Inf

Table 4.6 Experiment 2.c) - Numerical MATLAB solution and closed-form approximation

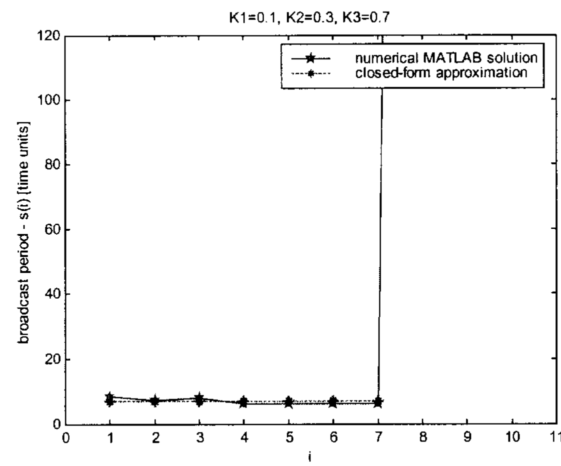


Figure 4.9 Experiment 2.c) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.3442$

(number of evaluations (iterations) = 404)

$BR(\tilde{S}_{optimal}[3]) = 0.3433$

(number of iterations = 3)

error: $$\frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.26\,[\%]$$

Experiment 3.a)

As indicated earlier, in Experiment 3.a) through 3.c) the items are assumed to be of unequal popularities (see Table 4.7). Furthermore, in Experiment 3.a), H(1), H(2), and H(3) are of the same values as in Experiment 1.a) and 2.a). Under such assumptions, all s(i) obtained already in the initial evaluation of $\tilde{S}_{optimal}$ are shown to satisfy the constraint $s(i)\lambda^{*}(1-a(k))\leq\chi$, for

$\forall k \in \{1,2,3\}$. Table 4.7 and Figure 4.10 illustrate the actual similarity between $\tilde{S}_{optimal}[0]$ and $S_{optimal}$ (as estimated using Matlab), while the obtained values of $BR(\tilde{S}_{optimal}[0])$ and $BR(S_{optimal})$ verify that the two broadcast scheduling schemes result in very close WDB system performance.

$$\mathbf{H(1)=\lambda^*(1\text{-}a(1))=0.1} \Rightarrow \chi/\lambda^*(1\text{-}a(1)) \approx 5/0.1=50$$
$$\mathbf{H(2)=\lambda^*(1\text{-}a(2))=0.2} \Rightarrow \chi/\lambda^*(1\text{-}a(2)) \approx 5/0.2=25$$
$$\mathbf{H(3)=\lambda^*(1\text{-}a(3))=0.3} \Rightarrow \chi/\lambda^*(1\text{-}a(3)) \approx 5/0.3=16.6667$$

item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
q (class 1)	0.04	0.05	0.1	0.02	0.03	0.0	0.0	0.0	0.13	0.04
q (class 2)	0.03	0.05	0.0	0.03	0.02	0.05	0.05	0.0	0.03	0.03
q (class 3)	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.04	0.13
l [time]	1	1	1	1	1	1	1	1	1	1
$\mathbf{S_{optimal}}$	9.0048	9.1278	9.5732	20.8158	17.4895	29.5197	29.5197	9.0459	5.5396	4.7416
$\tilde{S}_{optimal}[0]$	10.0000	10.0000	10.0000	20.0000	20.0000	20.0000	20.0000	10.0000	5.0000	5.0000

Table 4.7 Experiment 3.a) - Numerical MATLAB solution and closed-form approximation

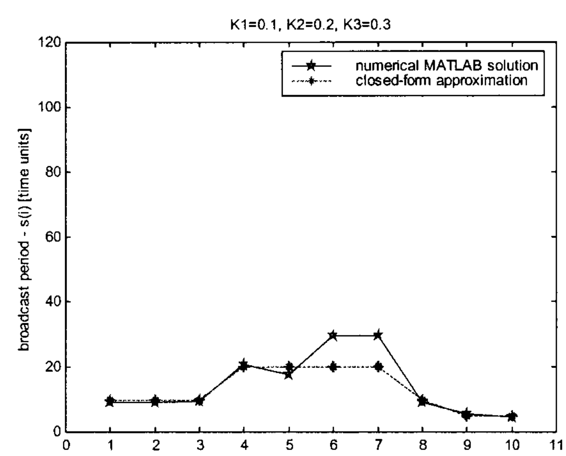


Figure 4.10 Experiment 3.a) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.5153$
(number of evaluations (iterations) = 1009)

$BR(\tilde{S}_{optimal}[0]) = 0.5140$
(number of iterations = 0)

error: $\dfrac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.25\,[\%]$

Experiment 3.6)

In this experiment, for the given values of H(1), H(2) and H(3), four s(i) of $\tilde{S}_{optimal}[0]$ are shown to violate the constraint $s(i)\lambda^*(1\text{-}a(k))\leq\chi$, with respect to k=2,3. By following the procedure presented in Table 4.9, the corresponding q(i,k) are gradually removed until, finally, all s(i) of $\tilde{S}_{optimal}[5]$ comply with the given constraint. The actual similarity between $\tilde{S}_{optimal}[5]$ and $S_{optimal}$ (as estimated using Matlab) is illustrated in Table 4.8 and Figure 4.11. The obtained values of $BR(\tilde{S}_{optimal}[5])$ and $BR(S_{optimal})$ suggest that the two broadcast scheduling schemes result in sufficiently close WDB system performance.

$\mathbf{H(1)=\lambda^*(1\text{-}a(1))=0.1} \Rightarrow \chi/\lambda^*(1\text{-}a(1))\approx 5/0.1=50$
$\mathbf{H(2)=\lambda^*(1\text{-}a(2))=0.3} \Rightarrow \chi/\lambda^*(1\text{-}a(2))\approx 5/0.3=16.6667$
$\mathbf{H(3)=\lambda^*(1\text{-}a(3))=0.5} \Rightarrow \chi/\lambda^*(1\text{-}a(3))\approx 5/0.5=10$

item	1	2	3	4	5	6	7	8	9	10
q (class 1)	0.04	0.05	0.1	0.02	0.03	0.0	0.0	0.0	0.13	0.04
q (class 2)	0.03	0.05	0.0	0.03	0.02	0.05	0.05	0.0	0.03	0.03
q (class 3)	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.04	0.13
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	8.1468	7.7496	8.3423	30.7282	14.9387	59.9898	59.9898	42.0045	4.7382	3.8364
$\tilde{S}_{optimal}[4]$	8.5000	8.5000	8.5000	42.5000	28.3333	Inf	Inf	8.5000	4.2500	4.2500

Table 4.8 Experiment 3.b) - Numerical MATLAB solution and closed-form approximation

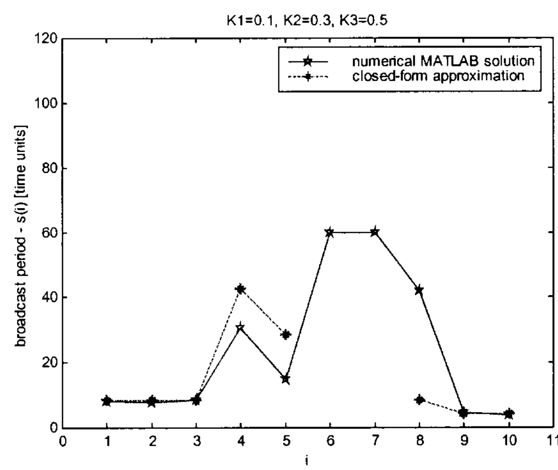


Figure 4.11 Experiment 3.b) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.4513$

(number of evaluations (iterations) = 922)

$BR(\tilde{S}_{optimal}[4]) = 0.4492$

(number of iterations = 4)

error: $$\frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.47\ [\%]$$

Iteration	Steps/Results
1	p = [0.1000 0.1000 0.1000 0.0500 0.0500 0.0500 0.0500 0.1000 0.2000 0.2000] $\tilde{S}_{optimal}$ = [10 10 10 20 20 20 20 10 5 5] s_4, s_5, s_6, s_7 overestimated for class2 and class3 remove q(2,6) (i.e. q(2.6)=0)
2	p = [0.1053 0.1053 0.1053 0.0526 0.0526 0.0000 0.0526 0.1053 0.2105 0.2105] $\tilde{S}_{optimal}$ = [9.5000 9.5000 9.5000 19.0000 19.0000 Inf 19.0000 9.5000 4.7500 4.7500] s_4, s_5, s_7 overestimated for class2 and class3 remove q(2,7) (i.e. q(2.7)=0)
3	p = [0.1111 0.1111 0.1111 0.0556 0.0556 0.0000 0.0000 0.1111 0.2222 0.2222] $\tilde{S}_{optimal}$ = [9.0000 9.0000 9.0000 18.0000 18.0000 Inf Inf 9.0000 4.5000 4.5000] s_4, s_5 overestimated for class2 and class3 remove q(2,4) (i.e. q(2.4)=0)
4	p = [0.1149 0.1149 0.1149 0.0230 0.0575 0.0000 0.0000 0.1149 0.2299 0.2299] $\tilde{S}_{optimal}$ = [8.7000 8.7000 8.7000 43.5000 17.4000 Inf Inf 8.7000 4.3500 4.3500] s_5 overestimated for class2 and class3 remove q(2.5) (i.e. q(2.5)=0)
5	p = [0.1176 0.1176 0.1176 0.0235 0.0353 0.0000 0.0000 0.1176 0.2353 0.2353] $\tilde{S}_{optimal}$ = [8.5000 8.5000 8.5000 42.5000 28.3333 Inf Inf 8.5000 4.2500 4.2500]

Table 4.9 Estimation procedure employing Modified FBE Algorithm for Experiment 3.b)

Experiment 3.c)

In Experiment 3.c), eight s(i) of $\tilde{S}_{optimal}(0)$ are shown to violate the constraint $s(i)\lambda^*(1-a(k))\leq\chi$, with respect to either one or both k=2,3. Through a gradual removal of the corresponding q(i,k), as presented in Table 4.11, finally all s(i) of $\tilde{S}_{optimal}$ obtained in the fifth iteration are shown to satisfy $s(i)\lambda^*(1-a(k))\leq\chi$. Again, Table 4.10 and Figure 4.12 verify the similarity between $S_{optimal}$ and $\tilde{S}_{optimal}(5)$, while the values of BR($S_{optimal}$) and BR($\tilde{S}_{optimal}(5)$) suggest that these two broadcast scheduling schemes result in almost equally good WDB system performance.

$\mathbf{H(1)=\lambda^*(1-a(1))=0.1} \Rightarrow \chi/\lambda^*(1-a(1))\approx 5/0.1=50$

$\mathbf{H(2)=\lambda^*(1-a(2))=0.3} \Rightarrow \chi/\lambda^*(1-a(2))\approx 5/0.3=16.6667$

$\mathbf{H(3)=\lambda^*(1-a(3))=0.7} \Rightarrow \chi/\lambda^*(1-a(3))\approx 5/0.5=7.1429$

item	1	2	3	4	5	6	7	8	9	10
q (class 1)	0.04	0.05	0.1	0.02	0.03	0.0	0.0	0.0	0.13	0.04
q (class 2)	0.03	0.05	0.0	0.03	0.02	0.05	0.05	0.0	0.03	0.03
q (class 3)	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.04	0.13
l [time]	1	1	1	1	1	1	1	1	1	1
$S_{optimal}$	8.0923	7.0463	7.9244	26.5250	15.1605	192.6612	192.6612	693.7058	4.4080	3.7573
$\tilde{S}_{optimal}[4]$	11.0000	7.7000	7.7000	15.4000	15.4000	Inf	Inf	Inf	3.8500	3.8500

Table 4.10 Experiment 3.c) - Numerical MATLAB solution and closed-form approximation

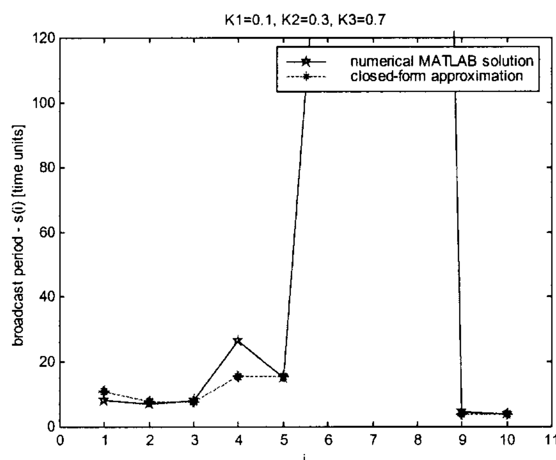


Figure 4.12 Experiment 3.c) - Numerical MATLAB solution and closed-form approximation

$BR(S_{optimal}) = 0.4336$

(number of evaluations (iterations) = 1008)

$BR(\tilde{S}_{optimal}[4]) = 0.4327$

(number of iterations = 4)

error: $$\frac{BR(S_{optimal}) - BR(\tilde{S}_{optimal}[0])}{BR(S_{optimal})} = 0.21\,[\%]$$

Iteration	Steps/Results
1	p=[0.1000 0.1000 0.1000 0.0500 0.0500 0.0500 0.0500 0.1000 0.2000 0.2000] $\tilde{S}_{optimal}$ = [10 10 10 20 20 20 20 10 5 5] s_1, s_2, s_3, s_8 overestimated for class3 s_4, s_5, s_6, s_7 overestimated for class2 and class3 remove q(3,8)
2	p=[0.1111 0.1111 0.1111 0.0556 0.0556 0.0556 0.0556 0.0000 0.2222 0.2222] $\tilde{S}_{optimal}$ = [9.0000 9.0000 9.0000 18.0000 18.0000 18.0000 18.0000 Inf 4.5000 4.5000] s_1, s_2, s_3 overestimated for class3 s_4, s_5, s_6, s_7 overestimated for class2 and class3 remove q(2,6)
3	p=[0.1176 0.1176 0.1176 0.0588 0.0588 0.0000 0.0588 0.0000 0.2353 0.2353] $\tilde{S}_{optimal}$ = [8.5000 8.5000 8.5000 17.0000 17.0000 Inf 17.0000 Inf 4.2500 4.2500] s_1, s_2, s_3 overestimated for class3 s_4, s_5, s_7 overestimated for class2 and class3 remove q(2,7)
4	p=[0.1250 0.1250 0.1250 0.0625 0.0625 0.0000 0.0000 0.0000 0.2500 0.2500 $\tilde{S}_{optimal}$ = [8 8 8 16 16 Inf Inf Inf 4 4] s_1, s_2, s_3 overestimated for class3 remove q(3,1)
5	p=[0.0909 0.1299 0.1299 0.0649 0.0649 0.0000 0.0000 0.0000 0.2597 0.2597] $\tilde{S}_{optimal}$ = [11.0000 7.7000 7.7000 15.4000 15.4000 Inf Inf Inf 3.8500 3.8500]

Table 4.11 Estimation procedure employing Modified FBE Algorithm for Experiment 3.c)

4.3 Estimation of Data-Item Request Probabilities in WDB Systems with Variable User Mobilities

In Section 2.5, of Chapter 2, we have shown that the estimation of data-item request probabilities, in a single-cell hybrid WDB system with a single class of users (i.e. with homogeneous user mobilities), is a rather challenging problem. In particular, we have proven that the given estimation has to rely not only on the statistics extracted from the unicast requests, but also on the knowledge of the current broadcast scheduling scheme and the other system and user behaviour parameters.

In this section, however, by extending on the discussion from Section 2.5, we show that the estimation of data-item request probabilities in a WDB system with variable user mobilities is a problem of progressively increased complexity. Namely, we prove that in the presence of variable user mobilities, the actual knowledge of different mobility patterns is yet another crucial factor in determining the true data-item request probabilities.

4.3.1 Invisible User Problem under Variable User Mobilities

By employing the assumption of Section 4.2, and based on the analogy with the discussion from Section 2.5, the expected number of unicast requests for item(i) (i=1,..,M), received by a hybrid WDB system with k classes of users, in an interval [0,ns(i)), can readily be obtained, as shown in (4.42).

$$\begin{aligned} &E[\mathrm{req}(0,\mathrm{ns}(i),\mathrm{item}(i))] = n \cdot E[\mathrm{req}(0,s(i),\mathrm{item}(i))] = \\ &= n \cdot P_{\mathrm{success}} \lambda \sum_{k=1}^{K} \frac{q(i,k)}{1-a(k)} \left[s(i) - \frac{1}{\lambda^{*}(1-a(k))}(1 - e^{-\lambda^{*}(1-a(k))s(i)}) \right] \end{aligned} \tag{4.42}$$

Now, similar as in Section 2.5, (4.42) clearly indicates that the number of received requests for item(i) is not a straightforward indicator of the item's actual popularity (p(i)), as this number turns out to be a non-linear function of the current broadcast scheduling scheme and the specific nature of users' mobility.

Lemma 4.2 The expected number of received unicast requests for an item depends on the actual mobility characteristics of the user population.

On average, there will be fewer received requests for those items that get mainly demanded by fast-moving users (i.e. users associated with small dwell-time), and there will be significantly more received requests for those items that get mainly demanded by slow-moving users (i.e. users associated with large dwell-time), regardless of the actual data-items' request probabilities.

Proof of Lemma 4.2

Let us assume that item(i) (i=1,..,M) gets requested by exactly one class of users, e.g. class(k) users. (Please recall, each class(k) is characterized with its own $a(k)=(1-P_{success})P_{reattempt}(k)$.) Under such an assumption (4.42) becomes

$$\begin{aligned} &E[req(0, ns(i), item(i))] = n \cdot E[req(0, s(i), item(i))] = \\ &= n \cdot P_{success} \lambda \frac{q(i,k)}{1-a(k)} \left[s(i) - \frac{1}{\lambda^{*}(1-a(k))} (1 - e^{-\lambda^{*}(1-a(i))s(i)}) \right] = \\ &= n \cdot P_{success} \lambda q(i,k) \cdot F(a(k)) \cdot G(a(k)) \end{aligned} \tag{4.43}$$

In (4.43),

$$F(a(k)) = \frac{1}{1-a(k)} \tag{4.44}$$

and

$$G(a(k)) = s(i) - \frac{1}{\lambda^*(1-a(k))}\left(1 - e^{-\lambda^*(1-a(i))s(i)}\right) \tag{4.45}$$

From (4.43), (4.46) is obtained

$$\frac{\partial E[req(0, ns(i), item(i))]}{\partial a(k)} = n \cdot P_{success} \lambda q(i,k)\left[F'(a(k))G(a(k)) + F(a(k))G'(a(k))\right] \tag{4.46}$$

where

$$F'(a(k)) = \frac{1}{(1-a(k))^2} \tag{4.47}$$

and

$$G'(a(k)) = -\frac{\lambda^*(1-a(k))(-\lambda^* s(i))e^{-\lambda^*(1-a(k))s(i)} + \lambda^*\left(1 - e^{-\lambda^*(1-a(k))s(i)}\right)}{\left(\lambda^*(1-a(k))\right)^2} = \tag{4.48}$$

$$= \frac{\lambda^*(1-a(k))s(i) + 1 - e^{\lambda^*(1-a(k))s(i)}}{\lambda^*(1-a(k))^2 e^{\lambda^*(1-a(k))s(i)}}$$

Accordingly,

$$F'(a(k))G(a(k)) + F(a(k))G'(a(k)) = \tag{4.49}$$

$$= \frac{1}{(1-a(k))^2}\frac{\lambda^*(1-a(k))s(i)e^{\lambda^*(1-a(k))s(i)} - e^{\lambda^*(1-a(k))s(i)} + 1}{\lambda^*(1-a(k))e^{\lambda^*(1-a(k))s(i)}} + \frac{1}{(1-a(k))}\frac{\lambda^*(1-a(k))s(i) + 1 - e^{\lambda^*(1-a(k))s(i)}}{\lambda^*(1-a(k))^2 e^{\lambda^*(1-a(k))s(i)}} =$$

$$= \frac{\lambda^*(1-a(k))s(i) + 2 + e^{\lambda^*(1-a(k))s(i)}\left(\lambda^*(1-a(k))s(i) - 2\right)}{\lambda^*(1-a(k))^3 e^{\lambda^*(1-a(k))s(i)}} =$$

$$= \frac{\lambda^*(1-a(k))s(i) + 2 + \left(1 + \lambda^*(1-a(k))s(i) + \frac{1}{2!}\left(\lambda^*(1-a(k))s(i)\right)^2 + \ldots\right)\left(\lambda^*(1-a(k))s(i) - 2\right)}{\lambda^*(1-a(k))^3 e^{\lambda^*(1-a(k))s(i)}} =$$

$$= \frac{\frac{1}{3!}\left(\lambda^*(1-a(k))s(i)\right)^3 + \frac{2}{4!}\left(\lambda^*(1-a(k))s(i)\right)^4 + \frac{3}{5!}\left(\lambda^*(1-a(k))s(i)\right)^5 + \ldots}{\lambda^*(1-a(k))^3 e^{\lambda^*(1-a(k))s(i)}} \geq 0$$

Since based on (49),

$$\frac{\partial E[req(0, ns(i), item(i))]}{\partial a(k)} \geq 0 \qquad (4.50)$$

we conclude that

dwell-time increases $\Rightarrow$ **$P_{reattempt}(k)/a(k)$ increases** $\Rightarrow$ $E[req(0, ns(i), item(i))]$ **increases**

and vice versa.

In other words, if an item gets mainly requested by 'slow moving' users (i.e. users associated with a large $P_{reattempt}$), then on average there will be more requests for this item received through on-demand connections, than if the item was requested by 'fast moving' users (i.e. users associated with a small $P_{reattempt}$), and the other way around.

End of Proof

4.4 Summary of Main Contributions

In this chapter, we have demonstrated that the theoretical framework presented in Chapter 2 can be easily extended so as to enable the analysis of hybrid WDB systems with variable user mobilities. Accordingly, the earlier proposed FBE algorithm can also be readily modified so as to provide a fast estimation (i.e. approximation) of $S_{otpimal}$ in such systems. Similar as in Chapter 3, the presented experimental results confirm that the modified FBE algorithm requires minimum computation, while resulting in performance almost identical to the one based on the true $S_{optimal}$.

Chapter 5

Time- vs. Frequency- Division Multiplex Wireless Data Broadcast[1]

Overview

As indicated in the preceding chapters, the broadcast scheduling scheme considered in this work is based on the concept of time-division-multiplexing (TDM). In particular, we assume that the overall broadcast bandwidth available in a single-cell (this bandwidth is annotated with B [bit·Hz][2]), is entirely allocated to a single broadcast channel, and the data-items are sent sequentially, i.e. periodically, over this channel (for an illustration see Figure 5.1).

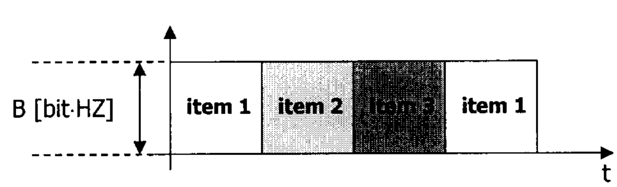


Figure 5.1 TDM-based broadcast scheduling scheme

[1] The work of this section is somewhat related to [Foltz]. However in [Foltz], a simplified scenario of a pure-push system populated with passive mobile users has been considered.

[2] Please note, bit·Hz=bit/s=bps. Hence, we can also refer to B as the overall broadcast datarate.

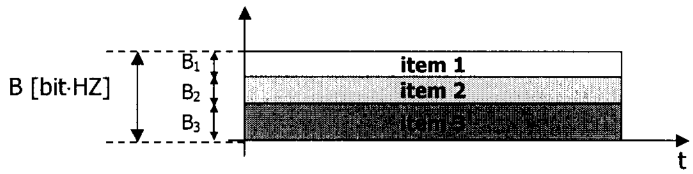


Figure 5.2 FDM-based broadcast scheduling scheme

For the completeness of our discussion, however, in this chapter we consider the possibility of frequency-division-multiplexing (FDM) WDB (Figure 5.2). In particular, we show that for any given FDM-based broadcast schedule there exists a TDM-based broadcast schedule that results in a better overall system performance. Hence, by proving the superiority of TDM- over FDM-WDB, we ultimately justify our initial decision to make TDM-WDB the main focus of this work.

We would like to emphasise, though, that the entire discussion concerning FDM-based WDB is significant primarily from a theoretical point of view. Namely:

- The idea of FDM-based WDB is built on the assumption that the available broadcast bandwidth is, i.e. can be, partitioned into a number of arbitrary-size sub-cannels. However, in most real-world systems, channel size is subject to standardization. Hence, in these systems, arbitrary bandwidth partitioning, and consequently FDM-based WDB, turn out to be practically unfeasible concepts.

- Most real-world wireless systems (e.g. mobile-terrestrial networks, satellite networks, etc.) utilize a single broadcast channel per cell. This channel is typically of predefined size and location, in the frequency spectrum, and it employs some form of TDM-based data scheduling.

5.1 FDM-Based WDB System

5.1.1 System Assumptions

In our analysis of FDM-based WDB systems we assume that both, the system and user-behaviour model preserve the characteristics as described in Section 2.2, except for the following:

(1) in the case of an FDM-based WDB system, the overall broadcast bandwidth is divided into a number of arbitrary-size sub-channels (see Figure 5.2);

(2) each data-item(i) is assigned and repeatedly broadcasted on its own sub-channel(i), i=1,..,M (see Figure 5.3);

(3) the users interested in retrieving item(i) (=1,..,M) find, i.e. tune into, the right broadcast sub-channel through either of the following two mechanisms:

(3.1) the system selects one of the sub-channels to serve as a 'bulletin board' (i.e. this sub-channel repeatedly broadcasts the information about the actual location and content of all the other sub-channels); hence, by tuning into the 'bulletin board' sub-channel, the users quickly learn about the actual location of sub-channel(i)

(3.2) the users search through the sub-channels, either in a random or in an orderly fashion, until they find sub-channel(i)

(4) a user can start downloading an item only from its starting point, regardless of the actual time when the user first tunes into the corresponding sub-channel (see Figure 5.3).

Please note, we employ assumption (4) mainly due to the possibility that some data-items contain time-sensitive information. Namely, subsequent copies of such a data-item could have slightly or significantly different content, byte/packet size or ordering, etc. Hence, a proper data-item reassembly would not be possible if one tried to combine bytes/packets from two or more of the data-item's subsequent copies. In addition, any assumption other than (4) would not result in a fair comparison of FDM- vs. TDM- based WDB systems, as studied in the preceding sections.

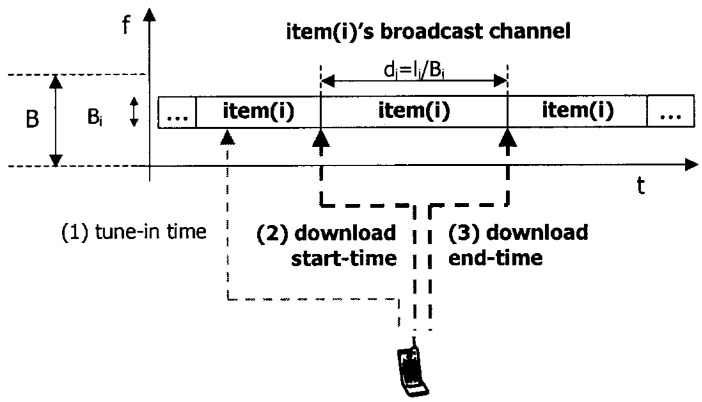


Figure 5.3 Broadcast data-item retrieval in FDM-based WDB system

5.1.2 Broadcast Service Ratio in FDM-WDB System (BR_{FDM})

Let us annotate the broadcast bandwidth assigned to sub-channel(i) of an FDM-based WDB system with B(i) [bit·Hz] (see Figure 5.2 and 5.3), and let us assume that the actual value of B(i) (i=1,..,M) is subject to optimization. That is, let us assume that the system employs a *broadcast bandwidth allocation scheme*[3] $\mathcal{B}$=(B(1), B(2), .., B(M)), that can dynamically adjust (i.e. the bandwidth assigned to each of the sub-channels can dynamically change), so as satisfy a given objective. Here, similar as in the case of TDM-based WDB, we will consider the maximization of the system's broadcast service ratio (BR_{FDM}) to be such an objective.

In order to find the actual expression for BR_{FDM}, first let us recall that the overall broadcast bandwidth available in a single-cell FDM-WDB system is B [bit·Hz]. Hence, from the assumptions of the above paragraph, the constraint (5.1) implies. (Figure 5.2 provides an illustration for (5.1).)

[3] A particular broadcast bandwidth allocation scheme is the actual partitioning of the available broadcast bandwidth into a fixed number (M) of broadcast sub-channels.

$$B = \sum_{i=1}^{M} B(i) \qquad (5.1)$$

Next, by recalling that $l(i)$ [bits] represents the size of item(i)[4] (Section 2.2), the time required for a single transmission of item(i) on the broadcast sub-channel(i) (as shown in Figure 5.3), can be found from (5.2).

$$d(i) = \frac{l(i)}{B(i)} \qquad (5.2)$$

(Please note that by employing (5.2) back in (5.1), an alternative form of the FDM-WDB system constraint (5.1) is obtained, as given in (5.3).)

$$1 - \sum_{i=1}^{M} \frac{l(i)/B}{d(i)} = 0 \qquad (5.3)$$

Now, based on the analogy with the TDM-WDB case discussed in Section 2.3.2, it can be easily verified that the expected number of users interested in retrieving item(i), who arrive in the system in between two consecutive broadcastings of this item on sub-channel(i), i.e. arrive in an interval $[(k-2)\cdot d(i),\ (k-1)\cdot d(i))$, $k \in N$ (see Figure 5.4), and eventually get served through broadcasting at time $(k-1)d(i)$ (we annotate this value with $n_{FDM}((k-2)d(i),(k-1)d(i),\text{item}(i),(k-1)d(i))$, corresponds to (5.4).

$$E_{FDM}[n_{FDM}((k-2)\cdot d(i),(k-1)\cdot d(i),\text{item}(i),(k-1)\cdot d(i))] = \frac{\lambda p(i)}{\lambda^{*}(1-a)}(1 - e^{-\lambda^{*}(1-a)d(i)}) \qquad (5.4)$$

Please note, for the simplicity of our discussion, here we assume that the bandwidth resources and/or time required to inform the newly arrived users about the actual location of sub-channel(i) (i=1,..,M) are negligible. Nevertheless, in Section 5.2.2, we relax this assumption, and discuss

[4] Here, for the sake of consistency with the preceding analysis of TDM-based WDB systems, we will assume that the length (i.e. byte/packet size) of each data-item does not change in time.

the possible drawbacks of FDM-WDB systems caused by unknown sub-channel size and location.

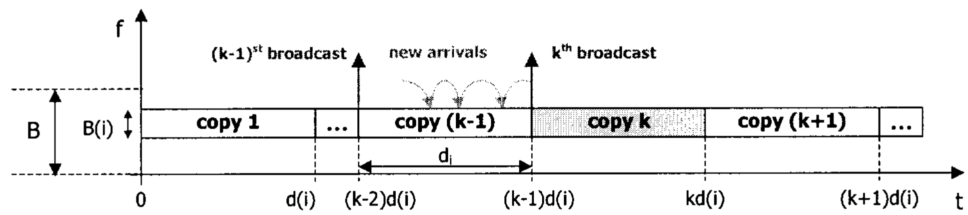


Figure 5.4 Broadcast service in FDM-WDB system – time perspective

Based on (5.4), the expected number of users who obtain item(i) from the broadcast sub-channel(i), within an entire broadcast schedule (C), can be obtained simply as (5.5).

$$E_{FDM}[n_{FDM}(0, C, \text{item}(i), k \cdot d(i) \mid k = 1,.., m(i))] = \frac{C}{d(i)} \frac{\lambda p(i)}{\lambda^{*}(1-a)}(1 - e^{-\lambda^{*}(1-a)d(i)}) \tag{5.5}$$

(Here, similar as in Assumption 2.1 (page 33), C is assumed to satisfy the following: for every $d_i \in R^+$ there exists $m(i) \in N^+$ such that $C = m(i) \cdot d(i)$ $(i=1,..,M)$. For an illustration, see Figure 5.5.)

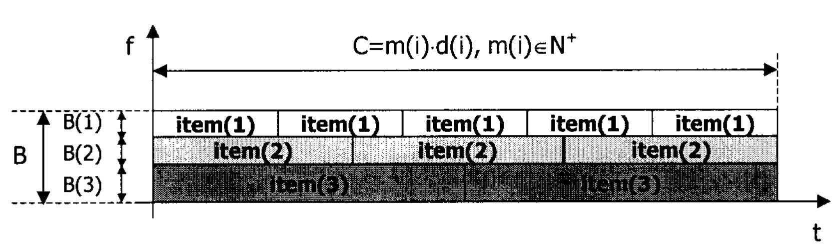


Figure 5.5 Entire broadcast schedule in FDM-WDB system

Consequently, the expected number of users served through either of the broadcast sub-channels, i.e. served with either of the M items, within C, corresponds to (5.6).

$$E[n_{FDM}(0,C,item(i),k\cdot d(i)\,|\,k=1,..,m(i);i=1,..,M)] = \tag{5.6}$$
$$=\sum_{i=1}^{M} E[n_{FDM}(0,C,item(i),k\cdot d(i)\,|\,k=1,..,m(i))] =$$
$$=\frac{C\lambda}{\lambda^*(1-a)}\sum_{i=1}^{M}\frac{p(i)}{d(i)}(1-e^{-\lambda^*(1-a)d(i)})$$

Finally, considering that $C\lambda$ users are expected to arrive in the system within the entire C (Section 2.2.2), and $E[n_{FDM}(0,C)]$ of them are expected to be served through either of the broadcast sub-channels, then the system's broadcast service ratio is simply (5.7).

$$BR_{FDM} = \frac{E[n_{FDM}(0,C,item(i),k\cdot d(i)\,|\,k=1,..,m(i);i=1,..,M)]}{C\lambda} = \tag{5.7}$$
$$=\frac{1}{\lambda^*(1-a)}\sum_{i=1}^{M}\frac{p(i)}{d(i)}(1-e^{-\lambda^*(1-a)d(i)})$$

5.1.3 Optimal Broadcast Schedule in FDM-WDB System

Once the expression for BR_{FDM} is known (see (5.7)), the optimal broadcast schedule of a single-cell FDM-based WDB system can be defined as follows:

Definition 5.1 The optimal broadcast schedule of a single-cell FDM-based WDB system ($D_{optimal}$=(d(1)*,d(2)*,.., d(M)*)) corresponds to the point in R^M which maximizes BR_{FDM}. That is,

$$D_{optimal} = \{D^* \in R^M \,|\, BR_{FDM}(D^*) = \max BR_{FDM}(D),\ \text{where}\ D=(d(1),d(2),..,d(M)) \in R^M\}$$

Here, it is important to observe that although we use the same term 'broadcast schedule', both in the case of TDM-based and FDM-based WDB systems, these two have considerably different physical meaning. Namely, in the case of TDM broadcast schedule S=(s(1),s(2),..,s(M)), each element (s(i)) corresponds to the broadcast period associated with one of the data-items. However, in the case of FDM broadcast schedule D=(d(1),d(2),..,d(M)), each d(i) represents the broadcast, i.e. transmission, time of item(i) on sub-channel(i), i=1,..,M (see Figure 5.3).

Based on the discussion of the preceding section, and (5.2) in particular, any given FDM broadcast schedule D=(d(1),d(2),..,d(M)) corresponds to a single FDM broadcast bandwidth allocation scheme $\mathcal{B}$=(B(1),B(2),..,B(M)). Hence, once the optimal FDM broadcast schedule, in the sense of Definition 5.1, is known, the corresponding, and thereby optimal, FDM broadcast bandwidth allocation scheme can be obtained simply as

$$\mathcal{B}_{\text{optimal}} = \{(B(1)^*, B(2)^*,.., B(M)^*) \in R^M \;\Big|\; B(i)^* = \frac{l(i)}{d(i)^*},\ i = 1,2..,M\} \tag{5.8}$$

(Please note, from a practical point of view, (5.8) implies that the problem of determining what bandwidth should be allocated to sub-channel(i) ($B(i)^*$), so as to maximize the system's BR_{FDM}, in fact translates into the problem of finding the optimal broadcast time of item(i) ($d(i)^*$), i=1,..,M.)

Now, similar as in the case of $S_{optimal}$ (see Section 2.4.2), the search for the actual $D_{optimal}$, in a given FDM-based WDB system, goes beyond a simple procedure of finding the maximization point for BR_{FDM}. Namely, due to the physical constraint (5.3), the search for $D_{optimal}$ turns out to be a constrained single-objective optimization problem, as stated in (5.9).

$$\text{maximize} \qquad BR_{FDM}(d(1), d(2),.., d(M)) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{d(i)} (1 - e^{-\lambda^*(1-a)d(i)}) \tag{5.9.a}$$

$$\text{subject to} \qquad C_{FDM}(d(1), d(2),.., d(M)) = 1 - \sum_{i=1}^{M} \frac{l(i)}{d(i)} = 0 \tag{5.9.b}$$

(Note, l(i)=*l*(i)/B [time units], in (5.9.b). In practical terms, l(i)= *l*(i)/B represents the time that would be required for the broadcasting of item(i), provided all the available broadcast bandwidth (B) was allocated to sub-channel(i). In other words, l(i) corresponds to the broadcasting time of item(i), as it would be found in an equivalent[5] single-cell TDM-based WDB system (see Section 2.4.2).))

Lemma 5.1 Let the time and bandwidth resources required to inform the users about the location of sub-channel(i) (i=1,..,M), in an FDM-based WDB system, be negligible. Then, the numerical value of the optimal broadcast schedule in the given system ($D_{optimal}$) is identical to the numerical value of the optimal broadcast schedule in an equivalent TDM-based WDB system ($S_{optimal}$). That is,

$$\mathbf{D_{optimal}=S_{optimal}}$$

Proof of Lemma 5.1

Let us recall from Theorem 2.1 (Section 2.4.2) that the problem of finding the optimal broadcast schedule in an equivalent TDM-based WDB system[6] ($S_{optimal}=(s(1)^*,s(2)^*,..,s(M)^*)$) corresponds to

maximize $$BR_{TDM}(s(1),s(2),..,s(M)) = \frac{1}{\lambda^*(1-a)}\sum_{i=1}^{M}\frac{p(i)}{s(i)}(1-e^{-\lambda^*(1-a)s(i)}) \quad (5.10.a)$$

subject to $$C_{TDM}(s(1),s(2),..,s(M)) = 1-\sum_{i=1}^{M}\frac{l(i)}{s(i)}=0 \quad (5.10.b)$$

[5] By 'equivalent' we mean a TDM-WDB system with the same overall broadcast bandwidth (B) available.
[6] Please note, here we use BR_{TDM} instead of the earlier BR, so as to make clear that this value corresponds to TDM-WDB systems.

A simple comparison of (5.9) and (5.10) reveals that, mathematically, the two problems are, in fact, identical. Therefore, we can claim that the numerical values of their respective solutions ($D_{optimal}$ and $S_{optimal}$) must be equal. (Please recall, in Theorem 2.2, the uniqueness of the solution to (5.9), i.e. (5.10), is proven. However, in Theorem 2.3 it is shown that no closed from expression for such a solution exists.)

End of Proof

5.2 FDM- vs. TDM- WDB System Performance

5.2.1 Idealized FDM-WDB System Assumptions

In the preceding sections, we have considered an idealized single-cell FDM-WDB system scenario, in which mobile users interested in retrieving item(i) (i=1,..,M) get instantaneously informed about the location of sub-channel(i), once they enter the system. Also, we have proven that under such conditions, the problem of finding the optimal FDM broadcast schedule ($D_{optimal}$) turns out to be mathematically identical to the problem of finding the optimal broadcast schedule in an equivalent TDM-WDB system ($S_{optimal}$), and thereby $D_{optimal}=S_{optimal}$. Consequently,

$$BR_{FDM}(D_{optimal}) = BR_{TDM}(S_{optimal}) \tag{5.11}$$

Hence, it is to expect that the performances of the two systems (FDM- and TDM- based), with respect to the overall broadcast service ratio, is the same. Still, in terms of the overall service time (ST), the performance of the TDM-WDB system appears to be considerably superior, as illustrated in Figure 5.6 and stated in (5.12).

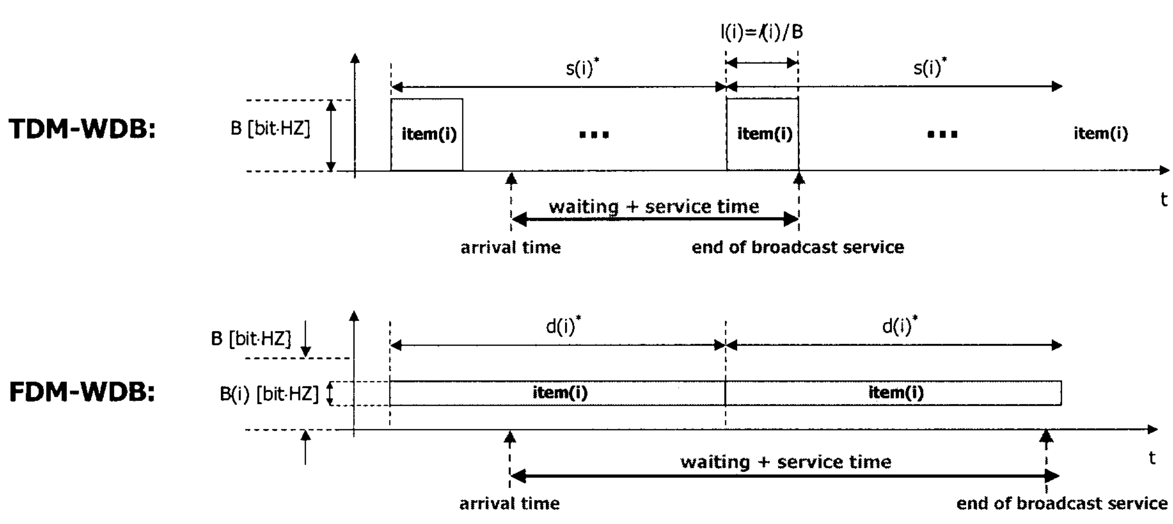


Figure 5.6 Overall broadcast service time in TDM-based and FDM-based WDB system

$$ST_{TDM}(S_{optimal}) < ST_{FDM}(D_{optimal}) \tag{5.12}$$

Namely, since $D_{optimal}=S_{optimal}$, i.e. $d(i)^*=s(i)^*$ ($i=1,..,M$), a randomly arrived user, interested in retrieving item(i), on average would have to wait the same amount of time until the beginning of the next item(i)'s broadcasting, in either of the two systems. However, once the broadcasting of item(i) begins, the entire transmission of the given item would take only $l(i)=l(i)/B<s(i)^*$ time units in the TDM-WDB system, while in the FDM-WDB system it would take the full $d(i)^*=s(i)^*$ time units (see Figure 5.6).

5.2.2 Realistic FDM-WDB System Assumptions

As indicated earlier, the assumption of negligible 'sub-channel tune-in time and/or resources', in FDM-WDB systems, may not hold in reality, due to the following reasons:

- Users that join the system for the first time may not be aware of the overall number of broadcast sub-channels, their individual location in the spectrum, size, and content. Thus, as stated in assumption (3) of Section 5.1.1, it would take some time for these users (i.e. their end devices) to perform the search and find a particular sub-channel. Optionally, though, the system could facilitate, i.e. speed up, the users' search by dedicating a portion of the overall broadcast bandwidth to the so-called 'bulletin board' sub-channel.

- As the system parameters change, the values of d(i) ($i=1,..,M$) would have to change accordingly, so as to maintain BR_{FDM} at its maximum (see (5.9)). As a consequence, the position and size of individual sub-channels ($B(i)=l(i)/d(i)$, $i=1,..,M$ – see (5.2)) would not remain the same. Therefore, even the users that have been previously served through broadcasting might still require the assistance of the 'bulletin board' sub-channel, in order to quickly learn about the latest broadcast bandwidth allocation scheme employed.

From the above examples, it is clear that the presence of a bulletin-board sub-channel, in an FDM-WDB system, could substantially reduce the users' tune-in time and, accordingly, improve the overall waiting time. Note, however, even for tune-in time → 0, the discussion from section 5.2.1, concerning the superiority of TDM-WDT over FDM-WDB with respect to service time (ST), still applies. Nevertheless, while providing improvement in the tune-in time, the utilization of the bulletin board sub-channel could have some drawbacks, including a lower upper bound on the system's broadcast service ratio (BR_{FDM}).

Lemma 5.2 Let B [bit·Hz] be the overall broadcast bandwidth available in a FDM-WDB system, and let B_{bb} [bit·Hz] ($0<B_{bb}<B$) be the fraction of B allocated to the bulletin board sub-channel. Then, the maximum broadcast service ratio of such a system (max BR_{FDM_bb}) is always less than the maximum broadcast service ratio of an equivalent FDM-WDB system with no bulleting board sub-channel (max BR_{FDM}). Consequently, max BR_{FDM_bb} is always less than the maximum broadcast service ratio of an equivalent TDM-WDB system (max BR_{TDM}). That is

$$\max BR_{FDM\_bb} < \max BR_{FDM} = \max BR_{TDM}$$

Proof of Lemma 5.2

Due to the utilization of the bulletin board sub-channel, the effective broadcast bandwidth of the FDM-WDB system gets reduced to

$$B_{effective} = B - B_{bb} \tag{5.13}$$

Hence, from (5.13) and based on the analogy with the idealized FDM-WDB system scenario studied in Section 5.1.3, the optimal broadcast schedule of the given system can be obtained as a solution to the following optimization problem.

maximize $$BR_{FDM\_bb}(d(1),d(2),..,d(M)) = \frac{1}{\lambda^*(1-a)}\sum_{i=1}^{M}\frac{p(i)}{d(i)}(1-e^{-\lambda^*(1-a)d(i)}) \quad (5.14.a)$$

subject to $$C_{FDM\_bb}(d(1),d(2),..,d(M)) = 1-\sum_{i=1}^{M}\frac{l(i)/B_{effective}}{d(i)} = 0 \quad (5.14.b)$$

(Please observe, the objective function in (5.14) remains the same as in (5.9). That is,

$$BR_{FDM\_bb}(d(1),d(2),..,d(M)) = BR_{FDM}(d(1),d(2),..,d(M)) \quad (5.15)$$

However, $B_{effective}$ in (5.14.b) replaces the earlier B in (5.9.b).)

Through a simple rearrangement of (5.14.b) (see (5.16)), and by annotating $B_{effective}/B$ with D, (5.17) and, subsequently, (5.18) are obtained. (Note, $B_{effective}>0$, $B>0$, $B_{effective}<B$ imply $0<D<1$.)

$$C_{FDM\_bb}(d(1),d(2),..,d(M)) = \frac{B}{B_{effective}}\left(\frac{B_{effective}}{B}-\sum_{i=1}^{M}\frac{l(i)/B}{d(i)}\right) = 0 \quad (5.16)$$

$$C_{FDM\_bb}(d(1),d(2),..,d(M)) = \frac{1}{D}\left(D-\sum_{i=1}^{M}\frac{l(i)}{d(i)}\right) = 0 \quad (5.17)$$

$$C_{FDM\_bb}(d(1),d(2),..,d(M)) = \frac{1}{D}\left(D - 1 + \left[1-\sum_{i=1}^{M}\frac{l(i)}{d(i)}\right]\right) = 0 \quad (5.18)$$

Since $0<D<1$, it is clear that (5.18), and consequently (5.14.b), will hold if and only if (5.19) holds. Accordingly, (5.19) could server as an alternative to (5.14.b).

$$1-\sum_{i=1}^{M}\frac{l(i)}{d(i)} = 1-D \quad (5.19)$$

Thus, based on (5.19), (5.14) can be transformed to a mathematically equivalent problem, as shown in (5.20). (Note, in the proceeding discussion, we will annotate the solution to (5.20), i.e. (5.14), with $D_{optimal\_bb}$.)

maximize $$BR_{FDM\_bb}(d(1), d(2),.., d(M)) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \frac{p(i)}{d(i)} (1 - e^{-\lambda^*(1-a)d(i)}) \qquad (5.20.a)$$

subject to $$1 - \sum_{i=1}^{M} \frac{l(i)}{d(i)} = 1 - D \qquad (5.20.b)$$

A simple comparison of (5.20) and (5.9) shows that the main difference between the problems of finding the optimal broadcast schedule in an FDM-WDB system with the bulletin board sub-channel ($D_{optimal\_bb}$) and finding the optimal broadcast schedule in an idealized FDM-WDB system, i.e. system with no bulletin board sub-channel (D_{optiml}), lies in the nature of their respective constraints – (5.20.b) and (5.9.b). Namely, while in (5.9.b) it is required that $D_{optimal}$ satisfies

$$1 - \sum_{i=1}^{M} \frac{l(i)}{d(i)} = 0 \qquad (5.21)$$

here in (5.20.b) (due to $0<D<1$) it is required that $D_{optimal\_bb}$ satisfies

$$1 - \sum_{i=1}^{M} \frac{l(i)}{d(i)} > 0 \qquad (5.22)$$

In other words, while $D_{optimal}$ should lie on the boundary of (5.23), $D_{optimal\_bb}$ should lie in its interior.

$$1 - \sum_{i=1}^{M} \frac{l(i)}{d(i)} \geq 0 \qquad (5.23)$$

Now, from this, based on Theorem 2.1, and by recalling that $BR_{FDM}(D)=BR_{FDM\_bb}(D)$ (see (5.15)), we obtain

$$BR_{FDM\_bb}(D_{optimal\_bb}) < BR_{FDM}(D_{optimal}) \tag{5.24}$$

Hence, (5.24) clearly indicates that the maximum achievable BR in an FDM-WDB system with a bulletin board sub-channel will always be less than the maximum achievable BR in an equivalent FDM-WDB system with no bulletin board sub-channel present. Consequently, based on Lemma 5.1, this implies that max $BR_{FDM\_bb}(D)$ will also always be less then the maximum achievable BR in an equivalent TDM-WDB system, as indicated in (5.25).

$$BR_{FDM\_bb}(D_{optimal\_bb}) < BR_{TDM}(S_{optimal}) \tag{5.25}$$

End of Proof

5.3 Summary of Main Contributions

In this chapter we have introduced the concept of frequency division multiplexing WDB, and we have studied two particular types of FDM-WDB systems:

(1) idealized – with negligible time that mobile users spend searching for a particular broadcast sub-channel (the-so called tune-in time);

(2) realistic – with a bulletin board sub-channel, which is intended to minimize the actual users' tune-in time.

We have proven that the performance of both FDM-WDB system types is inferior compared to the performance of WDB systems that employ time division multiplex (TDM) scheduling on the broadcast channel (i.e. TDM-WDB systems). In particular, we have shown that TDM-WDB systems outperform idealized FDM-WDB systems in terms of service time (ST), while TDM-WDB systems outperform realistic FDM-WDB systems both in terms of service time (ST) and broadcast service ratio (BR).

Chapter 6

Multicell Hierarchical Wireless Data Broadcast Systems

Overview

The content of this chapter is organized as follows:

Section 6.1 First, we introduce the concept of 'hierarchical cell structure' (HCS) and discuss the numerous advantages associated with the utilization of HCS as a design model for multi-cellular wireless/mobile networks.

Section 6.2 Next, by extending on the ideas from Chapter 2 and employing the main principles of HCS, we propose a model of hierarchically organized multi-cell WDB (HWDB) systems with user retrials.

Section 6.3 In Section 6.3, we show that the search for the optimal broadcast schedule ($S_{optimal}$) in an HWDB system exceeds the complexity of the corresponding problem as found in an equivalent single-layer WDB system. However, by exploiting our earlier obtained approximate closed-form expression for single-cell $S_{optimal}$ (see Chapter 3), we show that the given problem can be considerably simplified. Subsequently, we propose an algorithm for a fast approximation of $S_{otpimal}$ in HWDB systems. Through a series of test-cases, we prove the efficiency of the given algorithm.

Section 6.4 Finally, in Section 6.3, we conduct a comparative theoretical analysis of single-layer WDB v.s HWDB systems. The results of this analysis show that the higher operational cost of HWDB systems, compared to the cost of their single-layer counterparts, is well justifiable, as HWDB systems ultimately provide considerably superior performance in terms of the overall throughput, QoS, and GoS.

6.1 Hierarchical Cellular Structure (HCS)

Among the many challenges faced by today's wireless mobile data networks, the following three seem to be the most significant ones: 1) efficient utilization of scarce radio spectrum so as to provide coverage in sufficiently large areas, 2) maximization of system capacity, 3) provision of equal QoS and GoS to users of different mobility characteristics. Recently, *hierarchical cellular structure*[1] (HCS) has been proposed as a design model for wireless/mobile networks. Numerous studies have shown that HCS-based networks have the capacity to simultaneously achieve all of the above-mentioned goals, thus providing performance considerably superior to the performance of single-tier cellular systems.

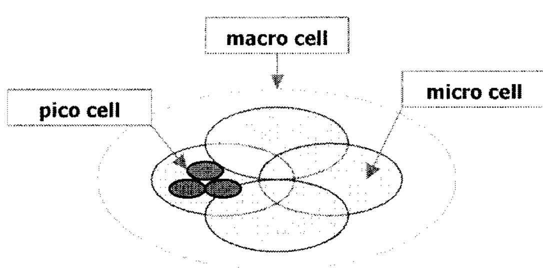


Figure 6.1 Hierarchical cellular structure

Hierarchical cellular structure (see Figure 6.1) consists of two or three layers of overlaid cells, with the lowest layer in the hierarchy corresponding to the physically smallest (pico) cells. The major advantage of HCS over a *single-tier structure* lies in the overlapping arrangement of pico/micro/macro cells. Such arrangement allows that traffic loads are shared among the tiers, so as to improve the overall system performance.

In general, small cells of HCS are used to

[1] Another commonly used name for HCS is multi-tier cellular structure.

- obtain greater spectral reuse;
- provide coverage to those points that larger cells cannot reach due to natural or man-made obstacles;
- allow the use of low-power lightweight handheld mobile devices.

On the other hand, large cells are used to

- provide additional (overflow) channels for clusters of small cells when heavily loaded;
- cover larger areas with low-cost implementation;
- cover spots that are difficult for radio propagation in small cells;
- support fast moving users (e.g. vehicles on highways) while requiring fewer handoffs.

(For a detailed overview of HCS see [*Ortigoza-Guerrero*].)

6.1.1 HCS in Cellular Mobile (PCS) Systems

Previous research works regarding HCS have mainly focused on frequency/channel allocation and mobility handling in PCS networks, i.e. cellular mobile systems. Here, we review a few related problems, together with the corresponding solutions proposed.

1) In HCS PCS systems, new/handoff calls, i.e. users, are directed to the appropriate tier based on their speed. (Typically, lower tiers accommodate low-mobility users, while upper tiers accommodate high-mobility users.) In case of congestion, though, if there are no available channels on the preferred tier, the call will be directed to the other (un-preferred) tier. This phenomenon is called *call-overflow*. However, a problem arises if there are no available channels in the corresponding micro- and macro- cells of both, the preferred and un-preferred tier. In [*LinKJ*], a new channel sharing strategy, the so-called vertical load-sharing, is proposed. According to this strategy, when a channel request arrives to a microcell, for example, if both the micro- and corresponding macro- cell have no channels available, the

macrocell forces one of its calls to one of the other (n-1) microcells, in order to vacate a channel. The presented results indicate considerably better performance of the given strategy, over the existing channel-sharing schemes, with respect to *call blocking*[2] and *call dropping*[3] probabilities.

2) In [*Mihailescu*], a performance comparison of non-reversible and reversible HCS PCS systems is presented. (In non-reversible HCS systems, MUs once directed towards the macrocell level do not return to the microcell level. In reversible systems, on the other hand, MUs can be re-directed from macro- to micro- cell level either upon entering a new microcell, or as soon as a resource is released in the associated microcell.) The presented results suggest that the reversible HCS systems provide better performance in terms of offered traffic and new call blocking probability, particularly for high-mobility users.

3) In [*Han*] the problem of channel allocation for multicast service in HCS (i.e. two-tier) cellular systems, consisting of N micro- and a single macro- cell, is considered. The study and results presented in the paper show that if the overall number of channels is limited, then a special method should be employed when deciding which of the two layers is more appropriate for the support of a given multicast group. In particular, the system performance with respect to channel efficiency and blocking probability can be significantly improved provided multicast traffic associated with microcell groups[4] of large sizes gets accommodated by the microcell layer, and vice versa.

4) According to [*Ekici*], there is a certain cost associated with the utilization of every micro, i.e. macro- cell channel in an HCS network. Consequently, an HCS network design is considered optimal if it minimizes the overall system cost, while preserving call blocking and call dropping probabilities below certain level. Hence, the optimal HCS network design problem turns out to be a constrained minimization problem. In [*Ekici*], Simulated Annealing (SA) is

[2] A new call is blocked if it cannot be served due to the lack of available (free) channels [*Ekici*].

[3] When a handoff call cannot find a free channel in the new cell, the call terminates forcefully. This situation is called call dropping [*Ekici*].

[4] According to [*Han*], microcell group size represents the number of microcells in which users belonging to a specific multicast group exist.

proposed as the most appropriate algorithm for solving the given type of constrained minimization problems, and SA's superiority over some other numerical estimation techniques is demonstrated.

6.1.2 HCS in Satellite Systems

The main ideas related to HCS are applicable to satellite systems, as well. For example, in [*Ercetin*], the satellite-terrestrial system is envisioned as a two-tier hierarchy, and the issue of broadcast data delivery in such a system is discussed. In particular, according to the concept proposed in [*Ercetin*], the main server in the upper tier of the hierarchy forwards the information items, via satellite, to the geographically distributed terrestrial base stations. In the lower tier, each terrestrial base station (BS) has the capacity to store only some of the items transmitted. Thus, at every satellite transmission the terrestrial base stations must decide whether to keep the transmitted item or not. Based on the experimental results presented in [*Ercetin*], the lowest possible average latency observed by the users is obtained provided the terrestrial BSs' cache management intelligently employs the knowledge of the satellite broadcast scheduling scheme.

6.1.3 Hierarchical Service Discovery Service in Networks of Pervasive Devices

According to our knowledge, no published research work has explored the idea of a direct implementation of HCS in networks of pervasive devices. However, in several research papers the utilization of another form of hierarchy (i.e. *chain-like hierarchy*) for the purpose of optimized *service discovery service* (SDS) in networks of pervasive devices has been proposed (see [*Czerwinski*] and [*Nidd*]).

SDS in networks of pervasive devices has a twofold function [*Gribble*]. On one side, it provides a mechanism by which services can announce their presence in the infrastructure. On the other hand, it provides a mechanism by which both human users and programs can locate these announced services across the wide area. (In both mechanisms, SDS servers play a key role, i.e. they are the ones who solicit information from the services and then use it to fulfill client queries.) As mentioned previously, networks of pervasive devices are generally envisioned as highly dynamic systems, with thousand of users simultaneously attempting to obtain service, using low-cost, low-power end devices [*Gupta*]. Hence, in such systems, it is crucial that SDS servers are able to automatically discover other running SDS servers, negotiate a hierarchy to chain themselves together, and propagate service announcements / user queries across the hierarchy, so as to achieve the optimal load balancing [*Gribble*].

The benefits of organizing a group of SDS servers into a *chain-like hierarchy* are undoubted. However, we believe that the main concepts of HCS, i.e. *pyramidal-like hierarchy*, are applicable to SDS infrastructure as well. Accordingly, the actual implementation of HCS for the purpose of SDS in networks of pervasive devices could provide improvements similar to the ones obtained in case of PCS networks (see Section 3.1.1).

6.1.4 Wireless Data Broadcast in HCS Systems

The previous research works on wireless data broadcast have almost exclusively focused on single-cell, i.e. single-tier, wireless systems[5] (see Chapter 1). On the other hand, as indicated in Section 3.1.2, HCS has already become a preferred form of cell organization in wireless networks, due to its numerous advantages over the single-tier cellular structure. Thus, the problem of optimized WDB in a multicell HCS-based system appears to be rather important, however still unexplored research area.

[5] According to our knowledge, [*Ercetin*] is the only published research work that deals with WDB in a hierarchically organized wireless system.

In the remainder of this chapter, we give our contribution to the subject of WDB in multicell HCS-based environments. In particular, by extending on the discussion from the preceding chapter, first we propose a model of a hybrid multicell HCS-WDB system. Consequently, we consider the problem of optimal broadcast scheduling scheme in such a system, and compare its performance against a single-tier structure composed of an equal number of independent generalized WDB cells.

6.2 Model of Hybrid Hierarchical WDB System

Our model of hierarchical wireless data broadcast (HWDB) systems is built upon the following assumptions:

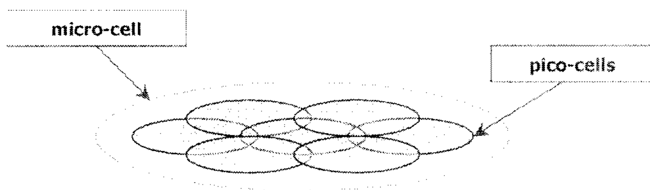


Figure 6.2 Two-layer hybrid HWDB system

The system consists of an arbitrary number of pico-cells overlaid by a single micro-cell, as illustrated in Figure 6.2. The pico-cells are used to ensure the capacity over smaller areas and to serve as 'entry points' in the system[6]. The micro-cell is used to provide additional coverage. A single broadcast and a number of unicast channels are assigned to each cell, at both levels of the hierarchy.

The set of data items available to mobile users through the broadcast and unicast channels is finite {item(i)|i=1,..,M}. The items are categorized in three groups, with respect to the current broadcasting policy:

1) items that get broadcasted at the micro-cell level only;
2) items that get broadcasted at the pico-cell level only;
3) items that do not get broadcasted at all, due to their low request probabilities.

(Note, we assume that the broadcasting of an item at both levels of the hierarchy is somewhat redundant. Thus, an item can appear on the broadcast channel of either pico- or micro- cell only.)

[6] A MU can get accepted in the system and request unicast service only by communicating with the nearest pico-cell's base station (i.e. pico-cell access point). In other words, a MU does not access the micro-cell resources directly. However, once accepted for service through the pico-cell level, the MU can be re-assigned to one of the micro-cell unicast channel.

The behaviour of individual users follows the model presented in Figure 6.3. This model extends upon the realistic-impatient MU behaviour model that has been presented in Chapter 2.

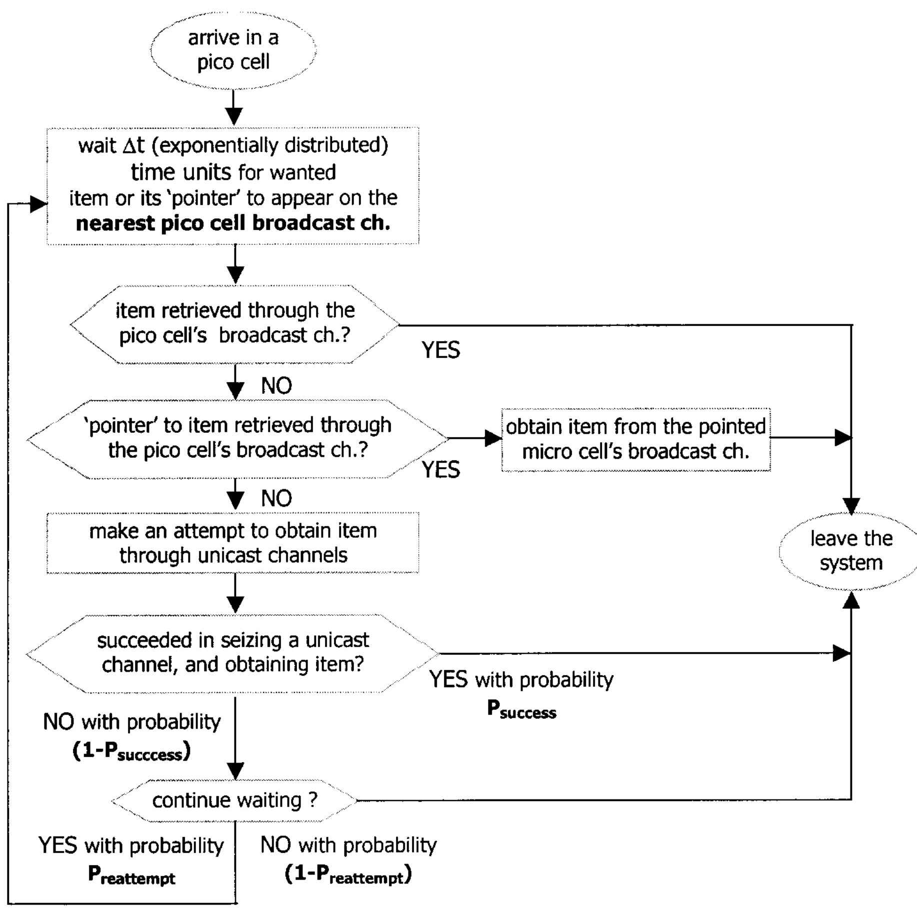


Figure 6.3 Realistic-impatient MU behaviour model in HWDB system

According to the model in Figure 6.3, every MU who appears in the system wants to retrieve a single item, and as soon as the item is obtained the MU leaves. Upon arrival in the system, MUs first tune into the broadcast channel of the nearest pico-cell, hoping to retrieve the desired item or the corresponding ***data pointer*** from there. Data pointers are used to inform MUs about the items that can be retrieved from the micro-cell broadcast channel. In particular, a data pointer contains the information regarding the exact location (frequency/slot) where the corresponding item could be found.[7] (The illustrations concerning the two possible data retrieval scenarios are provided in Figure 6.4 and 6.5.) However, as there is a chance that the given item is not included in the current broadcast schedule of either level in the hierarchy, the MU occasionally attempts to seize one of the point-to-point channels, and retrieve the desired information on-demand. These attempts are made in intervals of Δt seconds, where Δt is an exponentially distributed random variable with the mean $1/\lambda^*$. On each attempt the MU acquires a point-to-point connection with probability $P_{success}$. On each failed attempt the MU gets impatient and leaves the system with probability $(1\text{-}P_{reattempt})$.

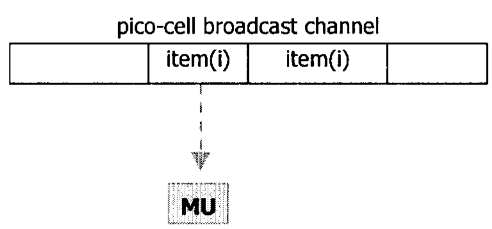
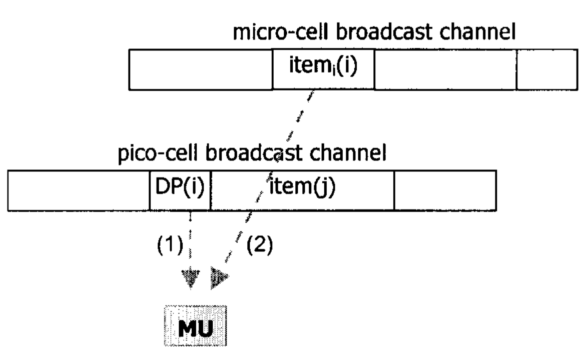


Figure 6.4 Data retrieval through pico-cell broadcast channel



Retrieval procedure:

(1) MU captures 'data pointer' from the lower level broadcast channel. ('Data pointer' contain information regarding the physical/logical location (i.e. frequency/slot) of the higher level broadcast channel where the actual data (i.e. item) can be found

(2) MU tunes into the pointed broadcast channel of the higher level cell, and retrieves the requested item from there

NOTE: dp(i)<<I(i)

Figure 6.5 Data retrieval through micro-cell broadcast channel

A MU of pico-cell(j) requests item(i) with probability p(i,j) (i=1,..M, j=1,..,N), where $\sum\sum p(i,j)=1$ at any point in time. Thus, if the MU arrivals in the system follow a Poisson process with rate λ,

[7] From the queueing theory perspective, the hybrid HWDB systems belong to the group of *referral queueing systems* [*Marianov*]. In these systems, users cannot obtain service from a higher-level server, unless a low-level server refers them to it.

then based on the splitting property [*Banks*], the arrival of MUs requesting item(i) in pico-cell(j) is also a Poisson process with rate λ(i,j) (i=1,..M, j=1,..,N). Item(i), i.e. the corresponding data pointer, is scheduled to appear on the broadcast channel of pico-cell(j) every $s_{pico}(i,j)$ time units (i=1,..,M, j=1,..,N) (i.e. equal-spacing broadcast is assumed - see Appendix 1), where $s_{pico}(i,j)$ is a real value in the range $[0,+\infty)$. Once item(i) or its data pointer appear on a pico-cell broadcast channel, all the corresponding MUs, which need item(i) and happen to be in the waiting phase of the retrieval cycle, are served simultaneously. (Please note, however, there is a qualitative difference between the two cases. If item(i) itself appears on the pico-cell broadcast channel, then the MUs leave the system immediately after the item gets retrieved. On the other hand, if item(i)'s data pointer appears on the pico-cell broadcast channel, the MUs need to tune into the micro-cell broadcast channel and retrieve the actual item from there, before leaving the system (see Figures 6.3 and 6.5).)

Remark 6.1 Let us assume that data-item request probabilities (p(i,j), i=1,..,M, j=1,..,N), with respect to each pico-cells of a hierarchical WDB system are known. Then, as the results presented in Chapter 3 show, there is a simple procedure for identifying which of the given items are worth of broadcasting with respect to the pico-cell(j)'s broadcast service ratio BR, and accordingly the overall system's broadcast service ratio (HBR).[8] Furthermore, as proven in Chapter 3, it is mathematically justifiable to do the following substitution[9]

$$\overline{p}_{new}(i,j) = \begin{cases} \dfrac{\overline{p}(i,j)}{\sum\limits_{i\in\mathcal{M}_1} \overline{p}(i,j)}, & \text{if item(i) is worth of broadcasting in pico-cell(j)} \\ 0, & \text{if item(i) is not worth of broadcasting in pico-cell(j)} \end{cases}$$

[8] See FEB algorithm in Figure 3.3.

[9] Here, analogous to the discussion from Chapter 3, $\mathcal{M}_1$ represents the set of items identified as worth of broadcasting in pico-cell(j).

Remark 6.2 In the remainder of Chapter 6 we assume that the data-item request-probabilities $\bar{p}(i,j)$ (i.e. $p(i,j)=r(j)\ \bar{p}(i,j)$), $i=1,..,M$, $j=1,..,N$, in fact correspond to the substitute values as stated in Remark 6.1.[10] Hence, for those items that correspond to $\bar{p}(i,j)\neq 0$ (i.e. $p(i,j)\neq 0$), the following also holds

$$\frac{l(i)}{\bar{p}(i,j)} \leq \frac{\chi}{\lambda^{*}(1-a)}$$

Consequently,

$$\frac{dp}{\bar{p}(i,j)} \leq \frac{\chi}{\lambda^{*}(1-a)}$$

must hold as well, since $dp \ll l(i)$, $i=1,..,M$. In practical terms the above inequalities suggest that as long as an item satisfies the criterion of being broadcasted in a given pico-cell, inherently its data-pointer will satisfy the same criterion as well.

Here and in the proceeding discussion, r(j) is used to annotated the fraction of users who appear in pico-cell(j). Accordingly, $\bar{p}(i,j)$ annotates the renormalized request-probability of item(i) with respect to pico-cell(j), i.e. $\bar{p}(i,j)=p(i,j)/r(j)$.

[10] In other words, we assume that data-item request probabilities have been 'pre-processed' using FBE algorithm.

6.3 Optimal Broadcast Schedule in HWDB Systems

6.3.1 Optimal Broadcast Schedule in HWDB Systems – Approximate Closed-form Solution

Let us assume that a hybrid HWDB system, as described in Section 6.2, is composed of N pico- and a single overlaying micro- cell. Furthermore, let us assume that a fixed number of items (L) are broadcasted at the micro-cell level, while their data pointers together with the remaining (M-L) items are broadcasted at the pico-cell level. (Please note, here we assume that only the number of items broadcasted at either level is known, not the items themselves.)

Definition 6.1 HWDB System Broadcast Service Ratio (HBR) is a real number $HBR \in [0,1]$ which represents the expected fraction of users who arrive in the system[11] and get served[12] through either of the broadcast channels within an interval of length $C \in R$ [time units], where C satisfies Assumption 6.1.

Assumption 6.1 For every $s_{pico}(i,j) \in R$ there exists a $k(i,j) \in N$ such that $C=k(i,j) \cdot s_{pico}(i,j)$, $(i=1,..,M, j=1,..,N)$.

In practical terms $k(i,j)=C/s_{pico}(i,j)$, represents the number of times that item(i) appears on the broadcast channel of pico-cell(j) within C time units. Similarly as in Chapter 2, we will refer to C as *the duration of an entire broadcast schedule.*

[11] Here, we assume a system of the generalized HWDB type, as defined in Section 6.2.
[12] We assume that a MU gets served as soon as he retrieves either the requested item itself, or the corresponding data-pointer from the nearest pico-cell broadcast channel.

In order to obtain the expression for HBR, first we need to determine the overall number of MUs who arrive in the system and get served through either of the broadcast channels within C. Based on the results from Chapter 2 ((2.31) in particular), it can be easily verified that expression (6.1) represents the expected number of MUs served through the broadcast channel of pico-cell(j) (j=1,..,N).

$$E[n_{BR\_pico-cell(j)}(0, C, item(i), m \cdot s_{pico}(i,j) \mid m = 1,..,k(i,j); i = 1,..,M)] = \tag{6.1}$$
$$= \frac{C\lambda(j)}{\lambda^{*}(1-a)} \sum_{i=1}^{M} \frac{\bar{p}(i,j)}{s_{pico}(i,j)} (1 - e^{-\lambda^{*}(1-a)s_{pico}(i,j)})$$

For better understanding of the above expression, let us annotate the fraction of users who appear in pico-cell(j) with r(j), where

$$r(j) = \sum_{i=1}^{M} p(i,j), \quad j = 1,..,N \tag{6.2}$$

Then, $\lambda(j)=\lambda r(j)$ of (6.1) represents the user arrival rate with respect to pico-cell(j), while $\bar{p}(i,j)= p(i,j)/r(j)$ of (6.1) represents the renormalized probability of item(i) being requested by a MU from pico-cell(j) such that $\sum_{i=1}^{M} \bar{p}(i,j) = 1$.

Based on (6.1), the expected overall number of MUs served through either of the broadcast channels is simply

$$E[n_{HBR}(0, C, item(i), m \cdot s(i,j) \mid m = 1,..,k(i,j); i = 1,..,M; j = 1,..,N)] = \tag{6.3}$$
$$= \sum_{j=1}^{N} E[n_{BR\_pico-cell(j)}(0, C, item(i), m \cdot s_{pico}(i,j) \mid m = 1,..,k(i,j); i = 1,..,M)]$$
$$= \sum_{j=1}^{N} \frac{C\lambda(j)}{\lambda^{*}(1-a)} \sum_{i=1}^{M} \frac{\bar{p}(i,j)}{s_{pico}(i,j)} (1 - e^{-\lambda^{*}(1-a)s_{pico}(i,j)})$$

By employing the following substitution, $\lambda(j)\ \bar{p}(i,j) = \lambda\ r(j)\ p(i,j) / r(j) = \lambda\ p(i,j)$, (6.3) becomes

$$E[n_{HBR}(0,C,item(i),m\cdot s(i,j)\mid m=1,..,k(i,j);\ i=1,..,M;\ j=1,..,N)]= \tag{6.4}$$
$$=\frac{C\lambda}{\lambda^*(1-a)}\sum_{j=1}^{N}\sum_{i=1}^{M}\frac{p(i,j)}{s_{pico}(i,j)}(1-e^{-\lambda^*(1-a)s_{pico}(i,j)})$$

Considering that $C\lambda$ users are expected to arrive in the given HWDB system within the entire C, and $E[n_{HBR}(0,C,item(i),m\cdot s(i,j)\mid m=1,..,k(i,j);\ i=1,..,M;\ j=1,..,N)]$ of them are expected to be served through either of the broadcast channels, then

$$HBR=\frac{1}{\lambda^*(1-a)}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)}{s_{pico}(i,j)}(1-e^{-\lambda^*(1-a)s_{pico}(i,j)}) \tag{6.5}$$

It is interesting to observe that in (6.5) only the values of pico-cell broadcast periods ($s_{pico}(i,j)$, i=1,..,M, j=1,..N) figure as the optimization variables. In other words, the actual value of HBR turns out to be independent of the micro-cell broadcast schedule ($s_{micro}(i)$, i=1,..,L). This result should not surprise, however. Based on the assumptions from Section 6.2, as soon as a user receives the requested data-item, or the corresponding data-pointer, from the nearest pico-cell broadcast channel, the user is considered 'served through broadcasting' from the system's point of view.[13] With this in mind, it seems reasonable to expect that the overall fraction of users served through broadcasting (HBR) mainly depends on the efficiency of the pico-cells' broadcast schedule, as depicted in (6.5). It should be noted, however, that while the efficiency of micro-cell broadcast schedule does not play a significant role with respect to HBR[14], it can affect some other HWDB system performance measures, including the average user waiting time (WT). For example, users that receive a data-pointer from the pico-cell broadcast channel, in place of the actual data-item, still need to wait for the given item to appear at the micro-cell broadcast

[13] By receiving a data-pointer, the user does not get served instantaneously. Nevertheless, he ceases to make further unicast attempts at that point, and tunes into the micro-cell frequency/time slot, specified by the data-pointer, in order to retrieve the actual data item from there.

[14] It can be easily verified that the micro-cell broadcast schedule does not appear in the expressions for UR, LR, RP, WP either.

channel[15]. Clearly, the overall waiting time of these users is directly affected by the efficiency of the micro-cell broadcast channel.

In our analysis of hierarchical WDB systems we consider the broadcast service ratio (HBR) as the performance measure of primary interest. Hence, in the remainder of this section we present the procedure for finding the set of optimal pico-cell broadcast periods with respect to HBR, i.e. the set of pico-cell broadcast periods that can ensure the maximization of (6.5).

Based on the analogy with the single-cell WDB based system, the problem of finding the broadcast scheduling scheme which can ensure the maximization of HBR ($S_{optimal}$), while satisfying the system's physical constraints, can be stated as follows:

$$\text{maximize} \quad HBR(s_{pico}(i,j) \mid i=1,..,M,\ j=1,..,N) = \frac{1}{\lambda^*(1-a)} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{p(i,j)}{s_{pico}(i,j)} (1 - e^{-\lambda^*(1-a) s_{pico}(i,j)}) \quad (6.6.a)$$

$$\text{subject to} \quad C_j(s_{pico}(i,j) \mid i=1,..,M) = \sum_{i=1}^{L} \frac{dp}{s_{pico}(i,j)} + \sum_{i=L+1}^{M} \frac{l(i)}{s_{pico}(i,j)} - 1 = 0, \quad j=1,..,N \quad (6.6.b)$$

$$\text{subject to} \quad C_{N+1}(s_{micro}(i) \mid i=1,..,L) = \sum_{i=1}^{L} \frac{l(i)}{s_{micro}(i)} - 1 = 0 \ ^{16} \quad (6.6.c)$$

In (6.6), dp represents the seize of data pointer in [time units], and it is obtained as dp=*dp*/r, where *dp* is the size of data pointer in [bits] and r is the broadcast channel data rate in [bps]. Each of the N equality constraints in (6.6.b) is associated with the limited capacity of the corresponding pico-cell(j) broadcast channel, j=1,..,N, while the constraint in (6.6.c) is associated

[15] Note, the character of waiting changes once the user receives the data-pointer. Namely, the data-pointer carries the information about the location and time where/when the actual data-item will appear at the micro-cell broadcast channel. Hence, between the time of receiving the data-pointer and the time of the data-item's delivery, the user's end device can switch to 'sleep mode' and thereby significantly reduce its battery consumption.

[16] Please observe, constraint (6.6.c) is not a function of the optimization variables $s_{pico}(i,j)$, (i=1,..,M, j=1,..,N). Hence, it could be entirely omitted.

with the limited capacity of the micro-cell broadcast channel.[17] Also, as indicated earlier, L out M items are assumed to be broadcasted at the micro-cell level, and only their data pointers of size dp [time units] are broadcasted at the pico-cell level. All data pointers as assumed to be of the same size, i.e. dp<<l(i) for every i=1,..,M. Hence, the lengths of only those items that actually occupy the pico-cell broadcast channels' bandwidth (l(i), i=L+1,..,M) appear in (6.6.b).

According to the Lagrange Multiplier theory [*Bertsekas*], $S_{optimal} \in R^M x R^N$ that maximizes (6.6.a) under the constraint (6.6.b) and (6.6.c) can be obtained as the solution to the set of differential equations given in (6.7), i.e. (6.8).

$$\nabla HBR(s_{pico}(i,j) \mid i=1,..,M,\ j=1,..,N) + \sum_{j=1}^{N} K_j \cdot \nabla C_j(s_{pico}(i,j) \mid i=1,..,M,\ j=1,..,N) = 0 \qquad (6.7)$$

$$\frac{\partial HBR(s_{pico}(i,j) \mid i=1,..,M,\ j=1,..,N)}{\partial s_{pico}(i,j)} + \sum_{j=1}^{N} K_j \frac{\partial C_j(s_{pico}(i,j) \mid i=1,..,M)}{\partial s_{pico}(i,j)} = 0, \quad i=1,..,M,\ \ j=1,..,N \qquad (6.8)$$

In (6.7) and (6.8), K_j, j=1,..,N are unknown scalars – Lagrange Multipliers. Also, in these equations we omit writing the constraint (6.6.c), as C_{N+1} is not a function of the optimization variables $s_{pico}(i,j)$, (i=1,..,M, j=1,..,N).

By employing the actual expressions for HBR and C_j (j=1,..,N), (3.8) gets transformed to

$$\frac{p(i,j)}{\lambda^*(1-a)} \frac{\lambda^*(1-a)e^{-\lambda^*(1-a)s_{pico}(i,j)} \cdot s_{pico}(i,j) - 1 + e^{-\lambda^*(1-a)s_{pico}(i,j)}}{s_{pico}(i,j)^2} - K_j \frac{dp}{s_{pico}(i,j)^2} = 0, \quad i=1,..,L,\ \ j=1,..,N \ (6.9.a)$$

$$\frac{p(i,j)}{\lambda^*(1-a)} \frac{\lambda^*(1-a)e^{-\lambda^*(1-a)s_{pico}(i,j)} \cdot s_{pico}(i,j) - 1 + e^{-\lambda^*(1-a)s_{pico}(i,j)}}{s_{pico}(i,j)^2} - K_j \frac{l(i)}{s_{pico}(i,j)^2} = 0, \quad i=L+1,..,M, j=1,..,N \ (6.9.b)$$

[17] In this particular case, the micro-cell bandwidth constraint (6.6.c) is not a function of the optimization variables $s_{pico}(i,j)$, (i=1,..,M, j=1,..,N). Hence, it could be entirely omitted from (6.6).

Note, (6.9.a) corresponds to $s_{pico}(i,j)$-s of those items that get broadcasted at the micro-cell level, while (6.9.b) corresponds to $s_{pico}(i,j)$-s of the data-pointers associated with the remaining (M-L) items that get broadcasted at the pico-cell level. By rearranging (6.9), (6.10) is obtained.

$$\frac{p(i,j)}{dp}\left(\lambda^*(1-a)e^{-\lambda^*(1-a)s_{pico}(i,j)}\cdot s_{pico}(i,j)-1+e^{-\lambda^*(1-a)s_{pico}(i,j)}\right)=K_j\lambda^*(1-a),\quad i=1,..,L,\ \ j=1,..,N \quad (6.10.a)$$

$$\frac{p(i,j)}{l(i)}\left(\lambda^*(1-a)e^{-\lambda^*(1-a)s_{pico}(i,j)}\cdot s_{pico}(i,j)-1+e^{-\lambda^*(1-a)s_{pico}(i,j)}\right)=K_j\lambda^*(1-a),\quad i=L+1,..,M,\ \ j=1,..,N \quad (6.10.b)$$

Clearly, (6.10) cannot be transformed any further, so as to obtain closed form expressions for $s_{pico}(i,j)$ (i=1,..,M, j=1,..,N) that satisfy (6.6). However, in the proceeding analysis, by employing the results presented in Chapter 3, we derive an approximate closed-form solution to (6.6).

In particular, based on the discussion from Section 3.2.2 and Remark 6.2 (see pp. 6-13), the following function approximation can be employed for every element of (6.10).

$$\lambda^*(1-a)e^{-\lambda^*(1-a)s_{pico}(i,j)}\cdot s_{pico}(i,j)-1+e^{-\lambda^*(1-a)s_{pico}(i,j)}\approx k\lambda^*(1-a)s_{pico}(i,j),\ \ i=1,..,M,\ \ j=1,..,N \quad (6.11)$$

By using the above approximation, (6.10) gets transformed to

$$\frac{p(i,j)}{dp}\left(k\lambda^*(1-a)s_{pico}(i,j)\right)=K_j\lambda^*(1-a),\quad i=1,..,L,\ \ j=1,..,N \quad (6.12.a)$$

$$\frac{p(i,j)}{l(i)}\left(k\lambda^*(1-a)s_{pico}(i,j)\right)=K_j\lambda^*(1-a),\quad i=L+1,..,M,\ \ j=1,..,N \quad (6.12.b)$$

Through simple rearrangement, (6.12) becomes

$$s_{pico}(i,j) = \frac{K_j}{k}\frac{dp}{p(i,j)}, \quad i=1,..,L, \ j=1,..,N \tag{6.13.a}$$

$$s_{pico}(i,j) = \frac{K_j}{k}\frac{l(i)}{p(i,j)}, \quad i=L+1,..,M, \ j=1,..,N \tag{6.13.b}$$

In (6.13), the exact values of parameters Kj (j=1,..,N) are unknown. However, by employing (6.13) back in (6.6.b), as presented in (6.14),

$$C_j(s_{pico}(i,j) \mid i=1,..,M) = \sum_{i=1}^{L} p(i,j)\frac{k}{K_j} + \sum_{i=L+1}^{M} p(i,j)\frac{k}{K_j} - 1 = 0, \quad j=1,..,N \tag{6.14}$$

the values of K_j, j=1,..,N, are obtained

$$K_j = k\sum_{i=1}^{M} p(i,j), \quad j=1,..,N \tag{6.15}$$

Finally, by replacing K_j, j=1,..,N, in (6.13) with (6.15), (6.13) becomes

$$s_{pico}(i,j) = \frac{dp}{\frac{p(i,j)}{\sum_{i=1}^{M} p(i,j)}} = \frac{dp}{\bar{p}(i,j)}, \quad i=1,..,L, \ j=1,..,N \tag{6.16.a}$$

$$s_{pico}(i,j) = \frac{l(i)}{\frac{p(i,j)}{\sum_{i=1}^{M} p(i,j)}} = \frac{l(i)}{\bar{p}(i,j)}, \quad i=L+1,..,M, \ j=1,..,N \tag{6.16.b}$$

Let us annotate $\{s_{pico}(i,j) \mid i=1,..,M, j=1,..,N\}$ of (6.16) with $\breve{S}_{optimal}$. Hence, $\breve{S}_{optimal}$ represents the approximate explicit form solution to (6.6), i.e. it represents an approximation of the optimal broadcast scheduling scheme in a hybrid HWDB based system, in the sense of Definition 6.1.

Remark 6.3 The derivation of expression (6.16) has been based on the assumption that the exact number of items broadcasted at the pico, i.e. and micro level is known. However, the identities of these items have not been specified. Thus, the actual search for $\breve{S}_{optimal}$ does not end at (6.16), but it also need to include a mechanism for identification of the optimal data-item partitioning, which would ensure the ultimate maximization of HBR.[18]

Remark 6.4 If L out M data-items can be accepted for broadcasting at the micro-cell level of a hierarchical WDB system, the overall number of possible data-item partitionings that need to examined for optimality is M!/L!(M-L)!. Hence in order to obtain the ultimate $S_{optimal}$ in such a system, one would have to find a solution to (6.6) for each of M!/L!(M-L)! combinations, and then select the combination that provides the highest value of HBR($S_{optimal}$).

The following two examples illustrate the practical significance of Remark 6.4: 1) if M=10 and L=3, then the overall number of possible partitionings is 120; 2) if M=10 and L=4, the overall number of possible partitionings increases to 210, etc. Clearly, the computational complexity associated with the search for the ultimate $S_{optimal}$ is considerable, even for small collections of broadcast data-items.

In the following section we propose a computationally efficient algorithm for optimal partitioning of broadcast items, such as to provide for the absolute maximization of the corresponding HBR.

[18] In other words, (6.16) enables us to find the set of optimal pico-cell broadcast periods for a specific (i.e. fixed) data-item partitioning. However, data-item partitioning is a subject to optimization itself. Hence, the space of all possible data-item partitionings should be searched in order to identify $\breve{S}_{optimal}$ that would eventually result in the highest HBR($\breve{S}_{optimal}$).

6.3.2 Optimal Layer-based Partitioning of Broadcast Items in Hierarchical WDB Systems

The expression for HBR at $\breve{S}_{\text{optimal}}$ is obtained simply by employing (6.16) back in (6.5). Thus,

$$\mathrm{HBR}(\breve{S}_{\text{optimal}}) = \frac{1}{\lambda^*(1-a)} \sum_{j=1}^{N} \left[\sum_{i=1}^{L} p(i,j) \frac{\overline{p}(i,j)}{dp} (1 - e^{-\lambda^*(1-a)\frac{dp}{\overline{p}(i,j)}}) + \sum_{i=L+1}^{M} p(i,j) \frac{\overline{p}(i,j)}{l(i)} (1 - e^{-\lambda^*(1-a)\frac{l(i)}{\overline{p}(i,j)}}) \right] \quad (6.17)$$

By rearranging the order of summation, (6.17) becomes

$$\mathrm{HBR}(\breve{S}_{\text{optimal}}) = \frac{1}{\lambda^*(1-a)} \left[\sum_{i=1}^{L} \left(\sum_{j=1}^{N} \frac{p(i,j)\overline{p}(i,j)}{dp} (1 - e^{-\lambda^*(1-a)\frac{dp}{\overline{p}(i,j)}}) \right) + \right. \quad (6.18)$$
$$\left. + \sum_{i=L+1}^{M} \left(\sum_{j=1}^{N} \frac{p(i,j)\overline{p}(i,j)}{l_i} (1 - e^{-\lambda^*(1-a)\frac{l(i)}{\overline{p}(i,j)}}) \right) \right]$$

Furthermore, by replacing p(i,j) with r(j) $\overline{p}$(i,j), (6.18) gets transformed to

$$\mathrm{HBR}(\breve{S}_{\text{optimal}}) = \frac{1}{\lambda^*(1-a)} \left[\sum_{i=1}^{L} \left(\sum_{j=1}^{N} r(j) \frac{\overline{p}(i,j)^2}{dp} (1 - e^{-\lambda^*(1-a)\frac{dp}{\overline{p}(i,j)}}) \right) + \right. \quad (6.19)$$
$$\left. + \sum_{i=L+1}^{M} \left(\sum_{j=1}^{N} r(j) \frac{\overline{p}(i,j)^2}{l(i)} (1 - e^{-\lambda^*(1-a)\frac{l(i)}{\overline{p}(i,j)}}) \right) \right]$$

Finally, with the substitution defined in (6.20),

$$G_i(x) = \sum_{j=1}^{N} r(j) \frac{\overline{p}(i,j)^2}{x} (1 - e^{-\lambda^*(1-a)\frac{x}{\overline{p}(i,j)}}) \quad (6.20)$$

(6.19) can be rewritten as

$$\mathrm{HBR}(\breve{S}_{\mathrm{optimal}}) = \frac{1}{\lambda^{*}(1-a)}\left[\sum_{i=1}^{L} G_i(dp) + \sum_{i=L+1}^{M} G_i(l(i))\right] \tag{6.21}$$

or

$$\mathrm{HBR}(\breve{S}_{\mathrm{optimal}}) = \frac{1}{\lambda^{*}(1-a)}\left[\sum_{i=1}^{L}\left[G_i(dp) - G_i(l(i))\right] + \sum_{i=1}^{M} G_i(l(i))\right] \tag{6.22}$$

Based on (6.21), it is clear that those items that get broadcasted at the micro-cell level contribute to the overall $\mathrm{HBR}(\breve{S}_{\mathrm{optimal}})$ through the corresponding $G_i(dp)$. On the other hand, the items that get broadcasted at the pico-cell level contribute to $\mathrm{HBR}(\breve{S}_{\mathrm{optimal}})$ through $G_i(l(i))$[19].

Now, if we assume that the grouping of items in the two sets (those that get broadcasted at pico-, i.e. micro- cell level) is made arbitrary, then based on (6.22), the largest $\mathrm{HBR}(\breve{S}_{\mathrm{optimal}})$ is obtained for the grouping that provides the maximization of (6.23). (Note, the second summation in (6.22) remains the same for any particular grouping of broadcast items.)

$$\sum_{i=1}^{L}\left[G_i(dp) - G_i(l(i))\right] \tag{6.23}$$

In practical terms, (3.23) gets maximized if L items, which individually provide the largest difference $\left[G_i(dp) - G_i(l(i))\right]$, are selected for broadcasting at the micro-cell level.

Based on the preceding discussion, in Figure 6.6 we propose an algorithm for optimal layer-based grouping of broadcast items in a hybrid HWDB based system.

[19] Please note, $G_i(x)$ is a monotonically decreasing function, so $G_i(dp)-G_i(l(i))>0$, since $dp<<l(i)$, for every $i=1,..,M$.

Step 1 Relative popularities p(i,j), with respect to pico-cell(j) (j=1,..,N), and length l(i) of item(i) are known (i=1,..,M).

For every item(i) calculate

$$d(i) = G_i(dp) - G_i(l(i)) = \sum_{j=1}^{N} r(j)\overline{p}(i,j)^2 \left(\frac{(1 - e^{-\lambda^*(1\text{-}a)\frac{dp}{\overline{p}(i,j)}})}{dp} - \frac{(1 - e^{-\lambda^*(1\text{-}a)\frac{l(i)}{\overline{p}(i,j)}})}{l(i)} \right) \quad (6.24)$$

Step 2 Sort d(i)-s of Step 1 in the descending order.

Step 3 Items that correspond the first L elements of the array obtained in Step 2 get placed in the group that is broadcasted at the micro-cell level. The remaining (M-L) items get placed in the group that is broadcasted at the pico-cell level.

Figure 6.6 Algorithm for optimal layer-based partitioning of broadcast items in hybrid HWDB based system

6.3.3 Special Case Analysis

According to the algorithm proposed in Figure 6.6, the decision on whether an item should be broadcasted at the pico- or micro- cell level, in order to maximize the corresponding HBR, is based on (6.24). Hence, the larger d(i) (of (6.24)) associated with item(i), the more likely this item will get broadcasted at the micro-cell level. It is interesting to observe, though, that the actual value of d(i) depends on item(i)'s length (l(i)), request probabilities ($\overline{p}(i,j)$), and the corresponding pico-cell MU arrival probabilities (r(j)), i=1,..,M, j=1,..,N. In this section, through the analysis of two special cases, we investigate how the given parameters affect the value of d(i) (i=1,..,M) and, consequently, the decision process regarding the optimal layer-based grouping of broadcast items.

Lemma 6.1 In order to maximize HBR in a hybrid HWDB system, longer items should be broadcasted at the micro-cell level.

Proof of Lemma 6.1

In order to determine how item(i)'s length l(i) (i=1,..,M) affects the decision on whether this item should be broadcasted at pico- or micro- cell level, let us observe a hybrid HWDB system with equal pico-cell MU arrival probabilities (r(j)=r(k), $\forall$j,k=1,..,N), and equal item request probabilities ($\overline{p}(i,j)=\overline{p}(k,j)$=p, $\forall$i,k=1,..,M). Under such assumptions, (6.24) gets transformed to (6.25).

$$d(i) = p^2\left(\frac{(1-e^{-\lambda^*(1-a)\frac{dp}{p}})}{dp} - \frac{(1-e^{-\lambda^*(1-a)\frac{l(i)}{p}})}{l(i)}\right) \tag{6.25}$$

Since, based on (6.25), for i≠k, i,k=1,..,M the following holds

$$l(i) > l(k) \quad \Rightarrow \quad \frac{(1-e^{-\lambda^*(1-a)\frac{l(i)}{p}})}{l(i)} < \frac{(1-e^{-\lambda^*(1-a)\frac{l(k)}{p}})}{l(k)} \quad \Rightarrow \quad d(i) > d(k) \tag{6.26}$$

Hence (according to Step 3 of Figure 6.6), in order to maximize HBR longer items should be broadcasted at the micro-cell level.

End of Proof

Lemma 6.2 In order to maximize HBR in a hybrid HWDB system, more popular items should be broadcasted at the micro-cell level.

Proof of Lemma 6.2

In order to determine how item(i)'s request probabilities p(i,j) (i=1,..,M, j=1,..,N) affect the decision on whether this item should be broadcasted at pico- or micro- cell level, let us observe a hybrid HWDB system with equal pico-cell MU arrival probabilities (r(j)=r(k), ∀j,k=1,..,N), and equal item lengths (l(i)=l(k), ∀i,k=1,..,M). Under such assumptions, $\overline{p}(i,j)=\overline{p}(i,k)=p(i)$, and (6.24) gets transformed to (6.27).

$$d(i) = p(i)^2\left(\frac{(1-e^{-\lambda^*(1-a)\frac{dp}{p(i)}})}{dp} - \frac{(1-e^{-\lambda^*(1-a)\frac{1}{p(i)}})}{1}\right) \tag{6.27}$$

By using the following substitution, $\lambda^*(1\text{-}a)=A$, $\lambda^*(1\text{-}a)\cdot dp=A\cdot dp=D$, and $\lambda^*(1\text{-}a)\cdot l=A\cdot l=E$, (6.27) becomes

$$d_i = Ap(i)^2\left(\frac{(1-e^{-\frac{D}{p(i)}})}{D} - \frac{(1-e^{-\frac{E}{p(i)}})}{E}\right) \tag{6.28}$$

(Note, dp<<l implies D<E.) Figure 6.7 suggests that (6.28) represents a monotonically increasing function with respect to p(i), for the entire range of E/D>1.

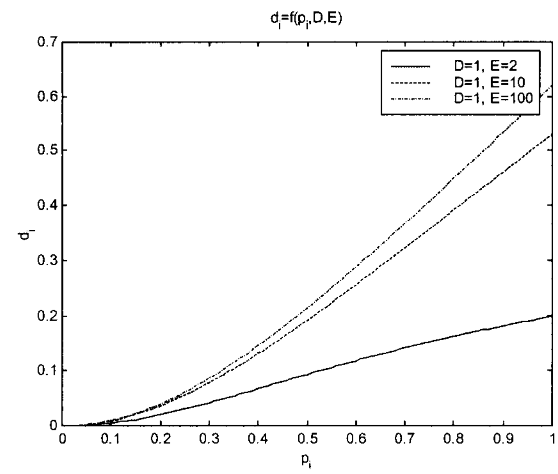


Figure 6.7 Monotonically increasing nature of d(i)=d(i)(p(i))

Thus, based on Figure 6.7, the following holds

$$p(i) > p(k) \quad \Rightarrow \quad d(i) > d(k) \tag{6.29}$$

for i≠k, i,k=1,..,M. Accordingly, we conclude that in order to maximize HBR more popular items (i.e. items of higher request probabilities) should be broadcasted at the micro-cell level.

End of Proof

6.3.4 Experimental Results

In this section, through a series of experimental results, we compare the actual value and performance of HWDB systems' optimal broadcast schedule ($S_{optimal}$), as obtained: 1) by solving (6.6) with Matlab's Optimization Toolbox [*MathWorks*], and 2) using our algorithm as outlined in Figure 6.6.

In all the experiments, the following assumptions are employed:

a) the observed HWDB system consists of 6 pico- and a single micro- cell
b) the overall number of data-items available in the system is 6
c) only 2 out of 6 items are allowed to be broadcasted at the micro-cell level
d) the data-pointers are of the size dp=1, while $\lambda^{*}(1\text{-}a)=0.1$
e) the data-item's sizes and popularities are varied – see Table 6.1 through 6.8

Under the given assumptions, the overall number of possible layer-based data-item partitionings is 6!/4!2!=15. Hence, in all the experiments, the numerical estimation approach requires 15 simulation runs in order to find the best data-item partitioning, i.e. the ultimate $S_{optimal}$ (see Table 6.1, 6.3, 6.5, and 6.7). At the same time, our algorithm identifies an approximate $S_{optimal}$ in a single round of computation (see Table 6.2, 6.4, 6.6, and 6.8). Please note that in Tables 6.1, 6.3, 6.5, and 6.7, the first column indicates the assumed layer-based data-items partitioning. Also, in these tables, the ultimate $S_{optimal}$, i.e. $S_{optimal}$ that results in the largest HBR are highlighted. On the other hand, in Table 6.2, 6.4, 6.6, and 6.8, the fourth raw contains the values of d(i) (i=1,..,6), as obtained using (6.24). The largest two d(i)-s, which correspond to the items that should be broadcasted at the micro-cell layer (see Figure 6.6), are also highlighted.

The results obtained in all the experiments appear very encouraging - they clearly indicate that our approximate $\breve{S}_{optimal}$, although being computationally significantly less expensive than $S_{optimal}$ obtained through numerical estimation, is still capable of providing equally good performance in terms of the overall HBR.

6.3.4.1 Experiment 1: Uniform Data-Item Popularities, Variable Data-Item Lengths

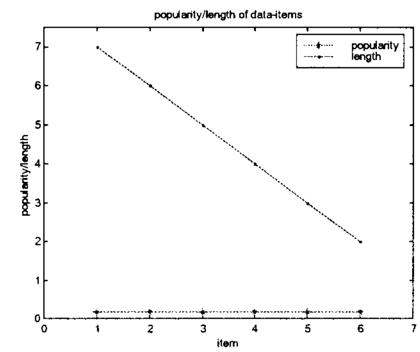


Figure 6.8 Data-items' lengths and popularities in Experiment 1

	item	1	2	3	4	5	6	
	$\bar{p}(i,j)$	0.167	0.167	0.167	0.167	0.167	0.167	
	l(i) [time]	7	6	5	4	3	2	
items 1, 2 – at micro level	$S_{optimal}$	7.0791	7.0787	29.4620	21.5435	16.0161	11.4431	**HBR($S_{optimal}$)=0.5411**
items 1, 3 – at micro level	$S_{optimal}$	6.9220	41.4383	6.9208	20.6857	15.5149	11.1411	HBR($S_{optimal}$)=0.5335
items 1, 4 – at micro level	$S_{optimal}$	6.8729	40.0508	27.3318	6.8741	15.3655	11.0474	HBR($S_{optimal}$)=0.5229
items 1, 5 – at micro level	$S_{optimal}$	6.8752	40.0804	27.3390	20.4482	6.8766	11.0543	HBR($S_{optimal}$)=0.5086
items 1, 6 – at micro level	$S_{optimal}$	6.9250	41.4767	27.8008	20.6968	15.5158	6.9220	HBR($S_{optimal}$)=0.4894
items 2, 3 – at micro level	$S_{optimal}$	76.0798	6.6061	6.6082	19.0986	14.5426	10.5508	HBR($S_{optimal}$)=0.5289
items 2, 4 – at micro level	$S_{optimal}$	70.4941	6.5930	24.8710	6.5880	14.5347	10.5362	HBR($S_{optimal}$)=0.5180
items 2, 5 – at micro level	$S_{optimal}$	72.6461	6.5982	24.8702	19.0662	6.6000	10.5412	HBR($S_{optimal}$)=0.5038
items 2, 6 – at micro level	$S_{optimal}$	84.6092	6.6159	25.0014	19.1311	14.5704	6.6161	HBR($S_{optimal}$)=0.4848
items 3, 4 – at micro level	$S_{optimal}$	59.4868	32.9146	6.6407	6.5687	14.6571	10.3876	HBR($S_{optimal}$)=0.5097
items 3, 5 – at micro level	$S_{optimal}$	62.3121	33.2777	6.5634	18.8936	6.5649	10.4731	HBR($S_{optimal}$)=0.4956
items 3, 6 – at micro level	$S_{optimal}$	68.8816	33.7498	6.5923	19.0245	14.4980	6.5909	HBR($S_{optimal}$)=0.4766
items 4, 5 – at micro level	$S_{optimal}$	49.5613	37.2555	24.4008	6.5847	6.6519	10.4924	HBR($S_{optimal}$)=0.4845
items 4, 6 – at micro level	$S_{optimal}$	65.3760	33.5304	24.6974	6.5796	14.4569	6.5773	HBR($S_{optimal}$)=0.4657
items 5, 6 – at micro level	$S_{optimal}$	54.4819	34.8096	25.6295	19.3886	6.6772	6.7571	HBR($S_{optimal}$)=0.4514

Table 6.1 Numerical estimation of $S_{optimal}$ in Experiment 1

item	**1**	**2**	**3**	**4**	**5**	**6**
$\bar{p}(i,j)$	0.167	0.167	0.167	0.167	0.167	0.167
l(i) [time]	7	6	5	4	3	2
$G_i(dp)-G_i(l(i))$	0.0086	0.0080	0.0072	0.0062	0.0048	0.0028
$S_{optimal}$	$dp/p_i=5.9880$	$dp/p_i=5.9880$	$l_i/p_i=29.9401$	$l_i/p_i=23.9521$	$l_i/p_i=17.9641$	$l_i/p_i=11.9760$
HBR($\breve{S}_{optimal}$)=0.5393 ⇒ estimation error with respect to HBR($S_{optimal}$): 0.33%						

Table 6.2 Hierarchical-FBE algorithm approach to estimation of $S_{optimal}$ in Experiment 1

6.3.4.2 Experiment 2: Variable Data-Item Popularities, Uniform Data-Item Lengths

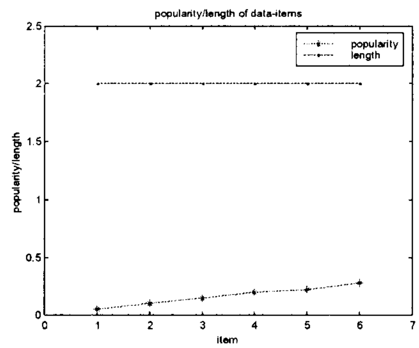


Figure 6.9 Data-items' lengths and popularities in Experiment 2

	item	1	2	3	4	5	6	
	$\bar{p}(i,j)$	0.05	0.10	0.15	0.2	0.22	0.28	
	l(i) [time]	2	2	2	2	2	2	
items 1, 2 – at micro level	$S_{optimal}$	17.3281	9.8403	12.1915	9.8402	9.1993	7.8136	$HBR(S_{optimal})=0.6395$
items 1, 3 – at micro level	$S_{optimal}$	16.9526	16.9493	7.3587	9.6835	9.0587	7.6983	$HBR(S_{optimal})=0.6465$
items 1, 4 – at micro level	$S_{optimal}$	16.6856	16.6722	11.8244	6.0684	8.9510	7.6122	$HBR(S_{optimal})=0.6523$
items 1, 5 – at micro level	$S_{optimal}$	16.5809	16.5825	11.7677	9.5247	5.7026	7.5820	$HBR(S_{optimal})=0.6543$
items 1, 6 – at micro level	$S_{optimal}$	16.3186	16.3261	11.6190	9.4162	8.8131	4.8840	$HBR(S_{optimal})=0.6599$
items 2, 3 – at micro level	$S_{optimal}$	39.8422	9.4339	7.1866	9.4358	8.8296	7.5136	$HBR(S_{optimal})=0.6565$
items 2, 4 – at micro level	$S_{optimal}$	39.8050	9.3049	11.4746	5.9226	8.7114	7.4195	$HBR(S_{optimal})=0.6621$
items 2, 5 – at micro level	$S_{optimal}$	39.7957	9.2610	11.4114	9.2613	5.5633	7.3856	$HBR(S_{optimal})=0.6641$
items 2, 6 – at micro level	$S_{optimal}$	39.7145	9.1380	11.2507	9.1403	8.5603	4.7621	$HBR(S_{optimal})=0.6696$
items 3, 4 – at micro level	$S_{optimal}$	39.8248	15.7026	6.9808	5.8301	8.5626	7.2975	$HBR(S_{optimal})=0.6688$
items 3, 5 – at micro level	$S_{optimal}$	39.8137	15.6043	6.9501	9.0959	5.4795	7.2638	$HBR(S_{optimal})=0.6708$
items 3, 6 – at micro level	$S_{optimal}$	39.7235	15.3399	6.8665	8.9787	8.4131	4.6912	$HBR(S_{optimal})=0.6762$
items 4, 5 – at micro level	$S_{optimal}$	39.7928	15.3298	11.0270	5.7350	5.4137	7.1732	$HBR(S_{optimal})=0.6763$
items 4, 6 – at micro level	$S_{optimal}$	39.6962	15.0681	10.8752	5.6697	8.3014	4.6365	$HBR(S_{optimal})=0.6816$
items 5, 6 – at micro level	$S_{optimal}$	39.6604	14.9760	10.8195	8.8148	5.3281	4.6168	**$HBR(S_{optimal})=0.6836$**

Table 6.3 Numerical estimation of $S_{optimal}$ in Experiment 2

item	1	2	3	4	5	6
$\bar{p}(i,j)$	0.05	0.10	0.15	0.2	0.22	0.28
l(i) [time]	2	2	2	2	2	2
$G_i(dp)-G_i(l(i))$	9.3059e-004	0.0020	0.0026	0.0031	0.0032	0.0035
$S_{optimal}$	$l_i/p_i=40$	$l_i/p_i=20$	$l_i/p_i=13.3333$	$l_i/p_i=10$	$dp/p_i=4.5455$	$dp/p_i=3.5714$
$HBR(\breve{S}_{optimal})=0.6770 \Rightarrow$ estimation error with respect to $HBR(S_{optimal})$: 0.97%						

Table 6.4 Hierarchical-FBE algorithm approach to estimation of $S_{optimal}$ in Experiment 2

6.4.3.3 Experiment 3: Variable Data-Item Popularities, Variable Data-Item Lengths

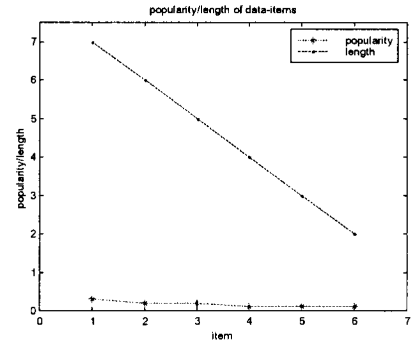


Figure 6.10 Data-items' lengths and popularities in Experiment 3

	item	1	2	3	4	5	6	
	$\overline{p}(i,j)$	0.3	0.2	0.2	0.1	0.1	0.1	
	l(i) [time]	7	6	5	4	3	2	
items 1, 2 – at micro level	$S_{optimal}$	4.5575	5.8079	19.4368	38.0150	25.1269	15.8052	**$HBR(S_{optimal})$=0.5933**
items 1, 3 – at micro level	$S_{optimal}$	4.5414	24.0950	5.8094	38.9295	23.4924	15.5998	$HBR(S_{optimal})$=0.5827
items 1, 4 – at micro level	$S_{optimal}$	4.7782	26.9920	21.4858	9.6752	26.0947	17.0365	$HBR(S_{optimal})$=0.5374
items 1, 5 – at micro level	$S_{optimal}$	4.7005	25.1244	20.8431	48.6655	9.4190	16.6393	$HBR(S_{optimal})$=0.5304
items 1, 6 – at micro level	$S_{optimal}$	4.7115	26.6418	20.5638	39.4751	26.6441	9.4810	$HBR(S_{optimal})$=0.5183
items 2, 3 – at micro level	$S_{optimal}$	19.5870	6.0918	6.0905	43.1947	29.3275	16.7532	$HBR(S_{optimal})$=0.5348
items 2, 4 – at micro level	$S_{optimal}$	25.7312	6.1428	17.4180	33.9705	27.3623	14.3849	$HBR(S_{optimal})$=0.4675
items 2, 5 – at micro level	$S_{optimal}$	26.2356	5.7301	18.0047	40.3517	20.6697	14.7213	$HBR(S_{optimal})$=0.4700
items 2, 6 – at micro level	$S_{optimal}$	21.4262	6.2655	23.6432	41.0929	28.1141	10.1838	$HBR(S_{optimal})$=0.4689
items 3, 4 – at micro level	$S_{optimal}$	21.7486	28.8727	6.4154	9.7695	27.7363	19.2349	$HBR(S_{optimal})$=0.4773
items 3, 5 – at micro level	$S_{optimal}$	21.2260	29.5373	6.3057	42.9740	9.9788	17.3601	$HBR(S_{optimal})$=0.4706
items 3, 6 – at micro level	$S_{optimal}$	20.5548	30.0998	6.2417	42.9625	29.1871	9.6138	$HBR(S_{optimal})$=0.4588
items 4, 5 – at micro level	$S_{optimal}$	22.3037	32.8767	24.3753	10.4216	10.4149	18.7695	$HBR(S_{optimal})$=0.4229
items 4, 6 – at micro level	$S_{optimal}$	22.0151	32.1191	24.0219	10.3392	32.0913	10.3214	$HBR(S_{optimal})$=0.4113
items 5, 6 – at micro level	$S_{optimal}$	20.1349	27.9538	21.7751	1.0524e+3	9.7574	9.7848	$HBR(S_{optimal})$=0.4063

Table 6.5 Numerical estimation of $S_{optimal}$ in Experiment 3

item	1	2	3	4	5	6
$\overline{p}(i,j)$	0.3	0.2	0.2	0.1	0.1	0.1
l(i) [time]	7	6	5	4	3	2
$G_i(dp)-G_i(l(i))$	0.0138	0.0093	0.0083	0.0038	0.0031	0.0020
$S_{optimal}$	dp/p_i=3.33	dp/p_i=5	l_i/p_i=25	l_i/p_i=40	l_i/p_i=30	l_i/p_i=20
$HBR(\breve{S}_{optimal})$=0.5854 ⇒ estimation error with respect to $HBR(S_{optimal})$: 1.35%						

Table 6.6 Hierarchical-FBE algorithm approach to estimation of $S_{optimal}$ in Experiment 3

6.3.4.4 Experiment 4: Variable Data-Item Popularities, Variable Data-Item Lengths

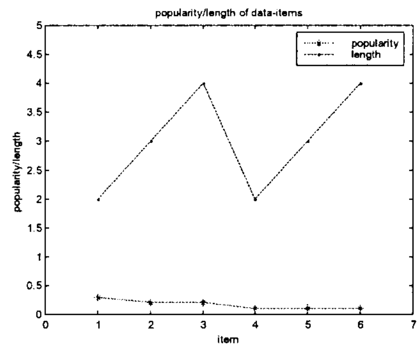


Figure 6.11 Data-items' lengths and popularities in Experiment 4

	item	1	2	3	4	5	6	
	$\overline{p}(i,j)$	0.3	0.2	0.2	0.1	0.1	0.1	
	l(i) [time]	2	3	4	2	3	4	
items 1, 2 – at micro level	$S_{optimal}$	4.5356	5.8623	15.3647	15.3336	24.4274	41.7658	$BR(S_{optimal})=0.6068$
items 1, 3 – at micro level	$S_{optimal}$	4.5346	12.1849	5.7780	15.4367	23.8142	38.2195	$BR(S_{optimal})=0.6234$
items 1, 4 – at micro level	$S_{optimal}$	4.6558	12.6566	16.1654	9.3463	25.5943	52.2537	$BR(S_{optimal})=0.5725$
items 1, 5 – at micro level	$S_{optimal}$	4.6820	12.7141	16.2834	16.2762	9.3998	52.9490	$BR(S_{optimal})=0.5845$
items 1, 6 – at micro level	$S_{optimal}$	4.8107	13.2004	17.0059	17.0051	27.5104	9.7097	$BR(S_{optimal})=0.5914$
items 2, 3 – at micro level	$S_{optimal}$	6.8827	5.7451	5.7552	15.3989	23.6082	38.2246	**$BR(S_{optimal})=0.6360$**
items 2, 4 – at micro level	$S_{optimal}$	7.1606	6.0068	16.2048	9.9369	25.7492	44.3186	$BR(S_{optimal})=0.5851$
items 2, 5 – at micro level	$S_{optimal}$	7.1472	5.9658	16.2509	16.2451	9.3846	52.1271	$BR(S_{optimal})=0.5972$
items 2, 6 – at micro level	$S_{optimal}$	7.3696	6.1404	16.9842	16.9742	27.4522	9.6959	$BR(S_{optimal})=0.6042$
items 3, 4 – at micro level	$S_{optimal}$	7.1463	12.8674	5.9837	9.4973	24.9572	42.3863	$BR(S_{optimal})=0.6020$
items 3, 5 – at micro level	$S_{optimal}$	7.1115	12.6004	5.9325	16.1208	9.3284	49.4733	$BR(S_{optimal})=0.6140$
items 3, 6 – at micro level	$S_{optimal}$	7.3227	13.0765	6.1040	16.8280	27.0333	9.6332	$BR(S_{optimal})=0.6211$
items 4, 5 – at micro level	$S_{optimal}$	7.3008	13.0219	16.7427	9.5956	9.5941	82.7813	$BR(S_{optimal})=0.5625$
items 4, 6 – at micro level	$S_{optimal}$	7.6405	13.8026	17.9312	10.0895	30.1219	10.0905	$BR(S_{optimal})=0.5687$
items 5, 6 – at micro level	$S_{optimal}$	7.7123	13.9740	18.1916	18.1927	10.1936	10.1965	$BR(S_{optimal})=0.5804$

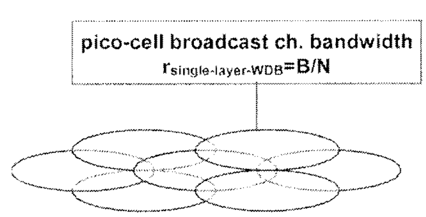
Table 6.7 Numerical estimation of $S_{optimal}$ in Experiment 4

item	**1**	**2**	**3**	**4**	**5**	**6**
$\overline{p}(i,j)$	0.3	0.2	0.2	0.1	0.1	0.1
l(i) [time]	2	3	4	2	3	4
$G_i(dp)-G_i(l(i))$	0.0036	0.0053	0.0070	0.0020	0.0031	0.0038
$S_{optimal}$	$l_i/p_i=6.6667$	$dp/p_i=5$	$dp/p_i=5$	$l_i/p_i=20$	$l_i/p_i=30$	$l_i/p_i=40$
$HBR(\breve{S}_{optimal})=0.6332$ $\Rightarrow$ estimation error with respect to $HBR(S_{optimal})$: 0.44 %						

Table 6.8 Hierarchical-FBE algorithm approach to estimation of $S_{optimal}$ in Experiment 4

6.4 Comparative Performance Analysis

Based on the discussion presented in the preceding sections, multicell HWDB systems appear to be considerably more complex, from an operational point of view, than single-layer WDB systems composed of an equal number of independent pico-cells (Figure 6.12). For example, the data retrieval procedure in an HWDB system requires that some form of cooperation between the broadcast channels of pico- and micro- cell level exists (see Section 6.2 and Figure 6.5). Consequently, the maximization of the systems' throughput[20] (i.e. HBR) requires that, in addition to the estimation of optimal inter-broadcast intervals, optimal layer-based item grouping be obtained as well (see Section 6.3). On the other hand, in single-layer WDB systems composed of independent pico-cells neither of the two is required.



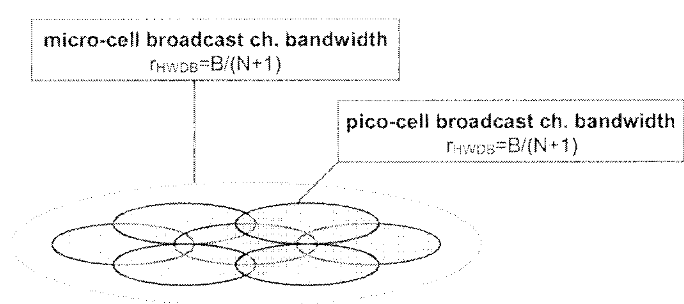


Figure 6.12.a Single-layer multicell WDB system Figure 6.12.b Hierarchical WDB system

In this section, we present the analysis concerning performance comparison of the two types of multicell WDB systems, presented in Figure 6.12, under some special constraints. In particular,

1) in Section 6.4.2 we investigate which of the two systems provides a higher throughput (i.e. BR/HBR), assuming the overall available broadcast bandwidth is fixed and partitioned equally among all the cells;

[20] Please recall, there is a dependency between the overall throughput of a generalized WDB system and its broadcast service ratio (HBR). Namely, in Chapter 2 we proved that the maximization of the broadcast service ratio directly implies the maximization of the overall system's throughput.

2) in Section 6.4.3, first we introduce the concept of bandwidth cost. Subsequently, we investigate which of the two systems provides a higher throughput per overall broadcast bandwidth cost. In both cases, the HWDB system appears to provide superior performance, thus justifying its higher operational complexity compared to the system composed of independent pico-cells only.

6.4.1 Performance Comparison under Fixed Broadcast Bandwidth Constraint

Let us assume that the overall bandwidth assigned to the broadcast channels of a multicell WDB system is fixed (B=const [bps]) and partitioned equally among all the existing cells.

Now, if we use $r_{single\text{-}layer\text{-}WDB}$ to annotate the pico-cell broadcast channel datarate of the single-layer system from Figure 6.12.a, then for this system under the given fixed broadcast bandwidth constraint, (6.30) holds.

$$r_{\sin gle-layer-WDB} = \frac{B}{N}[bps] \tag{6.30}$$

Similarly, if we use r_{HWDB} to annotate the broadcast channel datarate of the hierarchical system from Figure 6.12.b, under the same fixed broadcast bandwidth constraint the following is obtained

$$r_{HWDB} = \frac{B}{N+1}[bps] \tag{6.31}$$

(Please note, we assume that the two systems of Figure 6.12 keep the same configuration at the pico-cell level. However, the hierarchical system in Figure 6.12.b has an additional overlaying micro-cell. Thus, if the overall number of cells in the system of Figure 6.12.a is N, then there are (N+1) cells in the system of Figure 6.12.b. Consequently, r_{HWDB} is always lower then the corresponding $r_{single\text{-}layer\text{-}WDB}$. Hence, the smaller overall number of cells in the system of Figure 6.12.a is being compensated through its higher broadcast channel datarate.)

Throughout the preceding sections, l(i)[sec]=*l*(i)[bits]/r[bps][21] was used to annotate item(i)'s length (i=1,..,M). By replacing r[bps] with (6.30), i.e. (6.31), the substitute expressions for l(i) in the system of Figure 6.12.a and Figure 6.12.b, is obtained

$$l(i)[sec]\Big|_{r = r_{single-layer-WDB}} = \frac{\mathit{l}(i)[bits]}{r_{single-layer-WDB}[bps]} = N\frac{\mathit{l}(i)}{B}[sec] \tag{6.32}$$

$$l(i)[sec]\Big|_{r = r_{HWDB}} = \frac{\mathit{l}(i)[bits]}{r_{HWDB}[bps]} = (N+1)\frac{\mathit{l}(i)}{B}[sec] \tag{6.33}$$

According to the discussion from Section 6.3, (6.34) represents the overall broadcast service ratio in a single-layer multicell WDB system, consisting of N pico-cells.

$$BR_{single-layer-WDB} = \frac{1}{\lambda^*(1-a)}\sum_{j=1}^{N}\sum_{i=1}^{M}\frac{p(i,j)}{s_{pico}(i,j)}(1-e^{-\lambda^*(1-a)s_{pico}(i,j)}) \tag{6.34}$$

Now, based on (6.34), and by employing the approximate closed from expressions for $S_{optimal}$ in a single-layer WDB system, i.e. multicell HWDB system, the exact expressions for the maximum broadcast service ratio in the two systems of Figure 6.12 are obtained (see (6.35) and (6.36)).

$$BR_{single-layer-WDB}(\tilde{S}_{optimal}) = \frac{1}{\lambda^*(1-a)}\sum_{j=1}^{N}\sum_{i=1}^{M}p(i,j)\frac{\overline{p}(i,j)}{l(i)}(1-e^{-\lambda^*(1-a)\frac{l(i)}{\overline{p}(i,j)}}) \tag{6.35}$$

[21] Please recall, r represents the corresponding broadcast channel datarate.

$$HBR(\breve{S}_{optimal}) = \frac{1}{\lambda^*(1-a)}\left[\sum_{i=1}^{L}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)}{dp}(1-e^{-\lambda^*(1-a)\frac{dp}{\overline{p}(i,j)}})\right)+ \right.$$
$$\left. + \sum_{i=L+1}^{M}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)}{l(i)}(1-e^{-\lambda^*(1-a)\frac{l(i)}{\overline{p}(i,j)}})\right)\right] \tag{6.36}$$

After employing the function approximation presented in Appendix 8, (6.35) and (6.36) get transformed to (6.37) and (6.38), respectively.

$$BR_{single-layer-WDB}(\tilde{S}_{optimal}) = \frac{k}{\left(\lambda^*(1-a)\right)^{(1-1/n)}}\sum_{j=1}^{N}\sum_{i=1}^{M} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(j)^{(1-1/n)}} \tag{6.37}$$

$$HBR(\breve{S}_{optimal}) = \frac{k}{\left(\lambda^*(1-a)\right)^{(1-1/n)}}\left[\sum_{i=1}^{L}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{dp^{(1-1/n)}}\right)+\sum_{i=L+1}^{M}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}}\right)\right] \tag{6.38}$$

By replacing l(i) (i=1,..,M) in (6.37) and (6.38) with the corresponding substitute expressions, i.e. (6.32) and (6.33), (6.39) and (6.40) are obtained.

$$BR_{single-layer-WDB}(\tilde{S}_{optimal}) = \frac{kB}{\left(\lambda^*(1-a)N\right)^{(1-1/n)}}\sum_{j=1}^{N}\sum_{i=1}^{M} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}} \tag{6.39}$$

$$HBR(\breve{S}_{optimal}) = \frac{kB}{\left(\lambda^*(1-a)(N+1)\right)^{(1-1/n)}}\left[\sum_{i=1}^{L}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{dp^{(1-1/n)}}\right)+ \right.$$
$$\left. + \sum_{i=L+1}^{M}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}}\right)\right] \tag{6.40}$$

The ratio of HBR($\breve{S}_{optimal}$) and $BR_{single-layer-WDB}(\tilde{S}_{optimal})$, under the constraint of fixed overall broadcast bandwidth, is annotated with R^* and presented in (6.41).

$$R^* = \frac{HBR(\breve{S}_{optimal})}{BR_{single-layer-WDB}(\widetilde{S}_{optimal})} = \tag{6.41}$$

$$= \left(\frac{N}{N+1}\right)^{(1-1/n)} \frac{\sum_{i=1}^{L}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{dp^{(1-1/n)}}\right) + \sum_{i=L+1}^{M}\left(\sum_{j=1}^{N} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}}\right)}{\sum_{j=1}^{N}\sum_{i=1}^{M} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}}}$$

$\sum_{j=1}^{N}\sum_{i=L+1}^{M} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}}$ appears both in the numerator and denominator of (6.41). By annotating this double summation with H, (6.41) becomes

$$R^* = \frac{HBR(\breve{S}_{optimal})}{BR_{single-layer-WDB}(\widetilde{S}_{optimal})} = \left(\frac{N}{N+1}\right)^{(1-1/n)} \frac{\sum_{j=1}^{N}\sum_{i=1}^{L} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{dp^{(1-1/n)}} + H}{\sum_{j=1}^{N}\sum_{i=1}^{L} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}} + H} \tag{6.42}$$

Based on (6.42), the following conclusions can be made.

(1) number of pico-cells is larage $\Rightarrow$ hierarchical WDB provides better throughput

$$\lim_{N\to\infty} R^* = \left.\frac{\sum_{j=1}^{N}\sum_{i=1}^{L} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{dp^{(1-1/n)}} + H}{\sum_{j=1}^{N}\sum_{i}^{L} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{l(i)^{(1-1/n)}} + H}\right|_{l(i) > dp} > 1 \tag{6.43}$$

In practical terms, (6.43) proves that by organizing a WDB system with a large number of pico-cells (N>>) and limited broadcast bandwidth (B=const) in the hierarchical manner of Figure 6.12.b, a higher throughput can be obtained than if all the pico-cells were kept independent (as long as l(i)>dp, i=1,..,M).

(2) data-times are significantly larger than data pointers ⇒ hierarchical WDB provides better throughput

$$(N+1)dp < Nl(i) \quad (i=1,..,M) \quad \Rightarrow \quad R^* = \frac{\sum_{j=1}^{N}\sum_{i=1}^{L} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{((N+1)dp)^{(1-1/n)}} + H}{\sum_{j=1}^{N}\sum_{i=1}^{L} p(i,j)\frac{\overline{p}(i,j)^{(1-1/n)}}{(Nl(i))^{(1-1/n)}} + H} > 1 \qquad (6.44)$$

Again, from the practical point of view, (6.44) proves that the overall throughput of an arbitrary number of independent WDB cells that occupy a fixed broadcast bandwidth (B=const) can be improved simply by organizing the cells in the hierarchical manner of Figure 6.12.b, provided data pointers placed at the pico-cell level are at least (N+1)/N times smaller than the corresponding items.

6.4.2 Performance Comparison under Maximum Broadcast Service Ratio per Broadcast Bandwidth Cost Constraint

In the previous section the performance of a multicell WDB system under two different bandwidth allocation policies has been studied. (According to one policy the entire broadcast bandwidth is employed at the pico-cells level, while the other policy assumes that a fraction of the bandwidth gets allocated to the micro-cell level.) No special cost associated with the bandwidth, i.e. broadcast channel, utilization at the micro-cell level has been considered.

However, from the system point of view, there is a significant difference between the utilization of a broadcast channel at the pico- and micro- cell level. Namely, as already described in Section 6.1, micro-cells are intended to provide coverage in physically large areas, and therefore they employ antenna-transmitters of higher-power than their underlying pico-cells. This implies that the broadcasting of an item at the micro-cell level should be considered more costly, with respect to the system's energy/power consumption and bandwidth demand, than if the same item was provided through a pico-cell broadcast channel. Hence, when comparing the performance of two different mutlicell WDB systems, one has to consider both, the effectiveness of their service (i.e. their broadcast service ratio) and the associated service cost. Accordingly, the system capable of providing a higher broadcast service ratio at a lower broadcast bandwidth cost is assumed to be of a more favourable performance.

In order to define broadcast service ratio per broadcast bandwidth cost for the two multi-cell WDB systems presented in Figure 6.12, let us introduce the following notation:

1) pico-cell broadcast channel datarate: r_{pico} [bps]
2) micro-cell broadcast channel datarate: r_{micro} [bps]
3) utilization cost per pico-cell broadcast bandwidth unit: CB_{pico} [1/bps]
4) utilization cost per micro-cell broadcast bandwidth unit: CB_{micro} [1/bps]
5) overall broadcast bandwidth cost: $CBB=r[bps]\cdot CB[1/bps]$
6) broadcast bandwidth cost ratio: $A=CB_{micro}/CB_{pico}\geq 1$
7) broadcast service ratio per broadcast bandwidth cost: BRBC=BR/CBB

The single-layer multicell WDB system of Figure 6.12.a consists of N independent pico-cells only. Thus, the overall broadcast bandwidth utilization cost in this system is

$$CBB_{single-layer-WDB} = N \cdot CB_{pico} \cdot r_{pico} \tag{6.45}$$

On the other hand, the hierarchical WDB system of Figure 6.12.b, as explained in Sections 6.2, consists of N pico-cells and an overlaying micro cell. Thus, the overall broadcast bandwidth cost of this system is as given in (6.46).

$$CBB_{HWDB} = N \cdot CB_{pico} \cdot r_{pico} + CB_{micro} \cdot r_{micro} = CB_{pico} \cdot (N \cdot r_{pico} + A \cdot r_{micro}) \tag{6.46}$$

Optionally, we can also write,

$$\mathrm{CBB}_{\mathrm{HWDB}} = \mathrm{CBB}_{\mathrm{single-layer-WDB}} + \mathrm{CB}_{\mathrm{micro}} \cdot \mathrm{r}_{\mathrm{micro}} \geq \mathrm{CBB}_{\mathrm{single-layer-WDB}} \tag{6.47}$$

Based on (6.37) and (6.45), (6.38) and (6.46), the expression concerning the maximum broadcast service ratio per broadcast bandwidth cost of the single-layer system and the hierarchical system are presented in (6.48), and (6.49), respectively.

$$\mathrm{BRBC}_{\mathrm{single-layer-WDB}} = \frac{\mathrm{BR}_{\mathrm{single-layer-WDB}}(\tilde{\mathrm{S}}_{\mathrm{optimal}})}{\mathrm{CBB}_{\mathrm{single-layer-WDB}}} = \tag{6.48}$$

$$= \frac{\mathrm{k}\cdot(\mathrm{r}_{\mathrm{pico}})^{(1-1/\mathrm{n})}}{\mathrm{N}\cdot\mathrm{CB}_{\mathrm{pico}}\cdot\mathrm{r}_{\mathrm{pico}}\cdot(\lambda^{*}(1-\mathrm{a}))^{(1-1/\mathrm{n})}} \sum_{j=1}^{N}\sum_{i=1}^{M}\mathrm{p}(\mathrm{i},\mathrm{j})\frac{\overline{\mathrm{p}}(\mathrm{i},\mathrm{j})^{(1-1/\mathrm{n})}}{l(\mathrm{i})^{(1-1/\mathrm{n})}}$$

$$\mathrm{BRBC}_{\mathrm{HWDB}} = \frac{\mathrm{HBR}(\breve{\mathrm{S}}_{\mathrm{optimal}})}{\mathrm{CBB}_{\mathrm{HWDB}}} = \tag{6.49}$$

$$= \frac{\mathrm{k}\cdot(\mathrm{r}_{\mathrm{pico}})^{(1-1/\mathrm{n})}}{\mathrm{CB}_{\mathrm{pico}}\cdot(\mathrm{N}\cdot\mathrm{r}_{\mathrm{pico}}+\mathrm{A}\cdot\mathrm{r}_{\mathrm{micro}})\cdot(\lambda^{*}(1-\mathrm{a}))^{(1-1/\mathrm{n})}}\left[\sum_{i=1}^{L}\sum_{j=1}^{N}\mathrm{p}(\mathrm{i},\mathrm{j})\frac{\overline{\mathrm{p}}(\mathrm{i},\mathrm{j})^{(1-1/\mathrm{n})}}{dp^{(1-1/\mathrm{n})}} + \sum_{i=L+1}^{M}\sum_{j=1}^{N}\mathrm{p}(\mathrm{i},\mathrm{j})\frac{\overline{\mathrm{p}}(\mathrm{i},\mathrm{j})^{(1-1/\mathrm{n})}}{l(\mathrm{i})^{(1-1/\mathrm{n})}}\right]$$

The ratio of $\mathrm{BRBC}_{\mathrm{single-layer-WDB}}$ and $\mathrm{BRBC}_{\mathrm{HWDB}}$ is annotated with R^{**}, and presented in (6.50).

$$\mathrm{R}^{**} = \frac{\mathrm{BRBC}_{\mathrm{HWDB}}}{\mathrm{BRBC}_{\mathrm{single-layer-WDB}}} = \frac{\mathrm{N}\cdot\mathrm{r}_{\mathrm{pico}}}{\mathrm{N}\cdot\mathrm{r}_{\mathrm{pico}}+\mathrm{A}\cdot\mathrm{r}_{\mathrm{micro}}}\cdot\frac{\sum_{j=1}^{N}\sum_{i=1}^{L}\mathrm{p}(\mathrm{i},\mathrm{j})\frac{\overline{\mathrm{p}}(\mathrm{i},\mathrm{j})^{(1-1/\mathrm{n})}}{dp^{(1-1/\mathrm{n})}}+\mathrm{H}}{\sum_{j=1}^{N}\sum_{i=1}^{L}\mathrm{p}(\mathrm{i},\mathrm{j})\frac{\overline{\mathrm{p}}(\mathrm{i},\mathrm{j})^{(1-1/\mathrm{n})}}{l(\mathrm{i})^{(1-1/\mathrm{n})}}+\mathrm{H}} \tag{6.50}$$

Here, for analytical purposes, let us assume equal length broadcast items ($l(i)=l(j)$, $\forall i,j=1,..,M$). Consequently, through a simple rearrangement of (6.50), (6.51) is obtained.

$$R^{**} = \frac{BRBC_{HWDB}}{BRBC_{single-layer-WDB}} = \frac{1}{1+\frac{A \cdot r_{micro}}{N \cdot r_{pico}}} \cdot \left(\frac{l}{dp}\right)^{(1-1/n)} \tag{6.51}$$

(Please note, in (6.51) r_{micro}/Nr_{pico} can be interpreted as the ratio of the overall broadcast bandwidth employed at the micro- and the overall broadcast bandwidth employed at the pico-level in the hierarchical system of Figure 6.12.b. Also, please recall that $A=CB_{micro}/CB_{pico}$ in (6.51) represents the broadcast bandwidth cost ratio, and according to the assumptions from the beginning of this section, $A \geq 1$.)

The practical significance of (6.51) can be illustrated through the analysis of the following special-case scenarios.

(1) R^{**} as a function of micro/pico cell bandwidth cost

Let us assume that a given hierarchical WDB system consists of a fixed number of pico-cells (N), the size of a broadcast channel is the same regardless of its position in the hierarchy ($r_{pico}=r_{micro}$), and the data-item size is k times larger than the data-pointer size ($l=k \cdot dp$). Under these general assumptions, (6.51) enables us to observe at what value of micro/pico cell bandwidth cost ratio (CB_{micro}/CB_{pico}) hierarchical WDB ensures a better performance than single-layer WDB, in terms of R^{**} (see Figures 6.13 and 6.14).

For example, Figure 6.13 shows that in a multi-cell wireless system with N=6 and $l=5 \cdot dp$ hierarchically organized WDB will provide better performance than single-layer WDB, with respect to R^{**}, as long as $A=C_{micro}/C_{pico} \leq 20$, i.e. the cost of micro-cell broadcast bandwidth does not exceed the cost of pico-cell broadcast bandwidth over twenty times (see Figure 6.13). However, if $l=10 \cdot dp$, the same holds true for any $A \leq 40$ (see Figure 6.14).

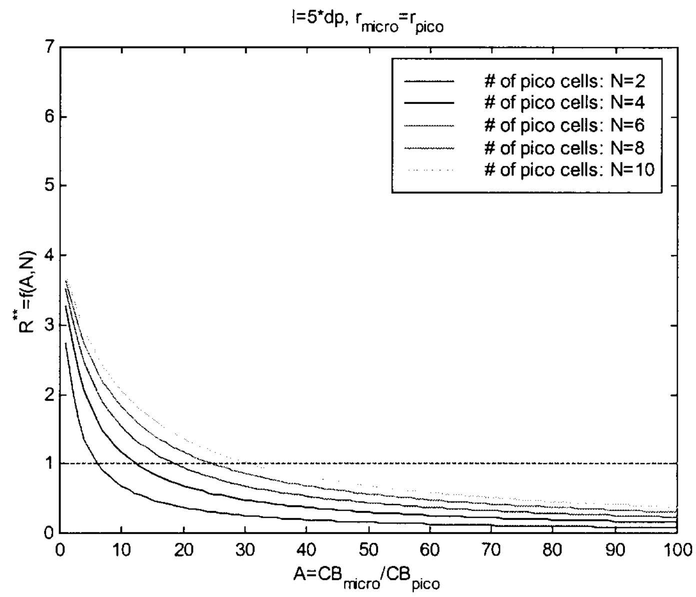


Figure 6.13 R^{**} as a function of micro/pico bandwidth cost ratio, for $r_{micro}=r_{pico}$ and $l=5 \cdot dp$

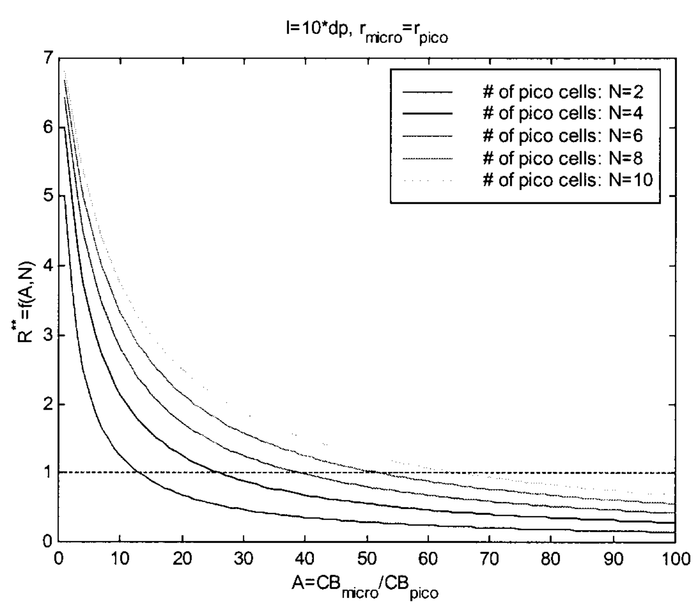


Figure 6.14 R^{**} as a function of micro/pico bandwidth cost ratio, for $r_{micro}=r_{pico}$ and $l=10 \cdot dp$

(2) R^{**} as a function of micro-cell broadcast bandwidth utilization

Now, let us assume that a given hierarchical WDB system consists of a fixed number of pico-cells (N), the data-item size is k times larger than the data-pointer size (l=k·dp), and the bandwidth cost ratio (A) is known. Furthermore, let us assume that it is possible to utilize only a fraction of the overall micro-cell broadcast bandwidth (r_{micro}) for the purpose of data-item delivery.[22] Under these assumptions, (6.51) enables us observe what fraction of the micro-cell broadcast bandwidth would be reasonable to utilize, under a specific value of A, so as to ensure a better performance of hierarchical over single-layer WDB, in terms of R^{**} (see Figure 6.15).

Figure 6.15 shows that under N=6, l=5·dp, and A=35, it is reasonable to utilize up to 50% of the nominal micro-cell broadcast bandwidth for the purpose of data-item delivery. At the same time, this number (i.e. percentage) increases as the cost of micro-cell broadcast bandwidth (i.e. the value of A) decreases. Thus, at A=20, the utilization up to 90% of the nominal micro-cell broadcast bandwidth is justifiable.

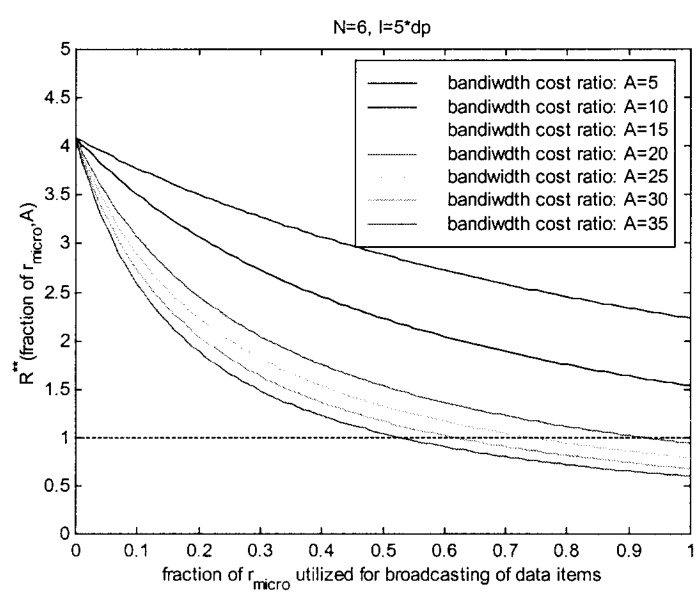


Figure 6.15 R^{**} as a function of utilized micro-cell broadcast bandwidth, for N=6 and l=5·dp

[22] Such a scenario would correspond to Cell Broadcast Service (see Section 1.3.1) in a hierarchical multi-cellular system. Namely, in CBC systems the broadcast channel is used for the deliver of: 1) system control information, and 2) common information that is in high demand by the users.

6.5 Summary of Main Contributions

In this chapter, by extending on the ideas from Chapter 2 and 3, first we have proposed a model of a hybrid hierarchical WDB (HWDB) system. Consequently, we have shown that the search for $S_{optimal}$ aimed at maximizing the throughput of such a system is a constrained optimization problem with multiple degrees of complexity[23] and without a tractable closed form solution. (Namely, in order to find the ultimate $S_{optimal}$, first the optimal grouping of those items that get broadcast at the pico- and micro- cell level has to be devised, i.e. L items that are worth of broadcasting at the micro-cell level need to be identified. Subsequently, the optimal broadcast periods of items and data-pointers that get to be broadcasted at the pico-cell level need to be found.)

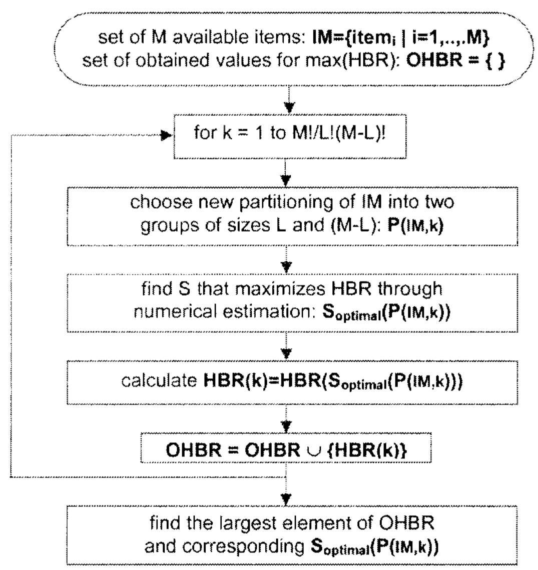


Figure 6.14 Steps involved in the numerical estimation of optimal broadcast schedule for an HWDB system

[23] In this chapter, we have employed the assumption that a predefined number of items get broadcasted at the macro-cell level. However, the given assumption can be relaxed, and the number of macro-cell broadcast items can be considered as another optimization factor.

Since the analytical approach to $S_{optimal}$ fails, numerical estimation of $S_{optimal}$ remains as an option (see Figure 6.14). Nevertheless, due to the nature of the given problem, the search for $S_{optimal}$ through numerical estimation turns out to be computationally rather expensive.

As an illustration of the actual complexity of the numerical estimation approach, let us look at the following example: M=10 items need to be broadcasted in an HWDB system. Due to the higher utilization cost of the micro-cell broadcast bandwidth, only L=3 items can be accepted for broadcasting at the micro-cell level. Clearly, for such L and M, there are M!/L!(M-L)!=120 possible ways of partitioning the items on those that get broadcasted at the pico- and micro-level. Hence, in order to obtain the ultimate $S_{optimal}$, one would have to find a numerical solution to (6.6) for each of the 120 combinations, and then select the combination (i.e. the corresponding S) that provides the highest value of HBR. Realization that items to be broadcasted will be in the range of hundreds and latter thousands make the impracticality of the numerical solution obvious.

Fortunately, we have shown that by exploiting our earlier obtained approximate closed-form expression concerning $S_{optimal}$ in single-cell WDB systems, the problem of finding $S_{optimal}$ in HWDB systems can be considerably simplified. Consequently, it becomes feasible to obtain both, the optimal layer-based item grouping and the optimal item scheduling within each micro-cell, in a single round of computation.

In general, the operational complexity/cost of multicell HWDB systems is expected to exceed the complexity/cost of their single-tier counterparts. However, our comparative analysis has shown that in a wide range of scenarios, HWDB systems have the potential to provide considerably superior performance (in terms of the overall throughput, QoS and Gos) over the performance of single-layer WDB systems. Hence, the higher operational complexity/cost of multicell HWDB systems turns out to be more than justifiable.

Chapter 7

Simulation Software

Overview

In this chapter, we introduce DESMO-J – a Java-based discrete-event simulation package. Also, we provide an overview of our simulation software, which has been built on the DESMO-J framework and enabled us to simulate the performance of single-cell hybrid WDB systems. The simulation results obtained using this software have been presented in Chapter 2 (see Section 2.6).

7.1 Overview of DESMO-J

A framework[1] for Discrete Event[2] Simulation and Modeling, named DESMO, was developed at the University of Hamburg in the late 1980s. It has been continuously evolving over the last decade. Starting with Modula-2, the framework was ported to various programming languages including Small-talk and C++. The latest version is DESMO-J, presented by Lechler and Page in 1999. It is based on a fully object-oriented architecture and is completely implemented in Java [*Gehlsen*]. Moreover, DESMO-J supports both event-oriented[3] and process-oriented[4] modeling, thus enabling a modeler to choose the most suitable approach to represent individual model elements. (The source code and a Java archive can be found on the DESMO-J Web-site (http://www.desmoj.de), together with a complete API documentation and an extensive tutorial.)

By choosing Java as a popular object-oriented programming language for the DESMO framework, a large and rapidly growing community of programmers is addressed. Java offers portability across diverse platforms, allowing the execution of Java based simulation experiments on most operating systems available. Its ease of use and robustness enable the user to shift focus on the modeling issue rather than implementation trapholes, such as dangling pointers and memory leaks likely to be encountered using C++ [*Lechler*].

---

[1] In general, the term 'framework' refers to basic algorithms and a suitable software architecture for solving problems of a defined application area. Hence, if a framework is available, in order to get a running application, the user just has to add or overload some classes or interfaces of the given framework according to the specific application demands, without having to reinvent the functionality over and over again.

[2] In discrete event simulation, a system is modeled in terms of its state at each point in time (the entities that pass through the system and the entities that represent system resources), and the activities and events that cause system state to change. Discrete-event models are appropriate for those systems for which changes in system state occur only at discrete points in time [*Banks*].

[3] In the case of event-oriented approach, a simulation analyst concentrates on events and their effect on system state [*Banks*].

[4] When using process-oriented modeling style, a simulation analyst thinks in terms of processes. In particular, the analyst defines the simulation model in terms of entities or objects and their life cycle as they flow through the system, demanding resources and queueing to wait for resources [*Banks*].

Being implemented in an object-oriented manner, using Java, the architecture of the DESMO-J framework is based on classes and makes use of inheritance. DESMO-J provides the user with the standard functionality for discrete simulation such as a scheduler controlling a list of future events, a simulation clock, a flexible messaging subsystem for trace-, debug-, report- and error-output, queues and a series of random distribution functions for use in stochastic models. These are the parts that can always be reused when implementing a discrete simulation program. DESMO-J also provides some abstract classes, which the modeler can use to add the model-specific behaviour to the framework [*Gehlsen*].

The basic classes to represent the entities, events and processes in a DESMO-J simulator are Entity, Event, SimProcess, and ExternalEvent. (Figure 1 shows the inheritance relation among these classes.) The class Schedulable merely collects the functionality common to all the three classes: 1) to cancel a scheduled event, entity or process, 2) move it within the event-list reusing reSchedule, or 3) give an answer if and at which point of the future simulation time an entity, event or process is scheduled [*Gehlsen*].

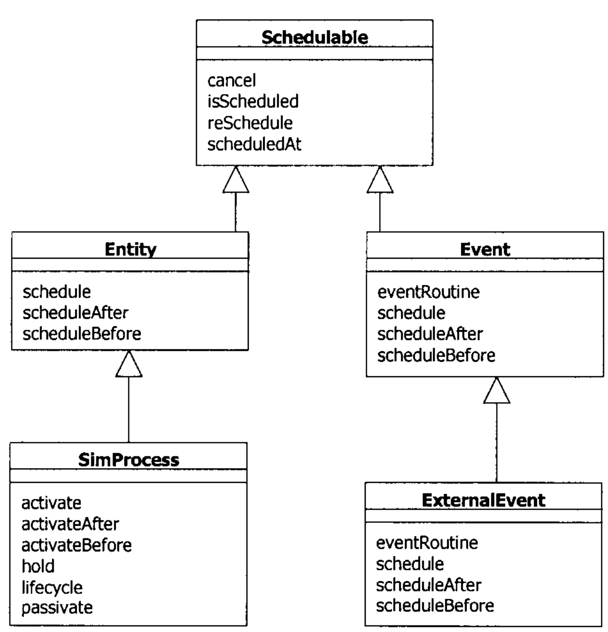


Figure 7.1 UML class-diagram of schedulable DESMO-J objects

When taking the event oriented perspective, there is always an object of class Event related to an object of class Entity. The event's method eventRoutine(Entity e) carries out the actual changes to the state of a given entity and the model in general. A pure event oriented model will have the scheduler in a loop invoking the events according to their temporal order on the list of future events.

The class ExternalEvent, derived from class Event, is primarily used by the framework itself to control 'external' events like switching trace-output on and off at certain points of simulation time. Unlike objects of class Event, they are not related to any particular entity and thus offer a parameterless method eventRoutine() to have the specific actions implemented.

A process has to be treated differently. The actions it performs during its existence in the model can be described by the modeler when implementing the parameterless and abstract method lifeCycle(). Te process' lifecycle can be controlled through the methods like activate(), hold() and passivate(). In a pure process oriented model the scheduler's task is to activate the next process waiting in the list of future activations and hand control over to the process' lifecycle until that process has finished its lifecycle or passivates itself again after finishing another section of its lifecycle. Control is then given back to the scheduler, which resumes its loop to pick the next process to be activated again.

7.2 "WDB-System" Simulation Software

The simulation software that we have developed and used in the analysis of single-cell hybrid WDB systems, has been based on the DESMO-J framework and entirely implemented in Java. The software consists of four main classes:

1) ***MVgenerator class*** (extends the DESMO-J's ExternalEvent class)
2) ***BroadcastServer class*** (extends the DESMO-J's ExternalEvent class)
3) ***MV class*** (extends the DESMO-J's SimProcess class)
4) ***SingleCellModel class*** (subclass of DESMO-J's Model class)

The first three classes are the core of the software - they implement the actual functionality, i.e. behaviour, of the single-cell hybrid WDB system and its users. (The UML and state diagram representations of these classes are shown in Sections 7.2.1 to 7.2.3, respectively.) The fourth class, on the other hand, provides a user interface to the software and enables the instantiation of all the other classes, i.e. objects. (The UML representation of this class is provided in Section 7.2.4).

7.2.1 Mobile User Generator (MUgenerator) Class

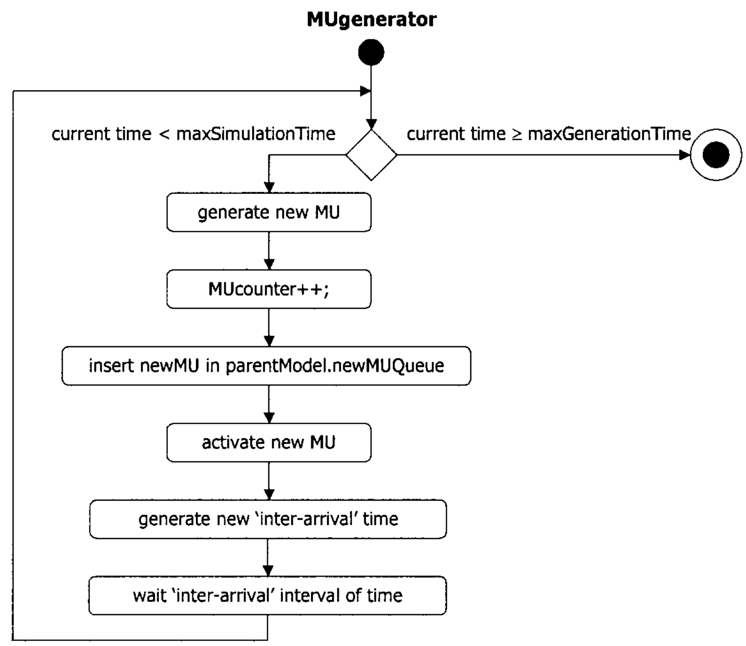


Figure 7.2 eventRoutine() of MUgenerator class

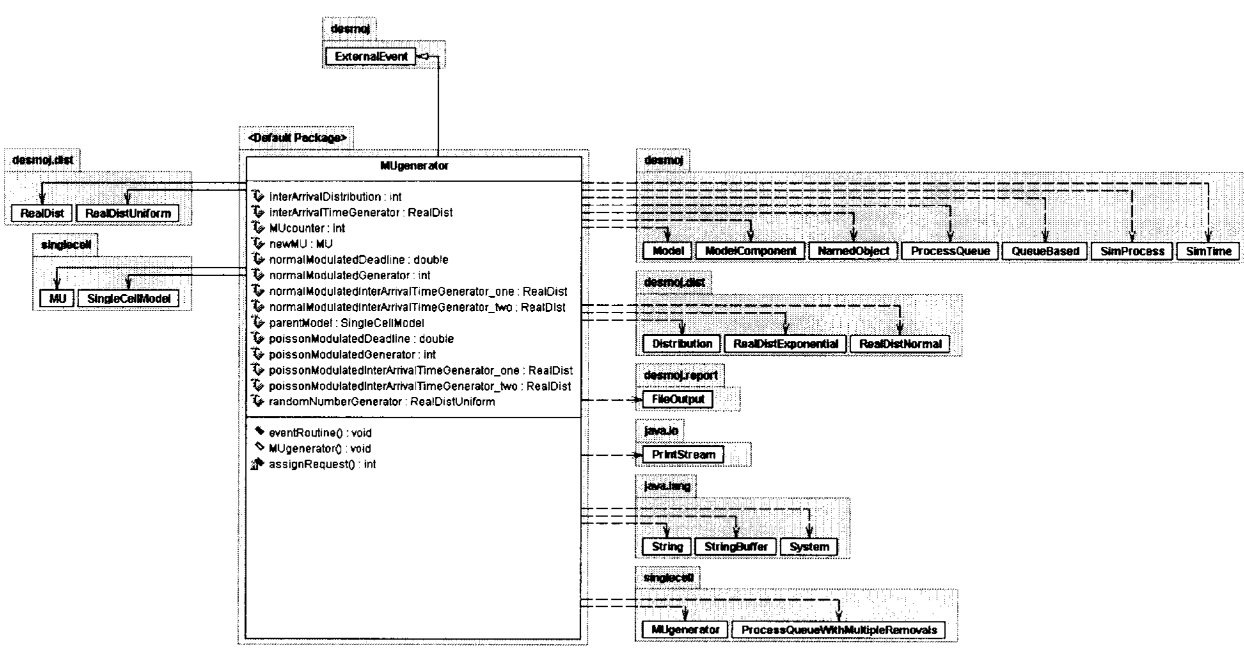


Figure 7.3 UML structure of MUgenerator class

7.2.2 BroadcastServer Class

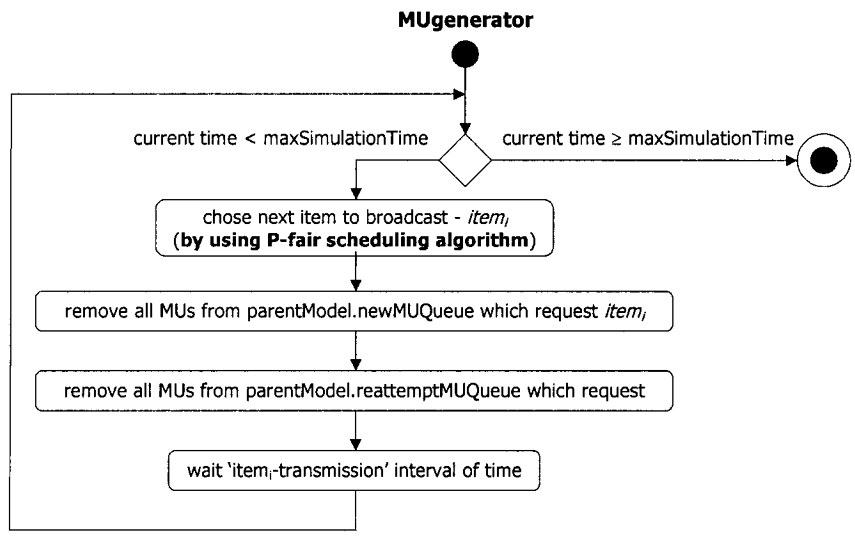


Figure 7.4 eventRoutine() of BroadcastServer class

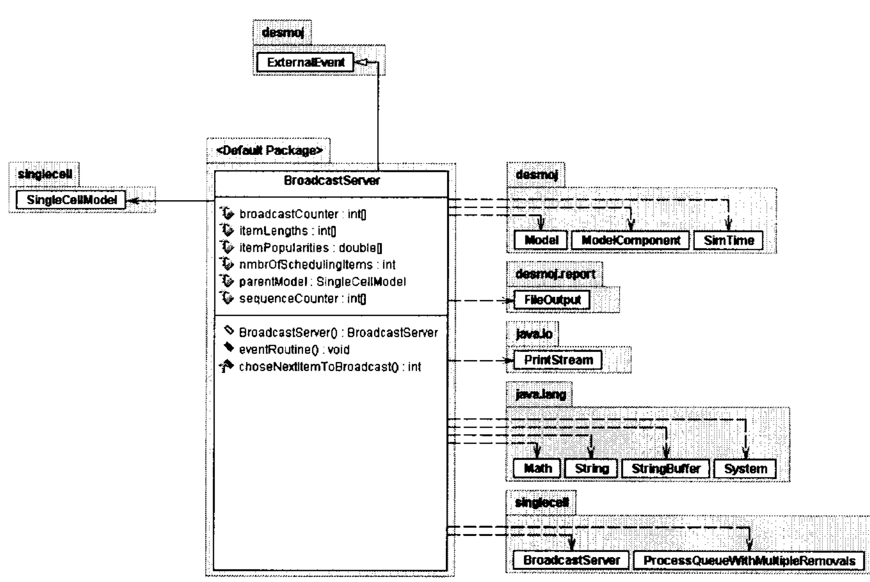


Figure 7.5 UML structure of BroadcastServer class

7.2.3 Mobile User (MU) Class

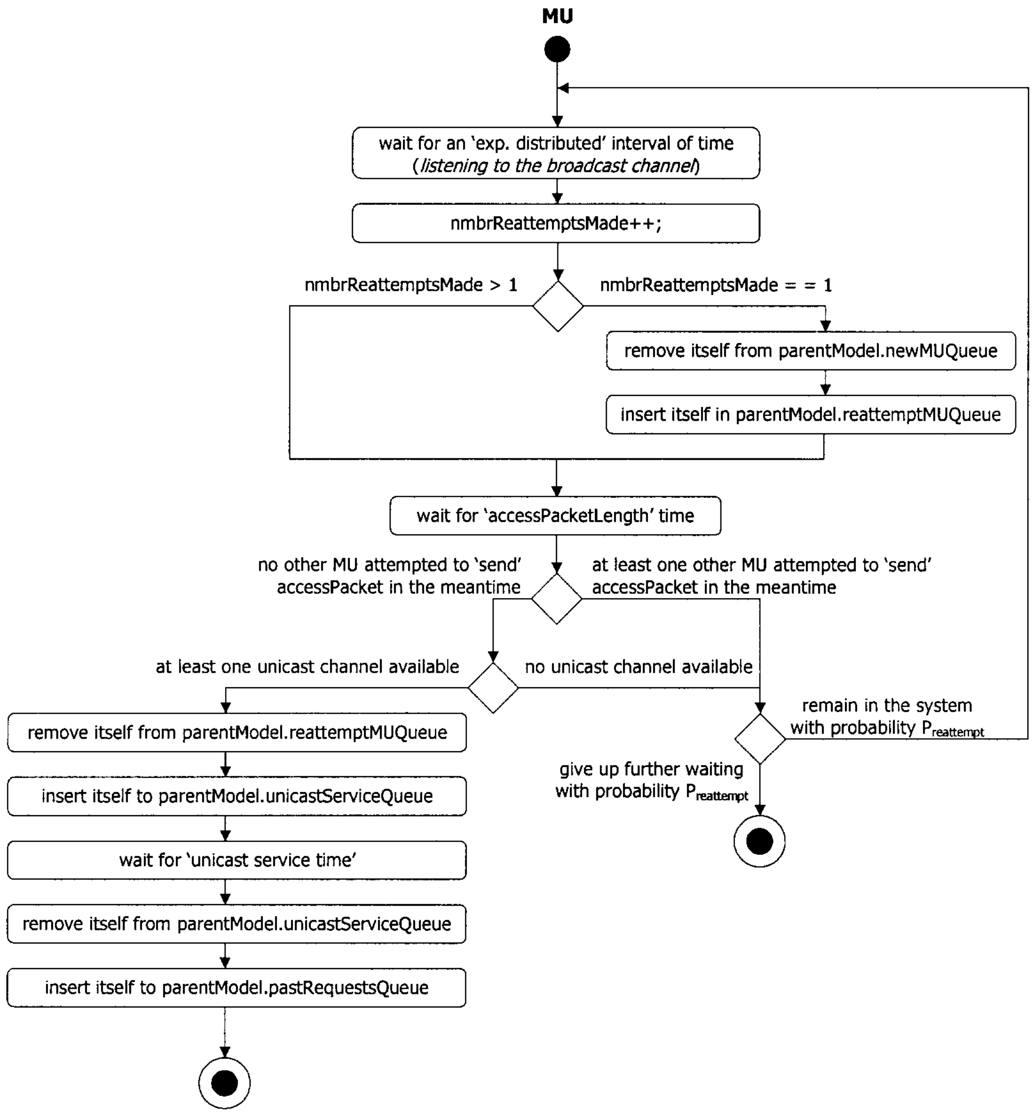


Figure 7.6 lifeCycle() of MU class

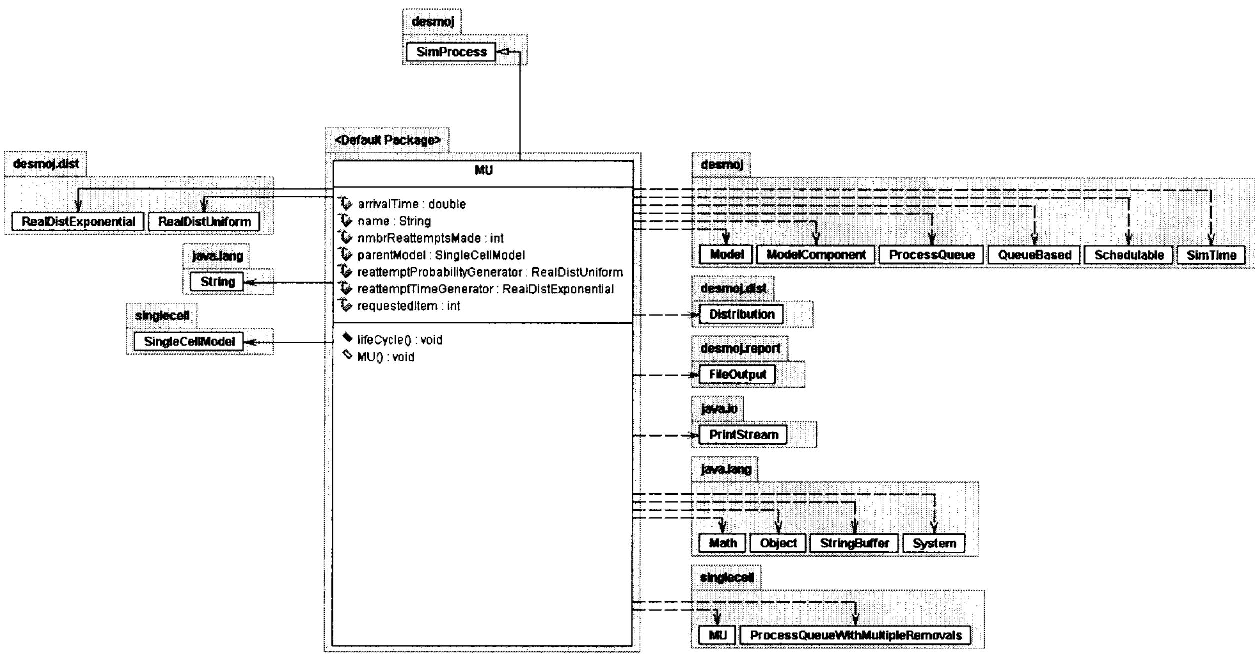


Figure 7.7 UML structure of MU class

7.2.4 SingleCellModel Class

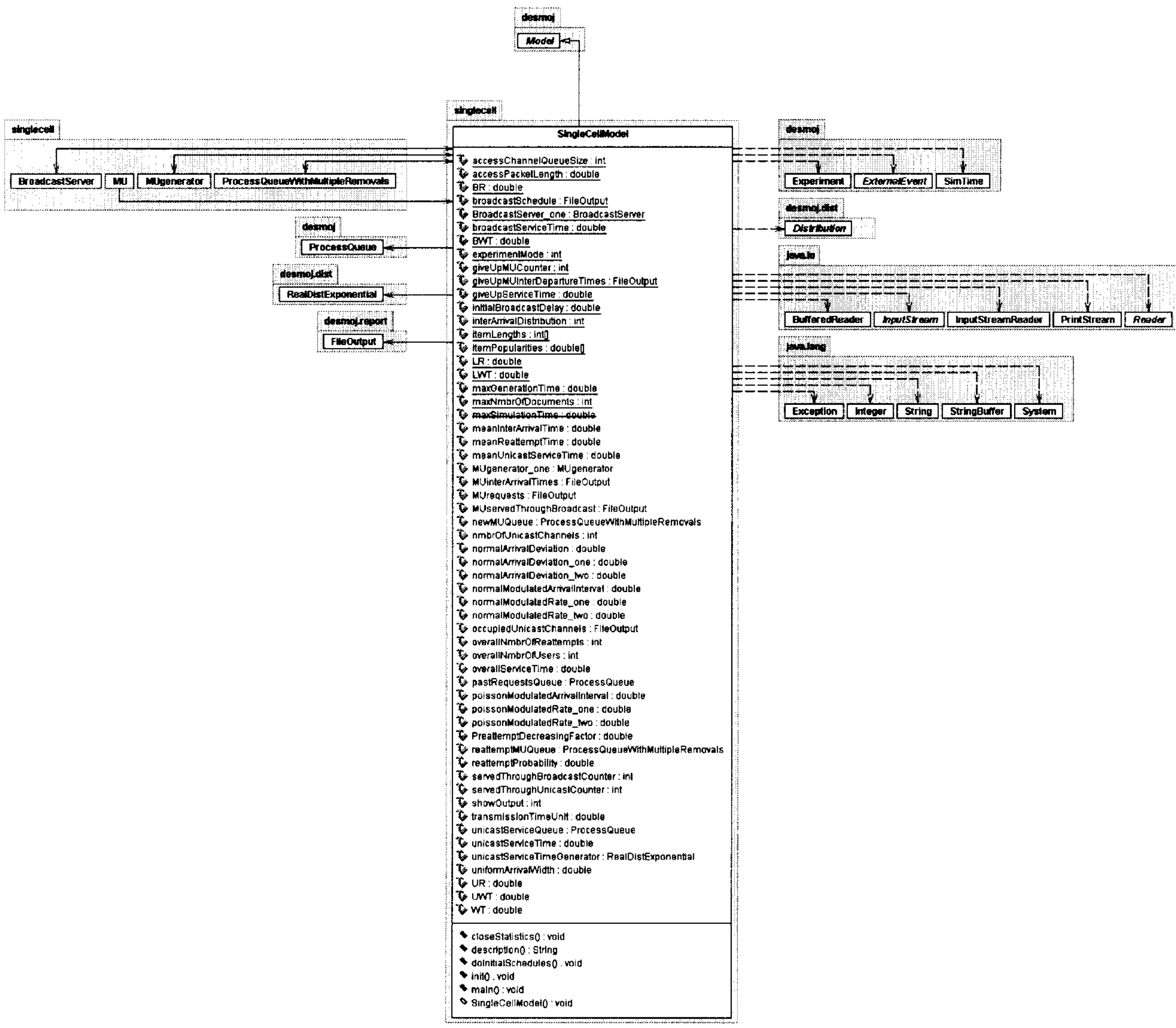


Figure 7.8 UML structure of SingleCellModel class

Future Work

With the work presented in this thesis, we attempted to provide a complete framework for the analysis of single-cell and multicell-hierarchical WDB systems. Nevertheless, we recognize that there is still some room for improvement, and we identify the following problems as interesting topics for future research:

(1) *Calculation of Performance Measures for Environments with Constraints*

The performance measures proposed and studied in this thesis (see Section 2.3) are appropriate for the analysis of a general class of mobile/wireless WDB-based systems. However, it would be interesting to explore some other performance measures, suited for environments with specific constraints. For example, in the case of mobile/wireless WDB-based systems populated with small battery-powered devices, *expected number of request re-transmissions per device/user* (RT) could be a useful performance measure. Namely, transmission attempts made by a wireless device are known to be a major contributor to its overall battery power consumption. Accordingly, RT could be used as an implicit indicator of the system's effectiveness in reducing the average power consumption per device/user.

(2) *Analysis of Hierarchical WDB Systems under 'Fixed Macro-Cell Broadcast Bandwidth' Constraint*

In the analysis of hierarchical WDB systems, we have assumed that a fixed number of items get broadcasted at the micro-cell level. However, another (very realistic) assumption/constraint could be employed instead - *occupancy of a fixed macro-cell broadcast bandwidth.* Unfortunately, such an assumption would introduce additional complexity to the problem of optimal broadcast scheduling scheme estimation. Namely, under the current assumption of a fixed number (L) of data-items being broadcasted at the macro-cell level, there are M!/L!(M-L)! layer-based item groupings that need to be examined for optimality (see Figure 6.14). On the other hand, under the assumption of a fixed macro-cell broadcast bandwidth (B), L would become a subject to optimization itself. Consequently, a much larger number of layer-based item groupings would need to be examined for optimality - in particular M!/L!(M-L)! (L=1,..,M) of them. It is to believe, however, that even with L as an optimization variable, it still might be possible to come up with an approach for a fast identification of the optimal layer-based data-item partitioning (similar as in the case studied in this thesis).

(3) Optimization of Waiting Time (WT) in Hierarchical WDB Systems

In the case of a HWDB system, we have defined $S_{optimal}$ as the broadcast scheduling scheme capable of maximizing the system's BR. Through the analogy with a single cell WDB-based system, it can be easily proven that the same $S_{optimal}$ also minimizes UR, LR, RP, WP. However, the same analogy does not apply to WT. Namely, WT in HWDB systems depends not only on the broadcast scheduling scheme employed at the pico-cell level, but also on the actual scheduling at the micro-cell level. Thus, it would be interesting to find the exact expression for WT in HWDB systems, and investigate under which condition the minimization of WT can be obtained.

Appendix 1

Optimality of 'Equal-Spacing Broadcast'

Let us assume that, in a single-cell hybrid WDB based system (as presented in Chapter 2), the broadcast channel bandwidth allocation is based on the following principle: *the bandwidth assigned to broadcasting of item(i), within the proceeding interval of fixed duration C*[i]*, is in proportion to the current item(i)'s popularity (p(i)), i=1,..,M.* Let us annotate this bandwidth with b(i)=B(i)[ii]/C [bits/time units]. Clearly, under the given assumption, (A1.1) has to hold.

$$\sum_{i=1}^{M} b(i) = b \tag{A1.1}$$

(In (A1.1) b [bits/time units] represents the entire broadcast channel bandwidth.) Furthermore, let us assume that item(i) is of a finite length l(i) [bits]. Hence, according to the given bandwidth allocation policy, item(i) gets to be broadcasted exactly n(i) times (see (A1.2)) within the observed C.

$$n(i) = \frac{B(i)}{l(i)} \tag{A1.2}$$

Consequently, if we annotate the duration of the sub-interval between the k^{th} and $(k+1)^{th}$ broadcasting of item(i) with s(i,k) ($1 \le k \le n(i)$), see Figure A1.1, then the following constraint must be satisfied

---

[i] For the precise definition of C, i.e. the so-called entire broadcast schedule, see Section 2.2.
[ii] Here, B(i) represents the bandwidth assigned to broadcasting of item(i), expressed in 'bit' units.

$$f(s(i,1), s(i,2),.., s(i,n(i))) = \sum_{k=1}^{n(i)} s(i,k) - C = 0 \qquad \text{(A1.3)}$$

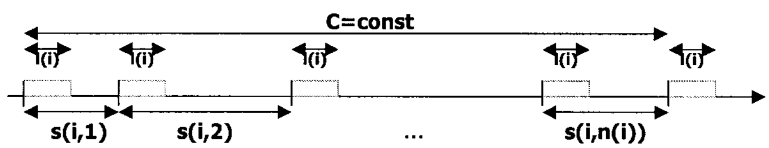


Figure A1.1 Arbitrary-spacing broadcast

(Please note, in general $s(i,k) \neq s(i,j)$ for $k \neq j$, $1 \leq k,j \leq n(i)$.) As n(i) in (A1.3) depends on the actual bandwidth allocation policy, we refer to (A1.3) as *bandwidth allocation constraint associated with item(i)*.

Based on (2.31), the number of users, interested in retrieving item(i), who get served through the broadcast channel at the end of s(i,k) ($1 \leq k \leq n(i)$) and within the entire C are given in (A1.4) and (A1.5), respectively.

$$E[n_{BR\_s(i,k)}(0, s(i,k))] = \frac{\lambda p(i)}{\lambda^*(1-a)}(1 - e^{-\lambda^*(1-a)s(i,k)}) \qquad \text{(A1.4)}$$

$$E[n_{BR}(0, C) \mid item = i] = \sum_{k=1}^{n(i)} \frac{\lambda p(i)}{\lambda^*(1-a)}(1 - e^{-\lambda^*(1-a)s(i,k)}) \qquad \text{(A1.5)}$$

Thus, the set of inter-broadcasting intervals associated with item(i) ($S(i)=\{s(i,k) \mid k=1,..,n(i)\}$) can be considered optimal, if it maximizes the number of users served through broadcasting within the entire C (A1.5), while satisfying the bandwidth allocation constraint associated with item(i), as given in (A1.3).

Theorem A1.1 If a given bandwidth allocation policy requires that item(i) be broadcasted n(i) times within C [time units], then the optimal set of inter-broadcasting intervals associated with this item ($S_{optimal}(i)$) is

$$S_{optimal}(i)=\{s(i,k)|\ s(i,k)=s(i,j)=C/n,\ \forall\ k,j=1,..,n(i)\},\quad i=1,..,M.$$

(In practical terms, Theorem A1.1 states that $S_{optimal}(i)$ consists of equal-length inter-broadcasting intervals.)

Proof of Theorem A1.1

Based on the Lagrange Multiplier theory [*Bertsekas*], $S_{optimal}(i)=\{s(i,k)|\ k=1,..,n\}$ that corresponds to the constrained maximum of (A1.5) (assuming the constraint is given in (A1.3)), can be obtained as the solution to the following set of differential equations.

$$\frac{\partial E[n_{BR}(0,C)\,|\,item=i]}{\partial s(i,k)} - K\frac{\partial f(s(i,1),s(i,2),..,s(i,n(i))}{\partial s(i,k)} = 0,\qquad k=1,..,n(i) \qquad (A1.6)$$

(K in (A1.6) is an unknown scalar, i.e. the so-called Lagrange Multiplier.)

The solution to (A1.6) can be obtained through the proceeding steps:

First, the exact expressions for $E[n(sm=1,t_a\in[0,C),item=i)]$ and C are employed in (A1.6) ((A1.5) and (A1.3) respectively), and (A1.6) is transformed to

$$\lambda p(i)e^{-\lambda^*(1-a)s(i,k)} - K = 0, \quad k = 1,..,n(i) \tag{A1.7}$$

Accordingly,

$$s(i,k) = \frac{1}{\lambda^*(1-a)} \ln\frac{\lambda p(i)}{K}, \quad k = 1,..,n(i) \tag{A1.8}$$

However, K in (A1.8) is unknown, thus the closed form expression for this parameter needs to be found, in order to obtain the exact expression for s(i,k) (k=1,..,n(i)). This can be accomplished by employing (A1.8) back in (A1.3) (see (A1.9) and (A1.10)).

$$\sum_{k=1}^{n(i)} s(i,k) = \sum_{k=1}^{n(i)} \frac{1}{\lambda^*(1-a)} \ln\frac{\lambda p(i)}{K} = n(i)\cdot\frac{1}{\lambda^*(1-a)} \ln\frac{\lambda p(i)}{K} = C \tag{A1.9}$$

$$K = \lambda p(i)e^{-\frac{C\lambda^*(1-a)}{n(i)}} \tag{A1.10}$$

Subsequently, by replacing K in (A1.8) with the expression provided in (A1.10), the final solution to (A1.6) is obtained.

$$S_{optimal}(i) = \{s(i,k) \mid s(i,k) = \frac{C}{n(i)}, \forall\, k = 1,..,n(i)\} \tag{A1.11}$$

Clearly, the elements of $S_{optimal}(i)$ are all of equal length. Thus, the optimality of 'equal-spacing broadcast' is proved.

End of Proof

Appendix 2

Random Splitting of Poisson Process

Definition A2.1 A continuous-time stochastic process $\{N(t) \mid t \geq 0\}$ is said to be a Poisson process with rate $\lambda > 0$ if the following is satisfied

(1) $N(0) = 0$;

(2) $N(t)$ has independent and stationary increments, i.e.

$P\{N(t_2) - N(t_1) = n\} = f(t_2 - t_1)$ [iii] and

$P\{N(t_2) - N(t_1) = n, N(t_3) - N(t_2) = m\} = P\{N(t_2) - N(t_1) = n\} \cdot P\{N(t_3) - N(t_2) = m\}$

for $\forall\ t_1, t_2, t_3 \geq 0$, such that $t_3 \geq t_2 \geq t_1$;

(3) the distribution of $N(t)$ is Poisson with mean λt, i.e.

$$P\{N(t) = n\} = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \qquad n=0,1,2, \ldots$$

Theorem A2.1 If each event in a Poisson process of rate λ is marked with probability p, independently from event to event, then the marked process $\{N_1(t) \mid t \geq 0\}$ (where $N_1(t)$ is the number of marked events up to time t), is a Poisson process with rate $p\lambda$. (This phenomenon is called 'random splitting' or 'random thinning' of a Poisson process [Ross]. For an illustration see Figure A2.1.)

[iii] Here f annotates an unknown function of the difference $(t_2 - t_1)$.

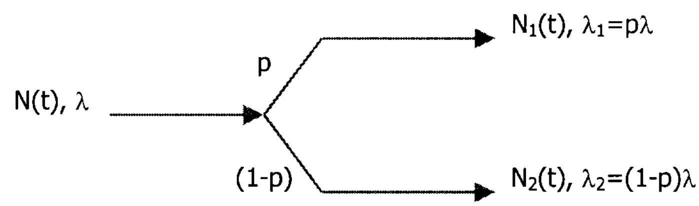


Figure A2.1 Random splitting of a Poisson process into two sub-processes

Proof of Theorem A2.1

(a) Initial Conditions

The initial condition concerning $N_1(t)$ implies directly from the initial condition concerning N(t). In particular,

$$N(0) = 0 \qquad \Rightarrow \qquad N_1(0) = 0 \tag{A2.1}$$

(c) Independent Increments

The independence of increments in $N_1(t)$ is directly inherited from the independence of the increments in the original Poisson process N(t).

(b) Stationary Increments

In order to prove that the increments of $N_1(t)$ are stationary, let us find the expression for $P\{N_1(t_2) - N_1(t_1) = n\}$.

$$
\begin{aligned}
&P\{N_1(t_2)-N_1(t_1)=n\} = \\
&=\sum_{k=n}^{\infty} P\{N_1(t_2)-N_1(t_1)=n \mid N(t_2)-N(t_1)=k\}\cdot P\{N(t_2)-N(t_1)=k\} = \\
&=\sum_{k=n}^{\infty}\binom{k}{n}\cdot p^n(1-p)^{(k-n)}\cdot P\{N(t_2)-N(t_1)=k\} = \\
&=\sum_{k=n}^{\infty}\binom{k}{n}\cdot p^n(1-p)^{(k-n)}\cdot\frac{(\lambda(t_2-t_1))^k}{k!}e^{-\lambda(t_2-t_1)} = \\
&=\sum_{k=n}^{\infty}\frac{k!}{(k-n)!n!}\cdot p^n(1-p)^{(k-n)}\cdot\frac{(\lambda(t_2-t_1))^{(n-k)}(\lambda(t_2-t_1))^n}{k!}e^{-\lambda(p+(1-p)(t_2-t_1)} = \\
&=\frac{(p\lambda(t_2-t_1))^n}{n!}e^{-\lambda p(t_2-t_1)}\cdot\left(\sum_{k=n}^{\infty}\frac{((1-p)\lambda(t_2-t_1))^{(n-k)}}{(k-n)!}\right)\cdot e^{-\lambda(1-p)(t_2-t_1)} = \\
&=\frac{(p\lambda(t_2-t_1))^n}{n!}e^{-\lambda p(t_2-t_1)}\cdot\left(\sum_{i=0}^{\infty}\frac{((1-p)\lambda(t_2-t_1))^i}{(i)!}\right)\cdot e^{-\lambda(1-p)(t_2-t_1)} = \\
&=\frac{(p\lambda(t_2-t_1))^n}{n!}e^{-\lambda p(t_2-t_1)}\cdot e^{\lambda(1-p)(t_2-t_1)}\cdot e^{-\lambda(1-p)(t_2-t_1)} = \\
&=\frac{(p\lambda(t_2-t_1))^n}{n!}e^{-\lambda p(t_2-t_1)}
\end{aligned}
\tag{A2.2}
$$

As (A2.2) indicates, the increment $N_1(t_2)$-$N_1(t_1)$ ($\forall t_1$, $t_2 \geq 0$, such that $t_2 \geq t_1$) depends on t_2 and t_1 only through their difference. Hence, we can claim that, in general, the increments of $N_1(t)$ are stationary.

(d) Cumulative Distribution Function

The steps involved in obtaining the expression for $P\{N_1(t)=n\}$ are given in (A2.3

$$
\begin{aligned}
&P\{N_1(t)=n\} = \\
&=\sum_{k=n}^{\infty} P\{N_1(t)=n \mid N(t)=k\}\cdot P\{N(t)=k\} = \\
&=\sum_{k=n}^{\infty}\binom{k}{n}\cdot p^n(1-p)^{(k-n)}\cdot P\{N(t)=k\} =
\end{aligned}
\qquad \text{(A2.3}
$$

$$
\begin{aligned}
&= \sum_{k=n}^{\infty} \frac{k!}{(k-n)!\,n!} \cdot p^{n}(1-p)^{(k-n)} \cdot e^{-\lambda t} \frac{(\lambda t)^{k}}{k!} = \\
&= \frac{p^{n} \cdot e^{-\lambda t}}{(1-p)^{n}\, n!} \sum_{k=n}^{\infty} \frac{1}{(k-n)!} \cdot \left((1-p)(\lambda t)\right)^{k} = \\
&= \frac{p^{n} \cdot e^{-\lambda t}}{(1-p)^{n}\, n!} \sum_{m=0}^{\infty} \frac{1}{m!} \cdot \left((1-p)(\lambda t)\right)^{m+n} = \\
&= \frac{(p\lambda t)^{n} \cdot e^{-\lambda t}}{n!} \sum_{m=0}^{\infty} \frac{1}{m!} \cdot \left((1-p)(\lambda t)\right)^{m} = \\
&= \frac{(p\lambda t)^{n} \cdot e^{-\lambda t}}{n!} e^{(1-p)\lambda t}
\end{aligned}
$$

From (A2.3), (A2.4) is obtained, and it clearly verifies that $N_1(t)$ follows a Poisson distribution with mean $p\lambda t$.

$$
P\{N_1(t) = n\} = e^{-p\lambda t} \frac{(p\lambda t)^{n}}{n!} \tag{A2.4}
$$

Based on (a) to (d), $N_1(t)$ satisfies all the requirements of Definition A2.1, and therefore is proven to be a Poisson process with rate $p\lambda$.

By following a similar procedure, one could easily prove that process $N_2(t)$ (see Figure A2.1), formed by *unmarked events*, is also a Poisson process with rate $(1-p)\lambda$.

Furthermore, it should be observed that the two processes $N_1(t)$ and $N_2(t)$ are independent, as illustrated in (A2.5).

$$
\begin{aligned}
&P\{N_1(t) = n \cap N_2(t) = m\} = \\
&= P\{N_1(t) = n \cap N_2(t) = m \mid N(t) = n+m\} \cdot P\{N(t) = n+m\} = \\
&= \binom{n+m}{m} \cdot p^{n} \cdot (1-p)^{m} \cdot e^{-\lambda t} \cdot \frac{(\lambda t)^{(n+m)}}{(n+m)!}
\end{aligned} \tag{A2.5}
$$

$$= \frac{(n+m)!}{n!m!} \cdot (p\lambda t)^n \cdot ((1-p)\lambda t)^m \cdot e^{-(1-p+p)\lambda t} \frac{1}{(n+m)!} =$$
$$= e^{-p\lambda t} \frac{(p\lambda t)^n}{n!} \cdot e^{-(1-p)\lambda t} \frac{((1-p)\lambda t)^m}{m!} =$$
$$= P\{N_1(t) = n\} \cdot P\{N_2(t) = m\}$$

Finally, please note, the above presented proof readily extends to the case of 'random splitting' into an arbitrary number of sub-processes, as illustrated in Figure A2.2, assuming (A2.6) is satisfied.

$$\sum_{i=1}^{n} p_i = 1 \tag{A2.6}$$

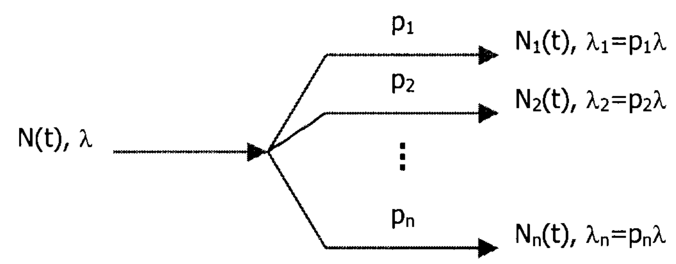


Figure A2.2 Random splitting of a Poisson process into n sub-processes

End of Proof

Appendix 3

Transient State Probabilities of Poisson (Counting) Process

The Poisson process, also known as *counting process*, is a special case of continuous-time Markov chain (CTMC) [*Bharucha-Reid*]. In particular, it is a CTMC in which transitions from *State i* can only go to *State i+1* (see Figure A2.1), while the actual values of transition rates are the same for all *State i* (i=0,1,2,...).

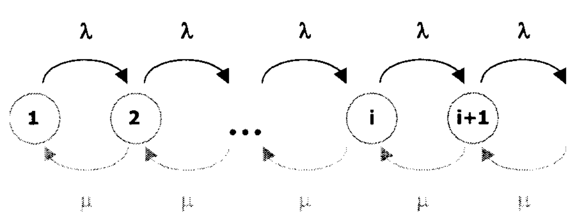


Figure A3.1 Pure-birth process

Now, let us observe a Poisson process, with the following transition rates and initial state probabilities

$$\begin{aligned} \lambda_i &= \lambda, \quad i = 0,1,2,... \\ \mu_i &= 0, \quad i = 0,1,2,... \end{aligned} \tag{A3.1}$$

$$P_i(t=0) = \begin{cases} 1, & i = 0 \\ 0, & i > 0 \end{cases} \tag{A3.2}$$

The generator matrix of such a process is given in (A3.3), and the corresponding Chapman-Kolmogorov equations are given in (A3.4).

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \ldots \\ 0 & -\lambda & \lambda & 0 & \ldots \\ 0 & 0 & \lambda & -\lambda & \ldots \\ \ldots & & & & \end{bmatrix} \tag{A3.3}$$

$$\frac{dP_0(t)}{dt} = -\lambda \cdot P_0(t) \qquad \text{(A3.4.a))}$$

$$\frac{dP_i(t)}{dt} = \lambda \cdot P_{i-1}(t) - \lambda \cdot P_i(t), \quad i > 0 \qquad \text{(A3.4.b))}$$

(A3.4.a)) clearly implies

$$P_0(t) = C_1 e^{-\lambda t} \tag{A3.5}$$

while (A3.4.b)) can be rewritten as

$$\frac{dP_i(t)}{dt} + \lambda \cdot P_i(t) = \lambda \cdot P_{i-1}(t), \quad i > 0 \tag{A3.6}$$

Since (A3.7) holds in general,

$$\frac{d}{dt}\left(e^{\lambda t} P_i(t)\right) = e^{\lambda t}\left[\frac{dP_i(t)}{dt} + \lambda \cdot P_i(t)\right], \quad i > 0 \tag{A3.7}$$

(A3.6) can be further transformed to (A3.8)

$$\frac{d}{dt}\left(e^{\lambda t}P_i(t)\right) = e^{\lambda t}\lambda P_{i-1}(t), \quad i > 0 \tag{A3.8}$$

and finally to the recursion expression (A3.9)

$$P_i(t) = e^{-\lambda t}\left[\lambda\int_0^t e^{\lambda t'}P_{i-1}(t')dt' + C_i\right], \quad i > 0 \tag{A3.9}$$

By combining the initial conditions (A3.2) with (A3.5) and (A3.9), it can be easily seen that

$$C_i = \begin{cases} 1, & i = 0 \\ 0, & i > 0 \end{cases} \tag{A3.10}$$

Hence, based on (A3.5), (A3.9), and (A3.10), we obtain the final expressions for the transient state probabilities of the Poisson process depicted in Figure A3.1, as shown in (A3.11).

$$P_i(t) = \frac{(\lambda t)^i}{i!}e^{-\lambda t}, \quad i = 0,1,2,... \tag{A3.11}$$

Appendix 4

Transient State Probabilities of MC User Behaviour Model under Assumption of Time Decreasing $P_{reattempt}$

The Chapman-Kolmogorov differential equations [*Bharucha-Reid*] associated with the model in Figure 2.8 (i.e. Figure A4.1) and the corresponding generator matrix (see (2.21) in Section 2.3.1.2) are given in (A4.1) to (A4.4).

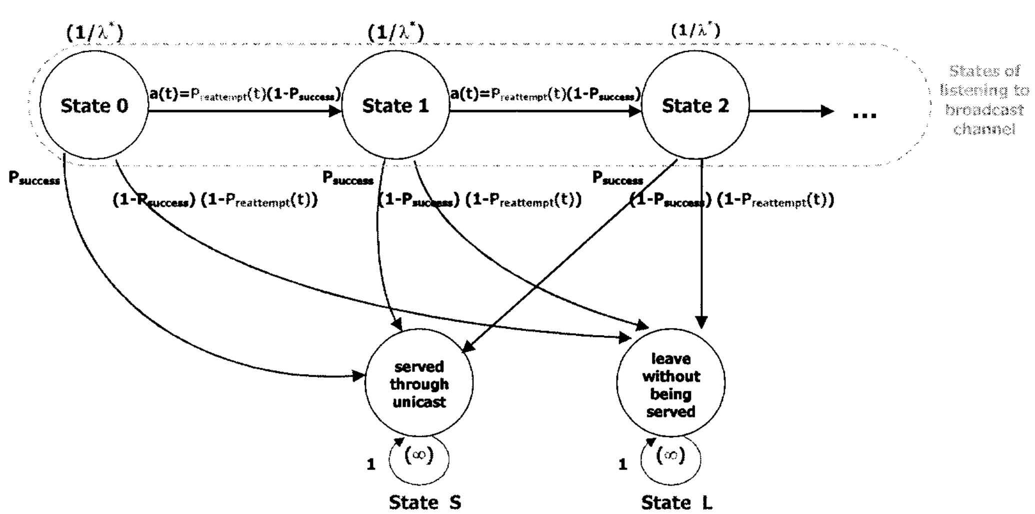


Figure A4.1 User behaviour under the assumption of time-decreasing $P_{reattempt}$

$$P_0'(t) = -\lambda^* P_0(t) \tag{A4.1}$$

$$P_i'(t) = \lambda^* a(t)\, P_{i-1}(t) - \lambda^* P_i(t), \quad i > 0 \tag{A4.2}$$

$$P_S'(t) = \lambda^* P_{success}\left(\sum_{i=0}^{\infty} P_i(t)\right) \tag{A4.3}$$

$$P_L'(t) = \lambda^*(1 - P_{success})(1 - P_{reattempt}(t))\left(\sum_{i=0}^{\infty} P_i(t)\right) \tag{A4.4}$$

Please note, the assumed initial conditions associated with the model in Figure A4.1 are

$$P_{index}(t=0) = \begin{cases} 1, & \text{index} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{A4.5}$$

By following a procedure similar to the one presented in Appendix 3, the unique solution to (A4.1) and (A4.2) is obtained, as shown in (A4.6).

$$P_i(t) = e^{-\lambda^* t}\frac{\left(\frac{\lambda^* a}{b}\left(1 - e^{-bt}\right)\right)^i}{i!}, \quad i \geq 0 \tag{A4.6}$$

Consequently, from (A4.5) and (A4.6), (A4.7) is derived.

$$\sum_{i=0}^{\infty} P_i(t) = e^{-\lambda^*\left(t - \frac{a}{b}\left(1 - e^{-bt}\right)\right)} \tag{A4.7}$$

By employing (A4.7) back in (A4.3), the implicit form expression for $P_S(t)$, (A4.8), is obtained.

$$P_S(t) = \lambda^* P_{success}\int_0^t e^{-\lambda^*\left(t' - \frac{a}{b}\left(1 - e^{-bt'}\right)\right)} dt' \tag{A4.8}$$

The actual procedure for finding $P_S(t)$ from (A4.8), and the final results, is presented in (A4.9).

$$P_S(t) = \lambda^* P_{success} \int_0^t e^{-\lambda^*\left(t' - \frac{a}{b}\left(1-e^{-bt'}\right)\right)} dt' = \left[e^{-bt'} = x \quad \Rightarrow \quad dt' = \frac{-1}{b \cdot x} dx \right] \tag{A4.9}$$

$$= \lambda^* P_{success} \frac{-1}{b} \int_1^{e^{-bt}} e^{-\lambda^*\left(\frac{-1}{b}\ln x - \frac{a}{b}(1-x)\right)} \frac{dx}{x} =$$

$$= \lambda^* P_{success} \frac{-1}{b} \int_1^{e^{-bt}} e^{\ln(x^{\frac{\lambda^*}{b}})} e^{\lambda^* \frac{a}{b}} e^{-\lambda^* \frac{a}{b} x} \frac{dx}{x} =$$

$$= \lambda^* P_{success} \frac{-e^{\lambda^* \frac{a}{b}}}{b} \int_1^{e^{-bt}} x^{\frac{\lambda^*}{b}-1} e^{-\lambda^* \frac{a}{b} x} dx = \left[\lambda^* \frac{a}{b} x = y \quad \Rightarrow \quad dx = \frac{b}{\lambda^* a} dy \right]$$

$$= \lambda^* P_{success} \frac{-e^{\lambda^* \frac{a}{b}}}{b} \int_{\lambda^* \frac{a}{b}}^{\lambda^* \frac{a}{b} e^{-bt}} \left(\frac{b}{\lambda^* a} y \right)^{\frac{\lambda^*}{b}-1} e^{-y} \frac{b}{\lambda^* a} dy =$$

$$= \frac{P_{success}}{a} \left(e^{\lambda^* \frac{a}{b}} \right) \left(\frac{b}{\lambda^* a} \right)^{\frac{\lambda^*}{b}-1} \left[\Gamma\left(\frac{\lambda^*}{b}, \lambda^* \frac{a}{b} \right) - \Gamma\left(\frac{\lambda^*}{b}, \lambda^* \frac{a}{b} e^{-bt} \right) \right]$$

Now, by employing a substitute expression for the incomplete Gamma function (i.e. the right-hand side of (A4.10)) in (A4.9), an alternative expression for $P_S(t)$ is found, as shown in (A4.11).

$$\Gamma(a,x) = \int_0^x t^{a-1} e^{-t} dt \overset{\text{approximation}}{=} x^a e^{-x} \sum_{n=0}^{\infty} \frac{x^n}{a(a+1)(a+2)..(a+n)} \tag{A4.10}$$

$$P_S(t) = \frac{P_{success}}{a} \left(e^{\lambda^* \frac{a}{b}} \right) \left(\frac{b}{\lambda^* a} \right)^{\frac{\lambda^*}{b}-1} \cdot \tag{A4.11}$$

$$\cdot \left[\left(\lambda^* \frac{a}{b} \right)^{\frac{\lambda^*}{b}} e^{-\lambda^* \frac{a}{b}} \sum_{n=0}^{\infty} \frac{\left(\lambda^* \frac{a}{b} \right)^n}{\frac{\lambda^*}{b}(\frac{\lambda^*}{b}+1)(\frac{\lambda^*}{b}+2)..(\frac{\lambda^*}{b}+n)} - \left(\lambda^* \frac{a}{b} e^{-bt} \right)^{\frac{\lambda^*}{b}} e^{-\lambda^* \frac{a}{b} e^{-bt}} \sum_{n=0}^{\infty} \frac{\left(\lambda^* \frac{a}{b} e^{-bt} \right)^n}{\frac{\lambda^*}{b}(\frac{\lambda^*}{b}+1)(\frac{\lambda^*}{b}+2)..(\frac{\lambda^*}{b}+n)} \right] =$$

$$= \frac{P_{success}}{a} \left(\frac{b}{\lambda^* a} \right)^{-1} \cdot \left[\sum_{n=0}^{\infty} \frac{\left(\lambda^* \frac{a}{b} \right)^n}{\frac{\lambda^*}{b}(\frac{\lambda^*}{b}+1)(\frac{\lambda^*}{b}+2)..(\frac{\lambda^*}{b}+n)} - \sum_{n=0}^{\infty} \frac{\left(e^{-bt}\right)^{\frac{\lambda^*}{b}} e^{-\lambda^* \frac{a}{b} e^{-bt}} e^{\lambda^* \frac{a}{b}} \left(\lambda^* \frac{a}{b} e^{-bt} \right)^n}{\frac{\lambda^*}{b}(\frac{\lambda^*}{b}+1)(\frac{\lambda^*}{b}+2)..(\frac{\lambda^*}{b}+n)} \right] =$$

$$= \frac{P_{success}}{a}\left[\sum_{n=0}^{\infty}\frac{\left(\lambda^*\frac{a}{b}\right)^{n+1}}{\frac{\lambda^*}{b}(\frac{\lambda^*}{b}+1)(\frac{\lambda^*}{b}+2)..(\frac{\lambda^*}{b}+n)} - e^{-\lambda^* t}e^{-\lambda^*\frac{a}{b}e^{-bt}}e^{\lambda^*\frac{a}{b}}\sum_{n=0}^{\infty}\frac{\left(\lambda^*\frac{a}{b}\right)^{n+1}e^{-nbt}}{\frac{\lambda^*}{b}(\frac{\lambda^*}{b}+1)(\frac{\lambda^*}{b}+2)..(\frac{\lambda^*}{b}+n)}\right] =$$

$$= \frac{P_{success}}{a}\left[\sum_{n=0}^{\infty}\frac{(\lambda^* a)^{n+1}}{\lambda^*(\lambda^*+b)(\lambda^*+2b)..(\lambda^*+nb)} - \sum_{n=0}^{\infty}\frac{e^{-\lambda^*(t-\frac{a}{b}(1-e^{-bt}))}(\lambda^* a)^{n+1}e^{-nbt}}{\lambda^*(\lambda^*+b)(\lambda^*+2b)..(\lambda^*+nb)}\right] =$$

$$= \frac{P_{success}}{a}\left[\sum_{n=0}^{\infty}\frac{(\lambda^* a)^{n+1}\left(1-e^{-\lambda^*(t-\frac{a}{b}(1-e^{-bt}))}e^{-nbt}\right)}{\lambda^*(\lambda^*+b)(\lambda^*+2b)..(\lambda^*+nb)}\right]$$

By finding the limit of (A4.11) at $b \to 0$, (A4.12) is obtained.

$$\lim_{b\to 0} P_S(t) = \frac{P_{success}}{1-a}\left(1-e^{-\lambda^*(1-a)t}\right) \qquad \text{(A4.12)}$$

Since (A4.12) is identical to (2.17) (see Section 2.3.1.1), which represents $P_S(t)$ for $P_{reattempt}$=const , the validity of (A4.11) is indirectly confirmed.

Now that $P_i(t)$ (i=0,1,2,...) and $P_S(t)$ are known, $P_L(t)$ is found simply from (A4.13).

$$P_L(t) = 1 - \sum_{i=0}^{\infty} P_i(t) - P_S(t) = \qquad \text{(A4.13)}$$

$$= 1 - e^{-\lambda^*(t-\frac{a}{b}(1-e^{-bt}))} - P_{success}\left[\sum_{n=0}^{\infty}\frac{(a)^n(\lambda^*)^{n+1}\left(1-e^{-\lambda^*(t-\frac{a}{b}(1-e^{-bt}))}e^{-nbt}\right)}{\lambda^*(\lambda^*+b)(\lambda^*+2b)..(\lambda^*+nb)}\right]$$

Again, by finding the limit of (A4.13) at $b \to 0$, (A4.14) is obtained. Since (A4.14) is identical to (2.19), which represents to $P_L(t)$ for $P_{reattempt}$=const , the validity of (A4.11) is, also, indirectly confirmed.

$$\lim_{b \to 0} P_L(t) = \frac{(1 - P_{reattempt})(1 - P_{success})}{1 - a}\left(1 - e^{-\lambda^*(1-a)t}\right) \qquad (A4.14)$$

Appendix 5

Procedure for Obtaining (2.51)

In order to arrive at the actual expression for three elements in the right-hand side summation of (2.50) (i.e. (2.51.a), (2.51.b), and (2.51.c)), let us find the expressions for individual probability density functions figuring in (2.50).

(1) In Definition 2.4, it has been assumed that WT refers to the average waiting time experienced by a randomly arrived user. Hence,

$$p(t_a \mid \text{item} = i) = 1/s(i), \quad \text{where } t_a \in [0, s(i)), \quad i = 1,.., M \tag{A5.1}$$

(2) Expression (2.27) (see Section 2.3.2) represents the probability that a MU, who arrives in the system at time $t_a=\tau\in[0,s(i))$ and wants to obtain item(i), does not leave the system or get served through a unicast channel before time s(i), i.e. $P(t_d \geq s(i) \mid t_a=\tau, \text{item}=i, t_{br}(i)=s(i))$. Based on this probability, the expression for $p(t_d \mid t_a=\tau, \text{item}=i)$ can be obtained, as given in (A5.2).

$$p(t_d \mid t_a = \tau, \text{item} = i) = \frac{\partial\left(1 - P(t_d \geq \mu \mid t_a = \tau, \text{item} = i)\right)}{\partial t_d} = \lambda^{*}(1-a)e^{-\lambda^{*}(1-a)(t_d-\tau)} \tag{A5.2}$$

(3) If a MU, who arrives in the system at time $t_a=\tau\in[0,s(i))$ and wants to obtain item(i), does not get served through a unicast channel or leave the system by some time $t_d=\mu<s(i)$, then this MU gets served through the broadcast channel (at time s(i)) with certainty one.

Therefore, the expression concerning p(sm=1|t_a=τ, t_d=μ, item=i) is of the form as given in (A5.3).

$$p(\mathrm{sm}=1 \mid t_a=\tau, t_d=\mu, \mathrm{item}=i)=\begin{cases}0, & \mu<s(i)\\ 1, & \mu\geq s(i)\end{cases} \tag{A5.3}$$

In order to obtain the expression for p(sm=2 | t_a=τ, t_d=μ, item=i), let us recall, from (2.35), that the expression concerning the probability that a MU (which arrives in the system at time $t_a=\tau\in[0,s(i))$ and requires to retrieve item(i)) seizes one of the point-to-point channels and gets served on-demand before time t_d=μ, is as given below.

$$P(t_d<\mu \cap \mathrm{sm}=2 \mid t_a=\tau, \mathrm{item}=i, t_{br}(i)=s(i))=\frac{P_{success}}{1-a}\left[1-e^{-\lambda^{*}(1-a)(\mu-\tau)}\right] \tag{A5.4}$$

The above probability can also be represented as $P(t_d<\mu \cap \mathrm{sm}=2 \mid t_a=\tau, \mathrm{item}=i, t_{br}(i)=s(i))=$ $=P(t_d<\mu \mid t_a=\tau, \mathrm{item}=i, t_{br}(i)=s(i))\cdot P(\mathrm{sm}=2 \mid t_a=\tau, \mathrm{item}=i, t_{br}(i)=s(i), t_d<\mu)$. However, since $P(t_d<\mu \mid t_a=\tau, \mathrm{item}=i, t_{br}(i)=s(i))=(1-e^{-\lambda^{*}(1-a)(\mu-\tau)})$ (based on (2.27)),

$$p(\mathrm{sm}=2 \mid t_a=\tau, t_d=\mu, \mathrm{item}=i)=\begin{cases}\dfrac{P_{success}}{1-a}, & \mu<s(i)\\ 0, & \mu\geq s(i)\end{cases} \tag{A5.5}$$

Similarly, based on (A5.5), the expression concerning p(sm=0| t_a=τ, t_d=μ, item=i) can be obtained, as given in (A5.6).

$$p(\mathrm{sm}=0 \mid t_a=\tau, t_d=\mu, \mathrm{item}=i)=\begin{cases}\dfrac{(1-P_{success})(1-P_{reattempt})}{1-a}, & \mu<s(i)\\ 0, & \mu\geq s(i)\end{cases} \tag{A5.6}$$

From (A5.1) to (A5.6), expressions (2.51.a), (2.51.b) and (2.51.c) of Section 2.3.5 can be derived, as shown in (A5.7.a), (A5.7.b), and (A5.7.c).

$$\int_{0}^{s(i)}\int_{\tau}^{\infty} wt_i(t_a=\tau,t_d=\mu,sm=1)p(t_a=\tau,t_d=\mu,sm=1\,|\,item=i)d\tau d\mu = \tag{A5.7.a}$$

$$= \int_{0}^{s(i)} d\tau \int_{s(i)}^{\infty} d\mu (s(i)-\tau)\frac{1}{s(i)}\lambda^*(1-a)e^{-\lambda^*(1-a)(\mu-\tau)} =$$

$$= \frac{1}{s(i)}\int_{s(i)}^{0}(s(i)-\tau)e^{-\lambda^*(1-a)(s(i)-\tau)}d\tau =$$

$$= \frac{1}{s(i)\left(\lambda^*(1-a)\right)^2}\left[1-\left(1+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]$$

$$\int_{0}^{s(i)}\int_{\tau}^{\infty} wt_i(t_a=\tau,t_d=\mu,sm=2)p(t_a=\tau,t_d=\mu,sm=2\,|\,item=i)d\tau d\mu = \tag{A5.7.b}$$

$$= \int_{0}^{s(i)} d\tau \int_{\tau}^{s(i)} d\mu (\tau-\mu)\frac{1}{s(i)}\lambda^*(1-a)e^{-\lambda^*(1-a)(\mu-\tau)}\frac{P_{success}}{(1-a)} =$$

$$= \frac{P_{success}}{\lambda^*(1-a)^2 s(i)}\int_{0}^{s(i)} d\tau\left(1-\left(1+\lambda^*(1-a)(s(i)-\tau)\right)e^{-\lambda^*(1-a)(s(i)-\tau)}\right) =$$

$$= \frac{P_{success}}{\lambda^*(1-a)^2 s(i)}\left[s(i)\frac{1}{\lambda^*(1-a)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)+\frac{1}{\lambda^*(1-a)}\left(-1+\left(1+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right)\right] =$$

$$= \frac{P_{success}}{\left(\lambda^*\right)^2\left(1-a\right)^3 s(i)}\left[-2+\lambda^*(1-a)s(i)+\left(2+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]$$

$$\int_{0}^{s(i)}\int_{\tau}^{\infty} wt_i(t_a=\tau,t_d=\mu,sm=0)p(t_a=\tau,t_d=\mu,sm=0\,|\,item=i)d\tau d\mu = \tag{A5.7.c}$$

$$= \int_{0}^{s(i)} d\tau \int_{\tau}^{s(i)} d\mu (\tau-\mu)\frac{1}{s(i)}\lambda^*(1-a)e^{-\lambda^*(1-a)(\mu-\tau)}\frac{(1-P_{success})(1-P_{reattempt})}{(1-a)} =$$

$$= \frac{(1-P_{success})(1-P_{reattempt})}{\lambda^*(1-a)^2 s(i)}\int_{0}^{s(i)} d\tau\left(1-\left(1+\lambda^*(1-a)(s(i)-\tau)\right)e^{-\lambda^*(1-a)(s(i)-\tau)}\right) =$$

$$= \frac{(1-P_{success})(1-P_{reattempt})}{\lambda^*(1-a)^2 s(i)}\left[s(i)-\frac{1}{\lambda^*(1-a)}\left(1-e^{-\lambda^*(1-a)s(i)}\right)+\frac{1}{\lambda^*(1-a)}\left(-1+\left(1+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right)\right] =$$

$$= \frac{(1-P_{success})(1-P_{reattempt})}{\left(\lambda^*\right)^2\left(1-a\right)^3 s(i)}\left[-2+\lambda^*(1-a)s(i)+\left(2+\lambda^*(1-a)s(i)\right)e^{-\lambda^*(1-a)s(i)}\right]$$

Appendix 6

Calculation of k and χ in (3.30)

In order to obtain the actual values of parameters k and χ in (3.30), first we need to find the interval $[A,+\infty]$ where F(x(i)) can be sufficiently approximated with the second segment of H(x(i)), i.e. F(x(i)) can be sufficiently approximated with –1.

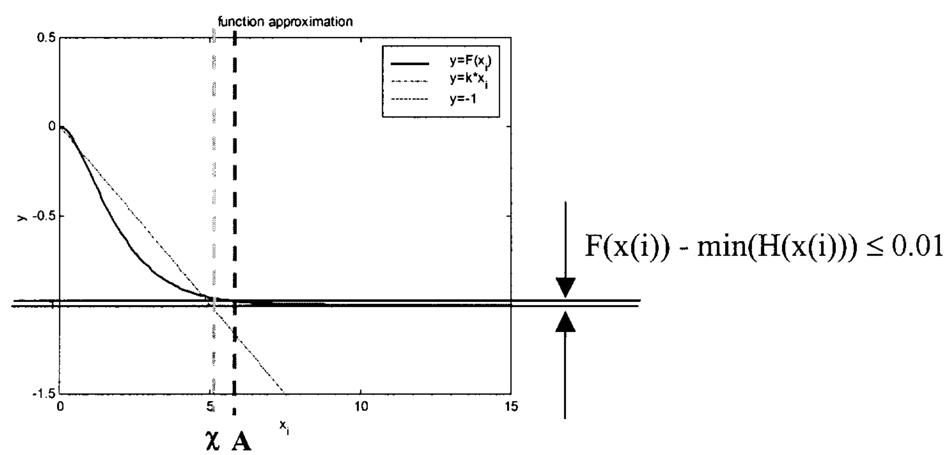


Figure A6.1 Sufficient second segment approximation of F(x(i))

According to Definition 3.3, $[A,+\infty]$ consists of points $x(i) \in R^{+}$ such that $|F(x(i))-(-1)| \leq 0.01$, as illustrated in Figure A6.1. Based on this, and by noting that F(x(i)) is a monotonically decreasing function with the values in the range [0,-1), we conclude that A simply corresponds to the point where F(x(i)) drops to –0.99. Accordingly, the actual value of A can be obtained as the solution to the non-linear equation given in (A5.1), i.e. (A5.2).

$$F(x(i)) = x(i)e^{-x(i)} + e^{-x(i)} - 1 = -0.99 \tag{A6.1}$$

$$x(i)e^{-x(i)} + e^{-x(i)} - 0.01 = 0 \tag{A6.2}$$

Based on the discussion of Section 3.2, we already know than no analytical closed-form solution to (A6.1), i.e. (A6.2) exists. Nevertheless, by employing a MATLAB estimation procedure, a numerical value of A can be reached, as shown in Figure A6.2.

Matlab Numerical Estimation of A

Step 1: write an M-file called f.m

```
function y=f(x)
y=x*2.7.^(-x)+2.7.^(-x)-0.01
```

Step 2: set the initial search point

```
A0=4
```

Step 3: find the zero point of y=f(x) given A0

```
A = fzero('f',A0)
A = 6.6903
```

Figure A6.2 Numerical estimation of A

Now that the value of A is known, we will first proceed by determining the value of parameter k, based on (3.30) and Definition 3.2. Subsequently, once k is known, χ will be readily obtained as the point where the kx(i) and −1 intersect (see Figure A6.1).

Procedure for Finding Parameter k

According to Definition 3.2, the first segment of H(x(i)), and accordingly parameter k, correspond to the minimum MSE linear approximation of F(x(i)) on [0,A). Hence, the problem of finding k can be mathematically formulated as shown in (A6.3)

$$\min_{k} E[d_{x(i)}(x(i), k)^2] \tag{A6.3}$$

where $d_{x(i)}(x(i),k)$ represents the point-wise distance between F(x(i)) and k·x(i)) on [0,A) (see (A6.4) and Figure A6.3).

$$d_{x(i)}(x(i), k) = F(x(i)) - kx(i), \qquad x(i) \in [0, A) \tag{A6.4}$$

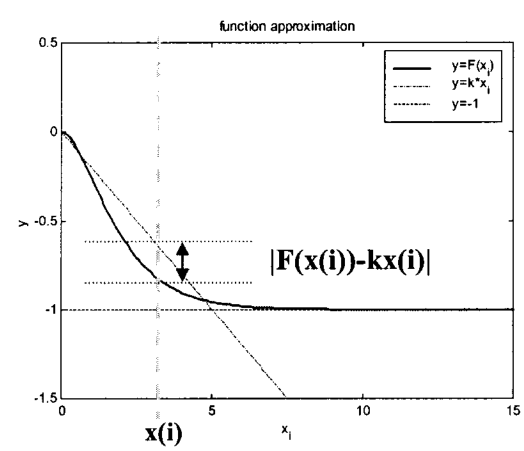


Figure A6.3 Point-wise distance between F(x(i)) and H(x(i)) on [0,A)

Based on (A6.4), the expanded expression for MSE figuring in (A6.3) turns out to be of the following form

$$E[d_{x(i)}(x(i),k)^2] = \int_0^A d_{x(i)}(x(i),k)^2 \cdot pdf_{x(i)}(x(i)) \cdot dx(i) = \tag{A6.5}$$

$$= \int_0^A [F(x(i)) - kx(i)]^2 \cdot \frac{1}{A} \cdot dx(i) =$$

$$= \frac{1}{A} \cdot \int_0^A \left[F^2(x(i)) - 2kF(x(i))x(i) + (kx(i))\right] dx(i) =$$

$$= \frac{1}{A} \cdot \int_0^A F^2(x(i))\,dx(i) - \frac{2k}{A}\int_0^A F(x(i))x(i)\,dx(i) + \frac{k^2}{A} \cdot \int_0^A x(i)^2\,dx(i) =$$

$$= \frac{1}{A} \cdot \int_0^A F^2(x(i))\,dx(i) - \frac{2k}{A}\int_0^A (x(i)e^{-x(i)} + e^{-x(i)} - 1) \cdot x(i)\,dx(i) + \frac{k^2A^2}{3} =$$

$$= \frac{1}{A} \cdot \int_0^A F^2(x(i))\,dx(i) - \frac{2k}{A}\int_0^A x(i)^2 e^{-x(i)}\,dx(i) - \frac{2k}{A}\int_0^A x(i)e^{-x(i)}\,dx(i) + \frac{2k}{A}\int_0^A x(i)\,dx(i) + \frac{k^2A^2}{3} =$$

$$= \frac{1}{A} \cdot \int_0^A F^2(x(i))\,dx(i) - \frac{2k}{A}\Gamma(3,A) - \frac{2k}{A}\Gamma(2,A) + kA + \frac{k^2A^2}{3}$$

Γ in (A6.5) stands for incomplete Gamma function, as already mentioned in Appendix 4 (see (A4.10)). From (A6.5) we further obtain

$$\frac{\partial E[d_{x(i)}(x(i),k)^2]}{\partial k} = -\frac{2}{A}\Gamma(3,A) - \frac{2}{A}\Gamma(2,A) + A + \frac{2kA^2}{3} \tag{A6.6}$$

$$\frac{\partial^2 E[d_{x(i)}(x(i),k)^2]}{\partial^2 k} = \frac{2A^2}{3} \tag{A6.7}$$

(A6.6) and (A6.7) prove that the given MSE function ($E[d_{x(i)}(x(i),k)^2]$) is convex with respect to k. Consequently, the solution to the initial problem (A6.3) can be obtained simply from

$$\frac{\partial E[d_{x(i)}(x(i),k)^2]}{\partial k} = -\frac{2}{A}\Gamma(3,A) - \frac{2}{A}\Gamma(2,A) + A + \frac{2A^2}{3}k = 0 \tag{A6.8}$$

Hence, based on (A6.8), the value of parameter k that correspond to the minimum MSE linear approximation of F(x(i)) on [0,A) turns out to be

$$k = \frac{3 \cdot \left(\frac{2}{A}\Gamma(3,A) + \frac{2}{A}\Gamma(2,A) - A \right)}{2A^2} \approx -0.1948 \qquad \text{(A6.8)}$$

The values of $\Gamma(3,A)=1.9258$ and $\Gamma(2,A)=0.9905$ in (A6.8) are obtained using Matlab.

Procedure for Finding Parameter χ

Now that the actual slope of the first-segment of H(x(i)) is known (see Figure (A6.1)), the knee point (χ) can be obtained simply as

$$k\chi \approx -1 \qquad \text{(A6.9)}$$

By employing (A6.8) in (A6.9), we finally obtain

$$\chi = \frac{1}{k} \approx 5.1335 \qquad \text{(A6.10)}$$

Appendix 7

Constrained Optimization Using MATLAB

The following example illustrates the main steps of a MATLAB procedure aimed at obtaining an approximate numerical solution to our constrained maximization problem, as defined in (2.55). (The procedure utilizes *Optimization Toolbox* routines. Detailed descriptions of these routines can be found in [*MathWorks*].)

Step 1: Write an M-file (fun.m) for the objective function and constraints:

```
   function[f,g] = fun(x)
% set K=λ*(1-a)
    K=0.3
% set other parameters - document request probabilities and lengths
    p=[0.4 0.2 0.2 0.1 0.1]
    l=[1 1 1 1 1]
% Objective function
    f=-(1/K)*((p(1)/x(1))*(1-exp((-K)*x(1)))+(p(2)/x(2))*(1-exp((-
    K)*x(2)))+(p(3)/x(3))*(1-exp((-K)*x(3)))+(p(4)/x(4))*(1-exp((-
    K)*x(4)))+(p(5)/x(5))*(1-exp((-K)*x(5))))
% Equality constraint first
    g(1)=l(1)/x(1)+l(2)/x(2)+l(3)/x(3)+l(4)/x(4)+l(5)/x(5)-1
% Inequality constraints
    g(2)=-x(1)
    g(3)=-x(2)
    g(4)=-x(3)
    g(5)=-x(4)
    g(6)=-x(5)
```

Step 2: **On command line, set initial value for x (x0) first, then invoke constrained optimization routine**

```
    p=[0.4 0.2 0.2 0.1 0.1]
    l=[1 1 1 1 1]
    x0=l./p
% Specify number of equality constraints - in our case there is only one
    options(13)=1
    x=constr('fun',x0,options)
```

Appendix 8

Function Approximation

In order to simplify the comparative performance analysis presented in Section 6.4, we introduce the following function approximation

$$1 - e^{-x} \approx k\sqrt[n]{x} \qquad \text{(A8.1)}$$

where k=2/3, and n=8. The two functions from (A8.1) are presented in Figure A8.1.

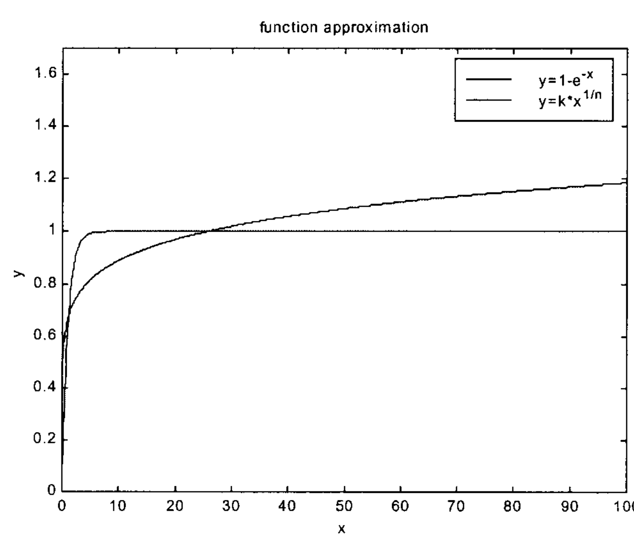


Figure A8.1 Function Approximation

References

[***Acharya_1***] S. Acharya, M. J. Franklin, S. Zdonik, *Dissemination-based Data Delivery Using Broadcast Disks, IEEE Personal Communications*, Vol. 2, No. 6, December 1995.
http://www1.bell-labs.com/user/acharya/papers/index.html

[***Acharya_2***] S. Acharya, S. Muthukrishnan, *Scheduling On-demand Broadcasts: New Metrics and Algorithms*, Proceedings of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom), pp.43-54 Dallas, Oct 1998.
http://www1.bell-labs.com/user/acharya/papers/index.html

[***Aksoy***] D. Aksoy, M. Frenklin, *RxW: A Scheduling Approach for Large-Scale On-Demand Data Broadcast*, IEEE/ACM Transactions on Networking, Vol. 7, No. 6, pp. 846-860, December 1999.
http://www.cs.ucdavis.edu/~aksoy/addtnl/publication.html

[***Akimaru***] H. Akimaru, K. Konosuke, *Teletraffic*, Springer-Verlag, London, 1993.

[*Banks*] J. Banks, J. S. Carson, B. L. Nelson, *Discrete-Event System Simulation*, Prentice-Hall, New Jersey, 1999.

[***Baruah_1***] S. Baruah, S. S. Lin, *Pfair Scheduling of Generalized Pinwheel Task Systems*, IEEE Transactions on Computers, Vol. 47, No. 7, pp. 812-816, July 1998.
http://www.cs.unc.edu/~baruah/Papers/Publications

[***Baruah_2***] S. Baruah, A. Bestavros, *Pinwheel Scheduling for Fault-tolerant Broadcast Disks in Real-time Database Systems*, Proceedings of IEEE International Conference on Data Engineering, pp 543-551, Birmingham, UK. April 1997.

[***Bharucha-Reid***] A. T. Bharucha-Reid, *Elements of the Theory of Markov Processes and Their Application*, McGraw-Hill, New York, 1960.

[***Belegundu***] A. Belegundu, T. R. Chandrupatla, *Optimization Concepts and Applications In Engineering*, Prentice-Hall, New Jersey, 1999.

[***Bertsekas***] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass., 1999.

[***Boyd***] S. Boyd and L. Vandenberghe, *Convex Optimization*, Course reader for EE364 (Stanford) / EE236B (UCLA), i.e. draft of a textbook that will be published in 2003.
http://www.stanford.edu/class/ee364/index.html#book

[***Carmeli***] B. Carmeli, B. Cohen, *PiNet: Wireless Connectivity for Organizational Information Access Using Lightweight Handheld Devices*, IEEE Personal Communications, pp. 18-23, Vol. 8, No. 4, August 2001.

[***CBF***] Cell Broadcast Forum, *Advantages and Services Using Cell Broadcast*, Berne, Switzerland, February 2002.
http://www.cellbroadcastforum.org

[***Cinlar***] E. Cinlar, Introduction to Stochastic Processes, Prentice-Hall Inc., Englewood Cliffs, N.J. 1975.

[***Czerwinski***] S. E. Czerwinski, B. Y Zhao, T. D. Hodes, A. D. Josehp, R. H. Katz, *An Architecture for a Secure Service Discovery*, In Proceedings of 5th ACM/IEEE International Conference on Mobile Computing (Mobicom'99), pp 24-25, Seattle, WA, August 1999.

[***DesmoJ***] http://www.desmoj.de/

[***Ekici***] E. Ekici, C. Ersoy, *Optimal Two-Tier Cellular Network Design*, Proceedings of IEEE International Conference on Computer Communications and Networks ICCCN'99, Boston, October 11-13, 1999.

[***Ercetin***] O. Ercetin, L. Tassiulas, *Information Delivery in Two-stage Satellite-Terrestrial Wireless Systems*, Proceedings of the 1998 Conference on Information Science and Systems CISS'98, Princeton, NJ, 1998.
http://www.wam.umd.edu/~ercetin/research.html

[***ETSI***] ETSI TS 123 041, *Technical realization of Cell Broadcast Service (CBS)*, version 3.4.0, Release 2001-06.

[***Fang***] Y. Fang, I. Chlamtac, *Hyper-Erlang Distribution and Traffic Modeling in Wireless and Mobile Networks*, In Proceedings of the IEEE Wireless Communications and Networking Conference (*WCNC'99)*, New Orleans, Louisiana, September 21-24, 1999.

[***Foltz***] K. Foltz, J. Bruck, *Time-Division is Better than Frequency-Division for Periodic Internet Broadcasting*, IEEE International Symposium on Information Theory, Washington, DC, June 2001.

[***Gehlsen***] B. Gehlsen, B. Page, *A Framework For Distributed Simulation Optimization*, Proceedings of the 2001 Winter Simulation Conference, Arlington, VA, USA, 9-12 December 2001.
http://www.informs-cs.org/wsc01papers/066.PDF

[***Gribble***] S. D. Gribble, M. Welsh, R. Behren, E. A. Brewer, D. Culler, N. Borisov, S. Czerwinski. R. Gummadi. J. Hill, A. Joseph, R. H. Katz, Z. M. Mao, S. Ross, B. Zhao, *The Ninja architecture for robust Internet-scale systems and services*, Computer Networks, Elsevier Science, pp. 473-497, Vol. 35, Issue 4, March 2001.

[***Gross***] D. Gross, C. M. Harris, *Fundamentals of Queueing Theory*, John Wiley & Sons, 1985.

[***Gupta***] S. K. S. Gupta, W. C. Lee, A. Purakayastha, P. K. Srimani, *An Overview of Pervasive Computing*, IEEE Personal Communications, pp. 8-9, Vol. 8, No. 4, August 2001.

[***Hameed***] S. Hameed, N. H. Vaidya, *Efficient Algorithms for Scheduling Data Broadcast*, ACM/Baltzer Wireless Networks (WINET), Vol. 5, pp 183-193, 1999.

[***Han***] I. Han, D. H. Cho, *An Efficient Allocation Scheme for Multicast Traffic in Multitier Cellular Systems*, IEICE Transactions on Communications, pp. 1087-1093, Vol. E84-B, No. 4, April 2001.
http://search.ieice.org/2001/pdf/e84-b_4_1087.pdf

[***Hermann***] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, A. Schade, *DEAPspace – Transient ad hoc networking of pervasive devices*, Computer Networks, Elsevier Science, pp. 411-428, Vol. 35, Issue 4, March 2001.

[***Jain***] R. Jain and J. Werth, *AirDisks and AirRaid: modeling and scheduling periodic wireless data broadcast*, in DIMACS Technical Report 95-11, Rutgers University, 1995.

[***Jiang_1***] S. Jiang, N. H. Vaidya, *Scheduling Data Broadcast to 'Impatient' Users*, Proceedings of International Workshop on Data Engineering for Wireless and Mobile Access, USA, Seattle, August 1999.
http://www.cs.tamu.edu/faculty/vaidya/mobile.html

[***Jiang_2***] S. Jiang, N. H. Vaidya, *Response Time in Data Broadcast Systems: Mean, Variance and Trade-Off*, Proceedings of 3rd ACM/IEEE International Workshop on Satellite-based Information Services (WOSBIS), Dallas, Texas, USA, 1998.

[***Kinoshita***] K. Kinoshita, M. Fujisawa, T. Takine, K. Murakami, *Performance Evaluation Method of Cellular Networks with Retrial*, pp. 389-393, Proceedings of 1999 IEEE International Conference on Personal Wireless Communications (ICPWC), February 17-19, 1999, Jaipur, India.

[***Lechler***] T. Lechler, B. Page, *DESMO-J: An Object Oriented Discrete Simulation Framework in Java*, Proceedings of 11th European Simulation Symposium, Erlangen, Germany, October 1999.
http://asi-www.informatik.uni-hamburg.de/themen/sim/forschung/Simulation/Desmo-J/ess/ess99.html

[***Leon-Garcia***] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, Addison Wesley, 1994.

[***LinCW***] C. W. Lin, D. L. Lee, *Adaptive Data Delivery in Wireless Communication Environments*, Proceedings of 20th International Conference on Distributed Computing Systems ICDCS 2000, pp. 444-452, Taipei, Taiwan, April, 2000.
http://www.cs.ust.hk/faculty/dlee/

[***LinKJ***] K. J. Lin, Y. C. Tseng, J. P. Sheu, *Channel-Sharing Strategies in Two-Tier Cellular PCS Systems*, Proceedings of IASTED International Conference on Applied Informatics, 2001.
http://www.csie.nctu.edu.tw/~yctseng/pub.html

[***Mandjes***] M. Mandjes, K. Tutschku, *Efficient call handling procedure in cellular mobile networks*, University of Würzburg Institute of Computer Science Research Report Series, No. 144, July 1996.
http://nero.informatik.uni-wuerzburg.de/staff/tutschku/pub/

[***Marianov***] V. Marianov, D. Serra, *Hierarchical location – allocation models for congested systems*, European Journal Of Operational Research, pp. 195-208, Vol. 135 (1), November 2001.

http://www.econ.upf.es/cgi-bin1/wpauthor1.php?author=Serra,%20Daniel

[***Marsan***] M. A. Marsan, G. D. Carolis, E. Leonardi, *Efficient Estimation of Call Blocking Probabilities in Cellular Mobile Telephony Networks with Customer Retrials*, IEEE Journal on Selected Areas in Communications, pp. 332-346, Vol. 19, No. 2, February 2001.

[***MathWorks***] The MathWorks Inc., *Optimization Toolbox User's Guide (for MATLAB 5)*, 1997.

[***Mihailescu***] C. Mihailescu, X. Lagrange, D. Zeghlache, *Analysis of a two-layer cellular mobile communication system*, Proceedings of IEEE Vehicular Technology Conference, Phoenix, USA, May 1997.
http://perso.enst.fr/~mihailes/English/Research.html

[***Nidd***] M. Nidd, *Service Discovery in DEAPspace*, IEEE Personal Communications, pp. 39-45, Vol. 8, No. 4, August 2001.

[***Onur***] E. Onur, H. Delic, C. Ersoy, M. U. Caglayan, *On the Retrial and Redial Phenomena in GSM Networks*, Proceedings of IEEE Wireless Communications and Networking Conference, pp. 885-889, Chicago, Illinois, September 23-28, 2000.

[***Orlik***] P. V. Orlik, S. S. Rappaport, *Traffic Performance and Mobility Modeling of Cellular Communications with Mixed Platforms and Highly Variable Mobility*, Proceedings of the IEEE, Vol. 86, pp. 1464-1479, No. 7, July 1998.

[***Ortigoza-Guerrero***] L. Ortigoza-Guerrero, A. H. Aghvami, *Resource Allocation in Hierarchical Cellular Systems*, Artech House, Norwood, MA, 2000.

[**Ross**] S. M. Ross, Introduction to Probability Models, Academic Press, 2000.

[***Schiller***] J. H. Schiller, *Mobile Communications*, Addison Wesley, 2000.

[***Stathatos***] K. Stathatos, N. Roussopoulos, J. S. Baras, *Adaptive Data Broadcast in Hybrid Networks*, Proceedings of 23rd Very Large Data Bases (VLDB'97) Conference, pp. 326-335, Athens, Greece, 1997.
http://www.cs.umd.edu/~nick/papers/kostas/vldb97.html

[***Stone***] C. J. Stone, *A Course in Probability and Statistics*, Duxbury Press, Belmont, USA, 1996.

[***Tran-Gia_1***] P. Tran-Gia, K. Leibnitz, K. Tutschku, *Teletraffic Issues in Mobile Communication Network Planning*, Proceedings of 11th ITC Specialist Seminar on Multimedia and Nomadic Communications, Yokohama, Japan, 1998.
http://nero.informatik.uni-wuerzburg.de/staff/tutschku/pub/

[***Tran-Gia_2***] P. Tran-Gia, M. Mandjes, *Modeling of Customer Retrial Phenomenon in Cellular Mobile Networks*, IEEE Journal on Selected Areas in Communications, pp. 1406-1414, Vol. 15, No. 8, October 1997.

[***UMTS_Report9***] UMTS Forum, *Report No. 9: The UMTS Third Generation Market – Structuring the Service Revenues Opportunities*, October 2000.
http://www.umts-forum.org/reports/report9.pdf

[***UMTS_ConvergingNetwork***] UMTS Forum, *Support of Third Generation Services using UMTS in a Converging Network Environment*, February 2002.
http://www.umts-forum.org/reports/report14.pdf

[***Vaidya***] H. H. Vaidya, S. Hameed, *Scheduling Data Broadcast in Asymmetric Communication Environments,* ACM/Baltzer Wireless Networks (WINET), Vol. 5, pp 171-182, 1999.

[***Wismer***] D. A. Wismer, R. Chattergy, *Introduction to Nonlinear Optimization A Problem Solving Approach*, New York: North Holland, 1979.

[***Xu***] J. Xu, D. L. Lee, Q. Hu, W. C. Lee, *Data Broadcast*, in Handbook of Wireless Networks and Mobile Computing, edited by I. Stojmenovic, John Wiley& Sons, in press.

List of Published Papers

N. Vlajic, C. D. Charalambous, D. Makrakis, *Performance Aspects of Data Broadcast in Wireless Networks with User Retrials*, Submitted to IEEE Transactions on Networking.

N. Vlajic, C. D. Charalambous, D. Makrakis, *Wireless Data Broadcast in Systems of Hierarchical Cellular Organization*, Proceedings of IEEE International Conference on Communications (ICC 2003), May 11-15, 2003, Anchorage, Alaska, USA.

N. Vlajic, C. D. Charalambous, D. Makrakis, *TDM- vs. FDM- Based Broadcast Scheduling in Wireless Systems with User Retrials*, Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'03), May 4-7 2003, Montreal Canada.

N. Vlajic, C. D. Charalambous, D. Makrakis, *Networks of Pervasive Devices: Fast CPU Conserving Algorithm for Optimal Broadcast Scheduling Scheme Estimation*, Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'02), May 2002, Winnipeg, Canada.

N. Vlajic, C. D. Charalambous, D. Makrakis, *Generalized Wireless Data Broadcast System Analysis: Multiobjective Performance Measures*, IASTED International Multi-Conference on Applied Informatics (AI2002), February 2002, Innsbruck, Austria.

N. Vlajic, C. D. Charalambous, D. Makrakis, *Near Optimal Wireless Data Broadcasting Based on an Unsupervised Neural Network Learning Algorithm*, International Joint INNS-IEEE Conference on Neural Networks, July 2001, Washington DC, USA.

N. Vlajic, C. D. Charalambous, D. Makrakis, *Wireless Broadcast Data Scheduling in Environment with Re-attempt Users: Optimization through Intelligent Feedback Monitoring*, IEEE International Conference on Third Generation Wireless and Beyond, May 2001, San Francisco, USA.