Situation Awareness for Mobile Information Access in Heterogeneous Wireless Networks

Dissertation

zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

der Fakultät für Ingenieurwissenschaften

der Universität Ulm

von

Michael Angermann

aus Gräfelfing

1. Gutachter:	Prof. Dr.-Ing. Jürgen Lindner
2. Gutachter:	Prof. Dr.-Ing. Jörg Eberspächer
Amtierender Dekan:	Prof. Dr.-Ing. Hans-Jörg Pfleiderer
Datum der Promotion:	17.12.2004

2004

Abstract

The physical properties of the wireless communication channel imply a sharing of the limited frequency spectrum among its users. Hence, the available transportation capacity of wireless links is typically several orders of magnitude below the capacity of comparable fixed wire connections. In consequence, mobile users, accessing information by means of a hypertext system, such as the world wide web, experience undesirable long waiting times for requested documents.

This thesis' primary objective is the improvement of information access for mobile users. The approach proposed and discussed in this thesis is partly based on the assumption that mobile devices are increasingly equipped with multiple wireless access technologies which facilitate the use of heterogeneous wireless access networks, differing in parameters such as range, data-rate and costs. Since some of these networks are deployed with partial coverage, a user's mobility causes changes of the actual network conditions. Furthermore, it is assumed that a particular user's scope of interest in information is time-variant and depending on this user's actual situation. Advances in the fields of ubiquitous and pervasive computing, particularly the work on context sensors and observation of user interaction, render it feasible to derive information on the user's context. In this thesis it is therefore proposed to employ knowledge about the user's context and behavior, to pro-actively transmit documents' data, especially during favorable network conditions, over the wireless communication link, before the user requests the documents.

Human behavior, as observed from outside, is inherently probabilistic, despite advances in sensor technology. Therefore a probabilistic model for the user context is formulated. In order to distinguish this model from non-probabilistic context models and context awareness, the terms "situation model" and "situation awareness" are introduced. Since the concept of situation awareness may as well be utilized for other applications, such as handover decisions, pro-active computing or future user interfaces to search engines, the situation model and its discussion is intentionally kept as general as possible, in order to enable its application to various domains. The properties of a situation model are analyzed from an information theoretic viewpoint. This perspective is used later on to illustrate the selection of suitable sensor data, based on metrics such as conditional entropy and mutual information.

The task of obtaining and continuously adjusting suitable probabilities for the model is formally treated as an estimation problem. Several estimators, such as maximum likelihood (ML), minimum mean square error (MMSE), and maximum

a posteriori (MAP) are discussed and related to Bayesian estimation. Particularly, the temporal development of the estimation is investigated and illustrated.

The concept of situation awareness is then applied to the prefetching of documents in hypertext systems. A thorough analysis yields qualitative and quantitative insights into the effects of situation aware prefetching on the average waiting time and transported data volume. The necessity for the assumption of a user policy is discussed and an optimum probability threshold is derived.

The investigation of situation-aware prefetching is further extended by means of simulations towards various mobile networking scenarios. For this purpose a novel mobility model is developed and used in conjunction with models for network topology and traffic to obtain insight into the influence of situation aware prefetching in both heterogeneous and hybrid wireless networking scenarios. Results of several simulations are discussed, showing the influence of parameters such as document probabilities, probability thresholds or level of network deployment. Simulation results show that significant reductions in average waiting time are obtainable with the proposed concept.

Finally, implementation aspects of the proposed concept are addressed. A system architecture for realizing situation-aware mobile information access in a heterogeneous wireless access infrastructure is proposed. Integration aspects as well as operational experiences obtained in an experimental testbed are discussed.

The thesis concludes with a short summary of the achieved results and a brief outlook to further research inspired by this work.

Preface

The research presented in this thesis has been carried out at the Institute of Communications and Navigation of the German Aerospace Center (DLR). Without the support of many individuals this work would never have been successfully completed. I am deeply indebted to every single one of them. It has been a pleasure to work with and learn from all colleagues at the department and especially the other core members of the "Heywow" project team, Jens Kammann, Frank Kühndel, Patrick Robertson, Thomas Strang and Kai Wendlandt.

Apart from these regular staff members, all students that have performed their diploma or masters thesis work and internships must be credited for their most valuable contributions to the "Heywow" project and this thesis. Among them are Stefan David, Andreas Hirschvogel, Susann Hofmann, Gerald Jeampierre, Daniel Katheininger, Bruno Lami, Haymo Lang, Sven Lange, Michael Lichtenstern, Ahmed Ouhmich and Christian Wasel.

The unbeatable organizing ability of Jutta Uellner and her skills to smoothly navigate the rough waters of DLR's administration, were always invaluable in my daily work. Jesus Selva's inexhaustible mathematical genius provided more than once a decisive hint and a second opinion on a mathematical problem. Dr. Patrick Robertson has not only been an excellent team leader, but also one of the initiators and most active contributors to the project. His keen perception, talent to understand the most diverse scientific topics and constantly good spirit are unmatched.

Dr. Uwe-Carsten Fiebig has been a challenging and always supporting head of department and is credited for the scientific steering and the excellent research conditions of the Communication Systems department at DLR.

I had the pleasure to be a part of the Institute of Communications and Navigation under the guidance of Dr. Friedrich Kühne. The atmosphere of wit and trust he created makes this institution unique. I am thankful to his successor Dr. Christoph Günther for his ongoing support and opportunity to continue in this fascinating field of research.

I am deeply indebted to my supervisors Professor Lindner (University of Ulm) and Professor Eberspächer (Technical University of Munich) for their advice and scientific guidance. The discussions in Ulm and Munich have been invaluable and helped to steer me towards worthwhile research goals. Furthermore, I want to thank Professor Unger and Professor Schuhmacher for their participation in the committee for my doctoral colloquium.

My family has always been providing the unconditional backing without which prevailing over the ups and downs of research would not have been possible. This work would not even have been started without my parents, whose wisdom and everlasting support could not be greater. My wife Anne has never lost faith, always stood by me, gave me hope and inspiration. To her I owe everything.

Michael Angermann
Oberpfaffenhofen, December 2004

Contents

Chapter 1

Introduction

1.1 Background

Today the Internet is one of the most frequently used infrastructures in many people's daily life. Though its social and economic impacts are far from being fully understood its technological foundations have reached a certain state of maturity and stability. Since its creation by the pioneers of packet switched networks and the introduction of the TCP/IP protocol suite, which still constitutes the technical basis of today's Internet, there have been two dominant applications that created formidable benefit as well as tremendous increases in bandwidth demand. Electronic mail (e-mail) was the first application of the Internet that proved to be useful and convenient and has been well accepted by many users. In its early stages e-mail was used almost exclusively by the scientific community that had access to the necessary network infrastructure. The asynchronous nature of e-mail made it a well suited tool for the worldwide discussion and dissemination of scientific results across different time-zones. The plain text character of the initial e-mail was soon to be enhanced to transport embedded data such as drawings, pictures or program code. Today the proliferation of e-mail has made it a natural form of communication for millions of users in business and countless private matters.

The World Wide Web (WWW) started as a project in 1989 under the leadership of Tim Berners-Lee at the European Organization for Nuclear Research[1] (CERN) located in Geneva, Switzerland. The purpose of the initial WWW project was to create a tool for sharing the tremendous amounts if information that are produced within CERN's typical large scale physics projects. On some of these projects over 1000 people collaborate for periods of 10 to 15 years. While the physical experiments take place in the Geneva laboratory most of the involved researchers are working in their offices at their home institutions which are geographically distributed throughout the world. The approach taken by the designers of the WWW protocols was to allow for decentralized creation and storage of documents. Documents can reference other

[1]It is interesting to note that the slogan of CERN, whose 2500 employees' main mission is to perform research with large particle colliders is *"CERN, where the Web was born"* [CER]

documents by so called *hyper-links*. Upon activation by the user these hyperlinks lead to automatic retrieval and display of the referenced document. Referenced documents do not have to be stored at the same physical location as the referencing document. Thus an automatic library system was created where referenced documents could be obtained within seconds instead of days or weeks. This convenience in combination with the simplicity of generating hyperlinked documents made the WWW the ideal tool for scientific publication. Today it complements and may eventually replace the traditional forms of scientific publication such as journals and to some extent conferences as primary media in many fields of research. The ability to instantaneously publish virtually any kind of information at almost no cost and effort has been creating a wealth of accessible information, provided by organizations and individuals, on all kinds of topics. The significance of this ability is perhaps comparable to the invention of the letterpress with flexible letters around 1450 by Johannes Gutenberg, which enabled the printing of books at much lower cost than the traditional form of copying books by handwriting. Today the WWW has become the most powerful source of information on all issues, upon which scientific topics represent only a small fraction. The WWW's use for general information distribution and retrieval has by far outgrown the scientific application.

Though the WWW has reduced the time that is necessary to retrieve information from days to seconds was frequently nicknamed *World Wide Wait* in its early days, when users were impatient to wait for more than a few seconds. These waiting times were caused by sometimes busy servers and the finite transportation capacity of the network's communication links. Great efforts have been taken to reduce these waiting times by increasing the capacity of the network links. Nevertheless, users still experience waiting times. The reason is the automatic bandwidth occupation effect[2] that usually takes place with all forms of communication means: The higher the capacity of the link the more sophisticated the data that one attempts to transport. In the beginning of the WWW, documents consisted mainly of lightly formatted text and occasionally small pictures. As more powerful communication links became common, larger, more colorful pictures, then animated pictures, sound and moving pictures of always increasing quality in terms of resolution and frame rates proliferated. As a result the waiting times remain fairly constant. In the future we might see documents that have the quality of today's three-dimensional computer games and will eventually reach a state that is comparable to today's high-end virtual-reality applications.

Apart from the automatic bandwidth occupation effect another, more recent, development makes the occupation with improving the communication links worthwhile. The desire to have convenient access to information not only when sitting in front of

[2]This automatic bandwidth occupation effect is probably the main reason why the work of the communication engineer is never finished but resembles the daily work of Sisyphos, which does not necessarily imply that the communications engineer is a tragic figure. We can also consider him lucky, as Albert Camus has put it: "Il faut s'imaginer Sisyphe heureux." (we have to imagine Sisyphos as being lucky), because Sisyphos' existence is never void, but entirely filled with the occupation of rolling the rock uphill.

a computer terminal in an office or at home, but in all kinds of situations and places, has led to the task to make the WWW and related services available on (small) mobile devices that can be used without fixed wire connections but with wireless communication links typically provided by mobile cellular communication networks.

Due to the fact that the physical properties of the wireless communication channel imply a sharing of the (limited) frequency spectrum among the communication links, the available transportation capacity of the wireless links is typically one to three orders of magnitude below the capacity of comparable fixed wire connections. The result for the user is undesired waiting in situations where patience is frequently even more unlikely than when sitting in front of a desktop computer. It is the primary objective of this research to develop and analyze a suitable concept to significantly reduce this undesired waiting.

1.2 Contributions and Structure of this Thesis

At the beginning of this thesis stood a straightforward idea with the potential to improve a mobile wireless communication system's performance as it is perceived by its users. Based on the following three assumptions it should be beneficial to combine several synergetic techniques and concepts:

a) the advances in the fields of ubiquitous and pervasive computing, particularly the work on deriving and providing context information from sensor data and observation of user interaction, will lead to the availability of increasing information upon the user's behavior (*assumption of context awareness*).

b) the dominating application, in terms of transferred data volume, is information retrieval with a hypermedia-oriented structuring of data (*assumption of a hypermedia system*).

c) the mobile device is equipped with multiple and heterogeneous wireless access technologies, differing e.g. in terms of range, data-rate, costs and multiple wireless access networks that are deployed with partial coverage (*assumption of heterogeneous wireless networks*).

Prefetching of documents is known to reduce the perceived latency of hypermedia systems, such as the World Wide Web. It is conjectured that prefetching is especially advantageous whenever the communication links are time-variant in terms of cost and performance. This is the case if heterogeneous wireless access networks are assumed. Since the future choice of a document by the user is not entirely known to the system, prefetching is an inherently probabilistic process. However, the probability of a certain document, becoming selected in the near future, strongly depends on the context of the user. It is, therefore, further conjectured that the more information about the user's context is known and used as a condition that determines a probability mass function for the candidate documents, the better the prefetching process will become, thus resulting in improved perceived performance by the user.

In short, we want to suggest and investigate the use of knowledge about the user's context and behavior to pro-actively transmit data over the wireless communication link before the user requests it.

Substantial research has been carried out on some of the individual techniques and concepts that we intend to combine. An overview on previous research in these topics that have the closest relation to our work, namely context awareness (Section 1.3.1) and prefetching (Section 1.3.2) is given at the end of this introductory chapter.

So far, previous research has treated the techniques from an isolated perspective, without considering their interdependencies and the special conditions imposed by the assumption of heterogeneous wireless networks. The research performed within this thesis is intended to contribute to an improved understanding of these special conditions and interdependencies. Furthermore it is intended to improve the individual techniques and our understanding of them, wherever new questions arise when looked upon them from an overall perspective of our suggested system.

The actual research process during this thesis consisted of continuous iteration over concepts, analysis and implementation. Hence the contributions are both on a conceptual and analytical level as well as including implementations and the insights gained on implementation aspects.
This dissertation is intended to document our insights and results and is structured as follows:

The chapter on **concepts and theoretical aspects** (Chapter 2) starts with the presentation of our novel model for representing user context and actions (Section 2.1). In contrast to previous models, this model strongly incorporates the probabilistic nature of human behavior. Though the presented model is well suited for prefetching purposes, it is kept general to allow also for other kinds of context-aware applications. Various aspects for adaptation of the model to its application domain as well as an investigation of the estimation process for continuous adaptation of the model's internal probabilities are presented. We proceed with an in-depth analysis of prefetching and its influence on perceived performance and traffic (Section 2.2). The obtained analytical results have been a significant contribution to the general understanding of prefetching. On this basis the dependencies between user policies and threshold probabilities are discussed.

We are especially interested in understanding the relation between situation-aware prefetching, user mobility and network heterogeneity. Therefore, we start the chapter on **system simulation** (Chapter 3) by presenting the employed network model (Section 3.1), our novel mobility model (Section 3.2) for realistically generating users' paths and speeds as well as the document and traffic model (Section 3.3). The results of Monte-Carlo simulations based on these models are presented and discussed for a single user, classical mobile network scenario as well as various multi-user, homogenous and heterogeneous network scenarios (Section 3.4).

Since the simulation results indicate that situation-aware prefetching does significantly improve the perceived performance of a hypertext system, especially in combination with a heterogeneous wireless access network, our investigation is extended to several **implementation aspects**, presented in Chapter 4. Based on a discussion of the relevant conditions and constraints in today's mobile networks and devices (Section 4.1) an architecture to enable situation-aware prefetching and the use of multiple radio networks is proposed (Section 4.2). Software components have been developed, integrated and tested (Section 4.3), following the proposed system architecture. The experiences obtained during the phases of deployment and initial operations (Section 4.4) conclude the discussion of the implementation aspects of our proposed concepts.

Chapter 5, **conclusions and outlook**, wraps up this thesis with a summary of our insights and conclusions (Section 5.1) and gives a brief outlook (Section 5.2) on future problems, research and developments.

1.3 Related Work

The multidisciplinary character of our research makes it necessary to briefly discuss work from related fields. Previous research in representing user context or situation has been predominantly performed under the research agenda of ubiquitous or pervasive computing. **Satyanarayanan** has compiled a fairly comprehensive taxonomy for research problems in pervasive computing [Sat01], which is depicted in Fig. 1.1.

Among the many aspects mentioned in this taxonomy that are of relevance to our work such as *mobile networking, mobile information access, adaptivity of applications, energy-awareness* and *context-awareness* we consider it potentially helpful to acquaint the reader with the state of previous research on context awareness (Section 1.3.1), since a considerable part of this thesis is concerned with this fairly new research topic. Furthermore we will give an overview on the long-established research on improving transfer *between* computer systems by means of caching and prefetching (Section 1.3.2).

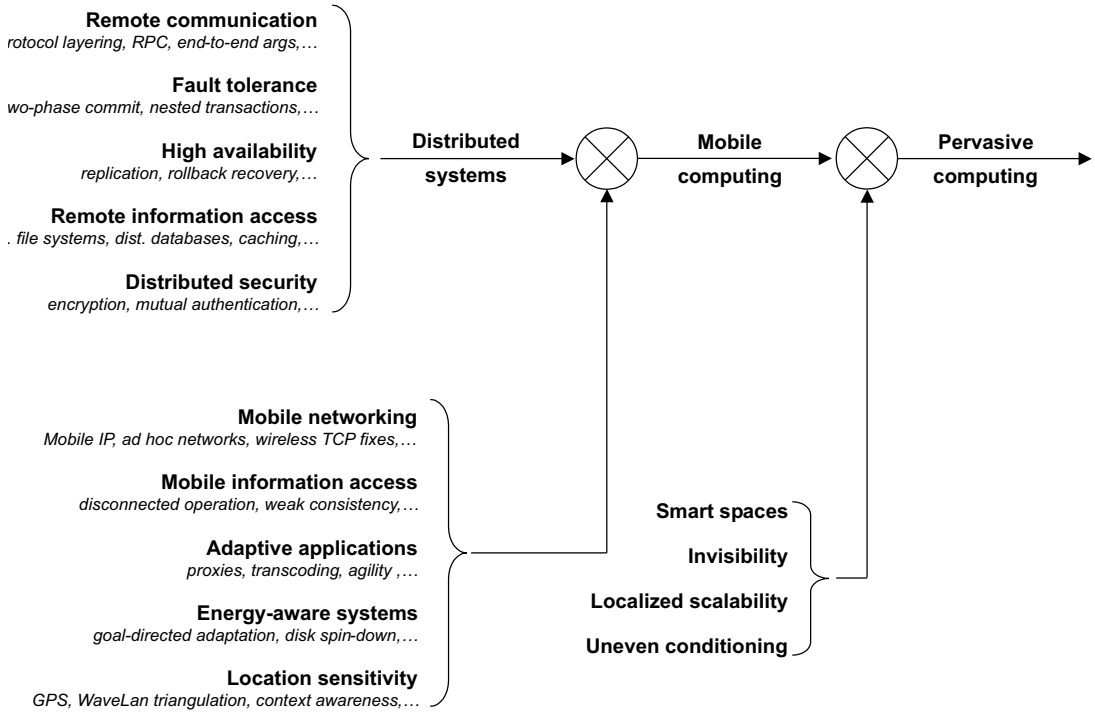


Figure 1.1: *Taxonomy of research problems in pervasive computing according to Satyanarayanan [Sat01].*

1.3.1 Context Awareness

In the early days of computing one expected a computer system to (re-)act always in the same way, regardless of the time of day, the weather or our degree of precipitance. A networked computer may sometimes have made us wait longer or shorter depending on the current network condition but a program's flow was not altered regularly by any of these surrounding conditions, with the exception of the error-case. Personalization of user accounts resulted in, for example, distinct look and feels of user interfaces or time intervals for checking our mailboxes. Yet, these preferences consisted mostly of time-independent parameters of programs, which were rarely changed by the user or administrator of the system.

Advances in miniaturization of computer and communications hardware made it possible to develop small portable computers as well as (embedded) computers that may reside – typically invisibly – and in large numbers in the physical environment. The changes that these developments would have on the interaction between human users and computers were first envisioned by **Mark Weiser**, to whom the vision of *ubiquitous computing* in generally accredited [Wei91, Wei93]. Early work on ubiquitous computing and context awareness started by **Want** et al. under Weiser's supervision as early as 1988 at Xerox PARC laboratories and Olivetti Research Labs and

resulted in early systems such as the *active badge location system* [WHFG92]. This system incorporated small low-cost IR emitters, worn as badges, sensors deployed in the environment (offices, common areas, major corridors)), delivering sightings of badges to a "network master", i.e. location server via a "telemetry network". Among the many applications proposed for this system has been intelligent call forwarding, which is still the canonical example for context-aware systems. Interestingly, Want et al. already raised the privacy issue of such systems. They are especially concerned about the misuse of such a system by employers. However, they come to the conclusion that "a company that has a bad management policy can, of course make life unpleasant for employees with or without a badge system...", and point out that "legislation must be drawn up to ensure a location system cannot be misused, ...". We share this view.
Schilit, **Adams** and **Want** further continued this work and introduced the concepts of *proximate selection* and *context-triggered actions*, where "proximate selection is a user interface technique where the located objects that are nearby are emphasized or otherwise made easier to choose" and "context-triggered actions are simple IF-THEN rules used to specify how context-aware systems should adapt. Information about context-of-use in a condition clause triggers consequent commands." [SAW][WSA+95]

The context information that the applications in these early systems used, was typically limited to a few aspects such as location, proximity to another object or time. **Dey** identified the lack of a framework that helped to abstract the provisioning of context by context sensors from the view of the applications that consumed the context information. He analyzed the design process for context-aware applications and derived a set of requirements and abstractions, resulting in a framework and implementation on top of which numerous applications have been built and demonstrated. Furthermore, Dey's definition of context is the most frequently used in literature:
"*any information that can be used to characterize the situation of an entity. An entity is person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves*" [Dey00].

Extensive work on ubiquitous computing, including many aspects of context awareness such as its acquisition, distribution and use has been performed by **Schmidt** [Sch02]. Schmidt analyzes and experiments with a multitude of "low-level physical sensors" and introduces several abstractions in order to define a "flexible context acquisition architecture". For further research activity in the field see [CK00], a rather comprehensive survey with a focus on experiments and projects until 2000, compiled by **Chen** and **Kotz**.

Numerous interesting ideas are introduced in a concept paper by **Rahlff**, **Rolfsen** and **Herstad** [RRH01]. They propose the "continuous logging" of users' "personal context" which, according to their definition "can be defined briefly as a snapshot of the state on the most important situational parameters: personal identification, time, location, task at hand, nearby objects, nearby people etc." This logging results in a "trace in the multidimensional context space". Various modes of interaction are enabled by such traces. A first mode only involves a user's personal trace and allows

to answer questions like: "When was I here last?" or "Where did I go from this place last time I was here?". Another mode is possible "... if access to other people's traces is granted ..." and enables one to answer questions like: "What do people usually do around here?" or "Where are the nearest people interested in "art"?" The mode of interaction that has the closest relation to our work is an autonomous mode where "the contexts that are closer to your own current context [...] according to some clustering metric where each context field has some predefined "closeness-delta" are assumed to be most relevant for the user [...]. These can be the user's own earlier context, or the contexts of people being simultaneously at the same location, for example." Hence they envision that this technique "may lead to serendipitous discovery where, for example, you suddenly detect the contextual "presence" of somebody else working simultaneously on the same task as you, and going to the same meeting tomorrow."

Jameson [Jam01] points out the importance "to consider, simultaneously, both the user's context and all of the properties of the users themselves...". He uses a concise, yet very illuminating graphical notation to structure the influence of "...different types of information about a user..." on a system's decisions. In slight contrast to our own model, Jameson distinguishes between what he calls "features of the situation", e.g. *U*'s [3] location, "the current state of *U*", e.g. his emotional arousal and "the longer term properties of *U*", e.g. his personal interests, whereas we subsume all this information to the user's situation. However, we absolutely agree to Jameson's view on the importance of recognizing the probabilistic character of context information. Jameson recognizes and explicitly points out: "Much of the evidence that a system *S* can obtain about *U*'s current situation and/or psychological state is unreliable. Often it is only on the basis of multiple pieces of evidence that *S* can make a useful (though still uncertain) inference."

1.3.2 Prefetching and Caching in Networks

The sudden increase in traffic, during the early stages of the World Wide Web, frequently stressed the transport capacity of the internet's data links. While electronic mail had been an asynchronous type of communication, user's were now quickly becoming annoyed when they had to wait, due to the synchronous nature of hypertext document retrieval. However, a fairly large proportion of the documents in a hypermedia system is accessed multiple times by the same user, since users navigate back and forth between documents. Furthermore, user's residing on the same premises, e.g. office or laboratory, frequently access the same documents, which are therefore transferred multiple times over the network. A natural conclusion for researchers, thriving to improve the situation, was to apply *caching*, which had been a well-known technique, typically applied for improving the performance of communication *within*[4]

[3] Jameson uses *U* to denote the User and *S* to denote the system

[4] It is interesting to notice, that with the advent of the concepts of distributed computing the lines between caching and prefetching, disk storage, network access etc. become blurry. If we consider

computers, between storage, memory and CPU, for the communication *between* computers. We shall use the insights of this previous research on caching and later prefetching to extend the application of these techniques to our scenarios of mobile information access over heterogeneous wireless networks.
Since today's mobile communication networks achieve similar data-rates to fixed telephone-line modem connections about 5-8 years ago, the observations made by **Fan** et al. provide insights that can be transferred to the mobile scenarios [FCLJ99]. They list empirical results for trace-driven simulation of prefetching using a *prediction by partial matching (PPM) algorithm* [CW84]. The HTTP-traffic traces are obtained from a University of Berkeley home dial-up population from November 1996, with data-rates around 20 Kb/s, which lies within the range of today's mobile networks. They report that "prefetching combined with large browser cache and delta-compression can reduce user-perceived latency up to 23.3%. In contrast to their assumption of a large browser cache we are particularly considering small caches into which only a small number of documents is prefetched. Due to knowledge about documents' probilities our simulation results show similar improvements in user-perceived latency, without the need for extensive on-device memory.

Vitter and **Krishnan** observe the analogies between the selection of documents for prefetching and problems arising in data compression [VK96]. They apply the Lempel-Ziv data compression algorithm to derive an "optimal universal prefetcher in terms of fault rate" and derive bounds for the fault rate for their algorithm based on Markov source models. In their paper it is pointed out that "in many hypertext and iterative database systems, there is often sufficient time between user requests to prefetch as many pages as wanted, limited only by the cache size". They "refer to prefetching in this setting as *pure prefetching*". In contrast to the work performed later by Jiang and Kleinrock [JK97, JK98], Tuah [TKV99, Tua00] and in this thesis, they restrict their analysis to this type of "pure prefetching".

Crovella and **Barford** investigate the influence of prefetching on the *burstiness* of the generated traffic [CB98]. Their investigation is based on a simulated network, consisting of 64 clients connected to one router and two servers connected to a second router, with a dominant bottleneck between the two routers. They show "that prefetching as it is usually implemented – that is, the transfer of multiple files together in advance of request – can create an undesirable increase in burstiness of individual sources. [...] Increases in source burstiness result in increases in variability of aggregate traffic at a wide range of scales. This makes straightforward approaches

the classical von Neumann architecture for a computer, consisting of a processing unit (CPU), fast (RAM) and slow memory (e.g. disk storage) and a communication link (bus) connecting them, we see that prefetching and caching show almost a fractal character in a highly distributed system where e.g. the responsibility for data persistence is delegated to a database subsystem which itself consists of numerous hosts and disk-arrays which themselves consist of memory and numerous CPUs which have various caches on the actual silicon die. From this perspective it may be favorable to think in more general terms of *dynamic allocation of data* to a system's components. An overview of various techniques that have been investigated and applied for improving data transfer *within* computer systems is given in Appendix C.

to prefetching [...] less attractive from the standpoint of network performance." They emphasize the benefits of *rate-controlled prefetching*, which in effect decreases the individual burstiness. "As a result, applications employing rate-controlled prefetching can have the best of both worlds: data transfer in advance of user request, and better network performance than is possible without prefetching."

Jiang and **Kleinrock** derive important theoretic results for prefetching in [JK97, JK98]. Recognizing "the tradeoff between system resource usage and latency," they "choose to measure the system performance in terms of cost which is comprised of the delay cost [...] and the system resource cost [...]. The delay cost indicates how valuable the time is to the user. The system resource cost includes the cost of processing the packets at the end nodes and that of transmitting them from the source to the destination.". They derive optimum threshold probabilities for prefetching in single-user and multi-user scenarios that minimize these costs.

Apart form Jiang and Kleinrock, **Tuah**'s comprehensive and thorough theoretical analysis [TKV99, Tua00] is the only work with an in-depth modelling of prefetching performance. Corresponding to Jiang's and Kleinrock's delay costs and system resource costs, Tuah defines "access improvement" and "excess retrieval cost". The combination of caching and prefetching is treated as a "stretch knapsack" optimization problem.

Various further publication provide a valuable source on theoretical and practical problems related to prefetching [IX00, HREM99, YZL01, TLAC95, Zha01, Dav01].

Chapter 2

Concepts and Theoretical Aspects

We start the discussion of our proposed scheme for situation awareness and its application to prefetching for improving mobile information access with a number of theoretical considerations. The discussion is split into three sections: In the first section (2.1) we introduce our **situation model** and the structure of a **situation space** (2.1.1), which is then extended towards a probabilistic model for dynamic (2.1.2) and observable (2.1.3) behavior. **Information theory** helps us to derive guidelines for the construction of a sensible situation space (2.1.4). Since the purpose of the situation model is to predict the future behavior of a user, estimation methods for **continuous adjustment of the model probabilities** are discussed (2.1.5). In the second section (2.2), the prediction enabled by the situation model is then applied to the **target domain of prefetching**. For this purpose, we present an analytical model (2.2.1) for the discussion on the theoretical performance of prefetching in terms of **reduction of waiting time** (2.2.2), **additional traffic** (2.2.3) and the influence of **user policies** and **probability thresholds** (2.2.4) under the assumption of a classical mobile networking scenario.

2.1 Situation Model

A user of any communication or information service, be it via a fixed or wireless medium, is influenced by the circumstances that surround him or her. Research has begun to investigate the use of information on these circumstances, e.g. location, for improving the provision of information services about 10 years ago. A considerable number of publications treating the topic, usually termed "context awareness", has already evolved. Currently, most research is performed on a computer science background on the semantic aspects of context. Proposed are a multitude of ontologies, languages and dialects for the purpose of representing and exchanging context information among software instances. Based on the specified representations algorithms that perform automatic logical reasoning are intended to derive sensible actions that a technical system should take under the given circumstances.

In many domains in which context awareness is proposed, e.g. booking or reservation of tickets for transportation or entrance, procurement of goods etc., the system is usually not given the authority to autonomously take actions. Instead, context information is used to make proposals to the user.

In this work we will present our concept of *situation* and *situation awareness*. In contrast to more general approaches of context, this concept is intended for improving the quantifiable performance of applications that typically involve frequently occurring events in usually highly specialized domains, such as transportation of content over communication networks, handover in wireless communication or ranking of search results. In these domains a system can be entrusted with the authority to autonomously decide upon and take actions, since the potentially adverse consequences of single actions are small and it can be shown that the sum of actions will lead to significant improvement of quantifiable performance *on the average*. This increased degree of autonomy is important, since often the frequency of necessary decisions/actions prohibit an active involvement of the user in the process.

We will use the terms *situation* and *situation awareness* within the description of our concept, in contrast to the more general concept of context and context awareness. The concept of situation-awareness defines and uses a formal model to represent and employ context information, particularly considering the inherent probabilistic character of context.

We will now start this section with an introduction of our concept of situation space and an investigation of its properties. Its application for frequently occurring events is facilitated by introducing stochastic transitions between situations. We will then drop the assumption of a priori known transition probabilities and investigate the estimation of these probabilities from a Bayesian perspective.

2.1.1 Situation Space

The canonical example for context information is the domain of location or position information. Within this domain the concept of *space* and its properties such as metrics for distance are well defined and frequently utilized. Similarly to a person's movements in geometric space, the perpetual change of situation in our daily life can be interpreted as *moving through a space of situations*.
In the following we will introduce our concept of a *situation space*.

Whenever natural language is used to describe a situation, a description is composed of one or more statements. A possible description might be: "*A person is at the airport for business purposes and ahead of his schedule.*"
To represent and use this information within a technical system some structuring is necessary.

The description of this particular person's situation can be decomposed into three "atomic" statements:

(a) the person is at the airport.

(b) the person is in a business context.

(c) the person is early.

Each of these three statements illuminates one aspect of the person's current situation. The statement *"the person is in a business context"* informs on an aspect we might call "sphere". *"The person is in private context"* would be another possible statement on the aspect of "sphere".

It is straightforward to define a formalism that can be used quite intuitively. We say that an *aspect* consists of the set of possible statements, each of which partially characterizes a situation. Each statement is in turn an element of this aspect. We term these statements *components* of the respective aspect. The two statements *"the person is in a business context"* and *"the person is in private context"* are components of the aspect "sphere".

In order to establish a terminology for the future discussion we define some helpful terms before illustrating their use. We start with a quasi-formal definition of the previously used term *aspect*:

Definition 2.1 (Aspect) *A set of mutually exclusive statements about the circumstances of an entity is termed an aspect* Γ*.*

Since an aspects is now defined to be a set, we can proceed with a more formal definition for this set's elements.

Definition 2.2 (Component) *The* N_{Γ_j} *elements of an aspect* Γ_j *are termed components* $\gamma_{j,k}, k = 1, \ldots, N_{\Gamma_j}$ *of this aspect.*

Airport Example:

The airport example helps to illustrate these abstract definitions. For the sake of conciseness we consider only the aspects "sphere" and "schedule" and abbreviate the statements.
Hence, the two aspects are

$$\Gamma_1 = \{\text{private}, \text{business}\} \quad (\text{"sphere"}),$$

$$\Gamma_2 = \{\text{early}, \text{late}\} \quad (\text{"schedule"}),$$

with $N_{\Gamma_1} = 2$ and $N_{\Gamma_2} = 2$.

Furthermore, we define the *fundamental set of the situation space*,

Definition 2.3 (Fundamental Set of the Situation Space) *The Cartesian product* $\Gamma_1 \times \Gamma_2 \times \ldots \times \Gamma_N$ *of the* N_Γ *aspects* $\Gamma_j, j = 1, \ldots, N_\Gamma$ *generates the fundamental set* X_σ *of the situation space.*

$$X_\sigma = \Gamma_1 \times \Gamma_2 \times \ldots \times \Gamma_N \tag{2.1}$$

and arrive at our definition of *situation*:

Definition 2.4 (Situation) *An arbitrary subset of the fundamental set* X_σ *is termed* situation σ_i.

$$\sigma_i \subset X_\sigma \quad \forall\, i. \tag{2.2}$$

Since the set of all subsets of the fundamental set X_σ is its associated *power set* $P(X_\sigma)$, the number N_σ of possible situations can be easily computed. The cardinality of a power set $P(X_\sigma)$ is

$$N_\sigma = |P(X_\sigma)| = 2^{|X_\sigma|}, \tag{2.3}$$

which is the number of possible situations in our situation space.

Definition 2.5 (Elementary Situation) *If a situation* σ_i *cannot be decomposed, i.e.* $\sigma_i \in X_\sigma$*, it is termed* elementary *situation. We may use a raised asterisk to indicate this property:* σ_i^*

It follows that

$$\sigma_i^* \ \wedge \ \sigma_j^* = \emptyset \quad \forall\, i \neq j. \tag{2.4}$$

The number N_{σ^*} of elementary situations is determined by the cardinality of all N_Γ aspects which also determines the cardinality of X_σ.

$$N_{\sigma^*} = \prod_{i=1}^{N_\Gamma} |\Gamma_i| = |X_\sigma| \tag{2.5}$$

Airport Example (continued):
Their Cartesian product $\Gamma_1 \times \Gamma_2$ generates the fundamental set

$$X_\sigma = \{ \quad (\text{private}, \text{early}), (\text{private}, \text{late}), \\ (\text{business}, \text{early}), (\text{business}, \text{late}) \quad \}.$$

We see that the cardinality of X_σ is 4, which is also computable by eq. 2.5:

$$|X_\sigma| = |\Gamma_1| \cdot |\Gamma_2| = 2 \cdot 2 = 4$$

This results in $N_\sigma = |P(X_\sigma)| = 2^{|X_\sigma|} = 2^{|4|} = 16$ possible situations, i.e. subsets of X_σ,
(We use the abbreviation *p, b, e* and *l*, for *private, business, early* and *late*.)

$$\begin{aligned}
\sigma_0 &= \emptyset \\
\sigma_1^* &= \{(\text{p}, \text{e})\} \\
\sigma_2^* &= \{(\text{p}, \text{l})\} \\
\sigma_3^* &= \{(\text{b}, \text{e})\} \\
\sigma_4^* &= \{(\text{b}, \text{l})\} \\
\sigma_5 &= \{(\text{p}, \text{e}), (\text{p}, \text{l})\} \\
\sigma_6 &= \{(\text{p}, \text{e}), (\text{b}, \text{e})\} \\
\sigma_7 &= \{(\text{p}, \text{e}), (\text{b}, \text{l})\} \\
\sigma_8 &= \{(\text{p}, \text{l}), (\text{b}, \text{e})\} \\
\sigma_9 &= \{(\text{p}, \text{l}), (\text{b}, \text{l})\} \\
\sigma_{10} &= \{(\text{b}, \text{e}), (\text{b}, \text{l})\} \\
\sigma_{11} &= \{(\text{p}, \text{e}), (\text{p}, \text{l}), (\text{b}, \text{e})\} \\
\sigma_{12} &= \{(\text{p}, \text{e}), (\text{p}, \text{l}), (\text{b}, \text{l})\} \\
\sigma_{13} &= \{(\text{p}, \text{e}), (\text{b}, \text{e}), (\text{b}, \text{l})\} \\
\sigma_{14} &= \{(\text{p}, \text{l}), (\text{b}, \text{e}), (\text{b}, \text{l})\} \\
\sigma_{15} &= X_\sigma
\end{aligned}$$

of which σ_1^* to σ_4^* are elementary situations.

While we now have a definition of elementary and non-elementary situations, we are still lacking an *interpretation* of these constructs to make them useful. In the following we will again refer to the airport example to demonstrate the use of these elementary and non-elementary situations to express different degrees of available knowledge.

The interpretation of elementary situations $\sigma_1^* \dots \sigma_4^*$ is straightforward. Each of these elementary situations expresses the complete knowledge of the person's circumstances within the situation space. Whenever only *partial knowledge* about the person's circumstances is available, non-elementary situations are used. Situation $\sigma_5 = \{(\mathrm{p}, \mathrm{e}), (\mathrm{p}, \mathrm{l})\}$ expresses the partial knowledge that the person is in private context, without any knowledge about whether the person is late or early. The person may either be in situation σ_1^* **OR** σ_2^*; it is not known in which one of these two. This can also be expressed by a formal statement:

$$\sigma_5 = \sigma_1^* \cup \sigma_2^*$$

The information that a person is in situation $\sigma_{11} = \{(\mathrm{p}, \mathrm{e}), (\mathrm{p}, \mathrm{l}), (\mathrm{b}, \mathrm{e})\}$ reveals even less information. This statement only tells us that the person is **NOT** both late and in a business context, which can be formulated as

$$\sigma_{11} = \overline{\sigma_4^*}.$$

The structure and operations can be illustrated by Venn diagrams. Fig. 2.1 shows a Venn diagram of the situation space for the example.

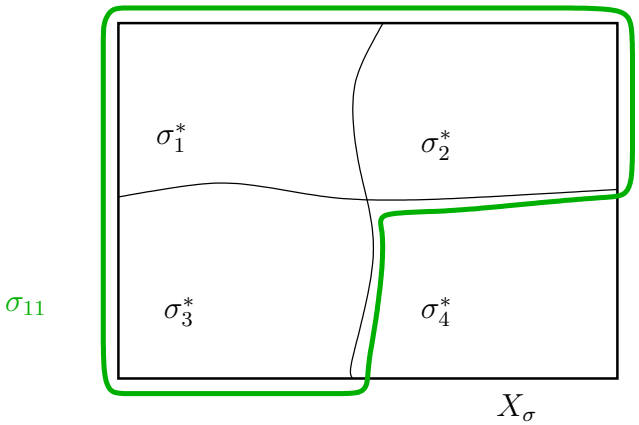


Figure 2.1: *The Venn diagram illustrates the interpretation of situation σ_{11} as the incomplete knowledge that the person is not in the elementary situation $\sigma_4^* = \{(business, late)\}$.*

The structure and interpretation of the information space also facilitates the combination of knowledge on the person's circumstances, e.g. stemming from distinct sources. If one source states that the person is in situation σ_7 and a second source expresses its knowledge that the person is in situation σ_{11}, this information can be combined by an **AND** operation:

$$\sigma_7 \cap \sigma_{11} = (\sigma_1^* \cup \sigma_4^*) \cap (\sigma_1^* \cup \sigma_2^* \cup \sigma_3^*) = \sigma_1^*,$$

yielding the complete information that the person is currently in situation σ_1^*, i.e. in a private context and early in whatever he is doing (see Fig. 2.2).

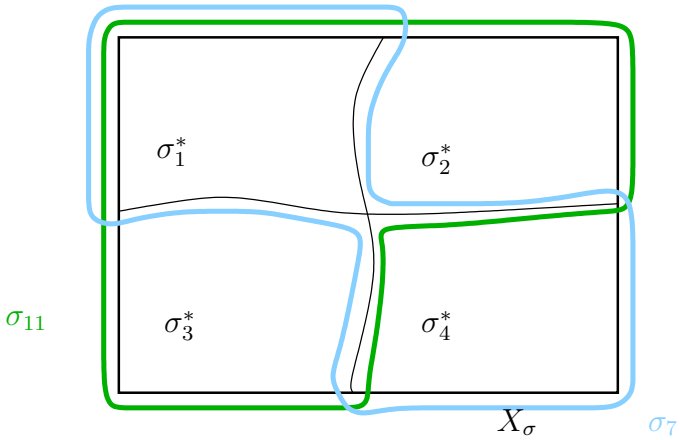


Figure 2.2: *The Venn diagram illustrates the combination of situations σ_7 and σ_{11} to situation σ_1^*, thus yielding complete knowledge.*

Performing an **AND** operation on two or more distinct elementary situation will result in situation $\sigma_0 = \emptyset$, which always indicates the occurrence of contradicting information.

The absence of information is represented by the situation that corresponds to the fundamental set X_σ, i.e. situation σ_{15} in our example.

Since "space" is a term with diverse mathematical definitions it should not be used carelessly. Therefore, we will briefly show that our situation space is an instance of a *topological space* and already equipped with its properties. A topological space is defined by a set X and a topology τ, defined on this set.

Definition 2.6 (Topology) *A collection*[1] *τ of subsets G_i of X ($G_i \subseteq X$) is a* topology *if and only if the following three conditions hold:*

[1]The term "collection" is synonymous with set, but often used in literature, because it is supposed to be less confusing to think of a "collection of sets" instead of a "set of sets".

1. *the complete set X and the empty set $\emptyset$ belong to τ.*

$$X \in \tau, \quad \emptyset \in \tau$$

2. *the union $\bigcup G_i$ of arbitrarily many sets of τ is in τ.*

$$\bigcup G_i \in \tau, \quad \textit{whenever} \quad G_i \subseteq \tau$$

3. *the intersection $\bigcap G_i$ of a finite number of sets is in τ.*

$$\bigcap G_i \in \tau \quad \textit{whenever} \quad G_i \textit{ is a finite subset } \tau.$$

Definition 2.7 (Topological Space) *The pair (X, τ) is called a* topological space.

A comparison of the definition of topology with our definition of situation shows the correspondence between a *fundamental set of the situation space X_σ* (def. 2.3) and a set X (def. 2.6) as well as the correspondence between a collection of situations $\{\sigma_0, \sigma_1, \ldots, \sigma_{N_\sigma-2}, \sigma_{N_\sigma-1}\}$ and a topology τ.

2.1.2 Dynamic Situation Model

In the previous section the concept of situations and aspects has been introduced. These situations are related to each other only by their set-theoretic relations. So far the model does not contain any representation for the "nearness" of one situation to another. If the situation is predominantly determined by the person's position, this nearness is typically expressed by the Euclidian distance of positions. "Location aware services" offered by today's mobile networks usually employ this metric for recommending shops, restaurants, gas stations etc. From common daily experience we know that more general situations may also be closely related to each other, whereas others have almost no relation. A situation in which a user's mobile device has run out of batteries is closely related to a situation in which the same user recharges the device, whereas it is almost not related to a situation in which the user is consuming a sandwich.
Some situations are related to each other in a temporal sequence. Extending the previously used airport example, a typical passenger arrives at the airport, uses his ticket to check the flight number, determines the terminal, checks in baggage, passes security, moves to the gate, waits and finally enters the aircraft. This temporal sequence of situations is typical but not necessarily deterministic, deviations may occur: the passenger might only have hand luggage or the gate may be changed. Even if a sequence is deterministic, it may depend on *indiscernable* circumstances (e.g. the passenger's sudden urge to shop for duty free goods). We therefore propose to treat the succession of situations as a stochastic process. In consequence, we argue that

probability of transition from one situation to another is a suitable metric[2] for representing the "nearness" between these two situations.

Initially, we will focus on the "atoms" of the overall random process, i.e. the transitions from some elementary situation σ_i^* at step k, towards its possible successors $\sigma_1^*, \ldots, \sigma_{N_{\sigma^*}}^*$ at step $k+1$. When observed repeatedly, these isolated transitions form their own random process. Typically, the process is not sampled in the time-domain, instead the event of a transition from one situation towards another marks the steps k.

We use the following shorthand for the N_{σ^*} transition probabilities $p_{i,j}$:

$$p_{i,j} = \Pr\left\{\boldsymbol{\sigma}^*(k+1) = \sigma_j^* \mid \boldsymbol{\sigma}^*(k) = \sigma_i^*\right\}, \quad i,j \in \{1,\ldots,N_{\sigma^*}\} \tag{2.6}$$

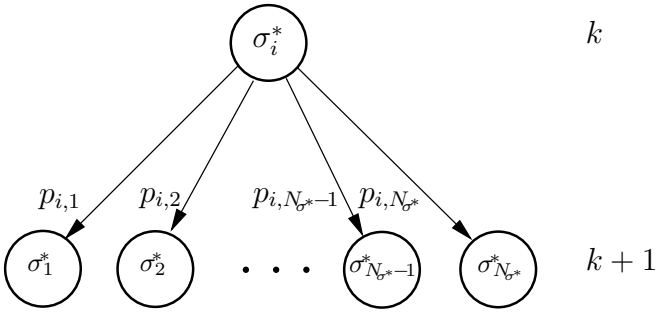
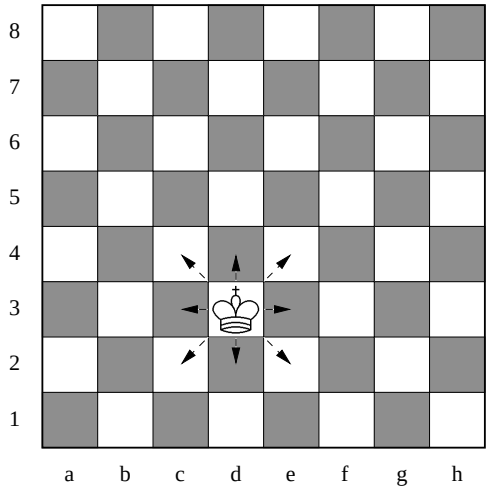


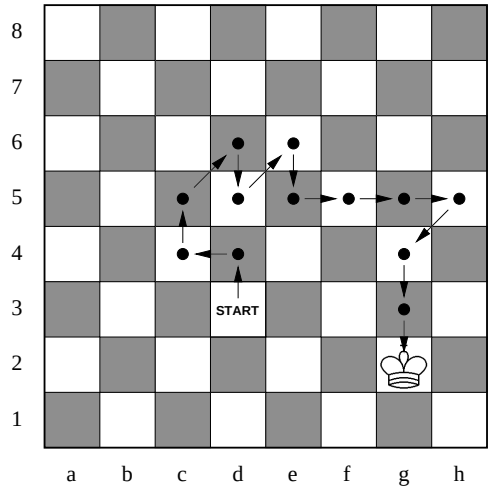
Figure 2.3: *Isolated random transition process from some elementary situation σ_i^* towards its possible successors σ_j^*, $i,j \in \{1,\ldots,N_{\sigma^*}\}$.*

Who or what chooses the transitions in the real world is of no immediate concern for this model. It might be the free will of a person choosing among e.g. different shops to enter or places of interest to visit. If the weather is an *aspect* of the situation it is the chaotic nature of the meteorological processes that might cause and determine a particular transition. Nevertheless, some degree of knowledge about the probabilities of transitions is usually available (obtained e.g. by repeating observations of transitions). The uncertainty about future situations depends on these probabilities and increases the farther we try to look into the future.
We will use the example of a king's random walk on a chess board, for its regularity and simplicity, to illustrate this property of our model. Fig. 2.4(a) shows the allowed moves of the king. The number of allowed moves depends on the position of the king on the board. Eight moves are allowed on the inner 36 fields, five moves at the 24 edge-fields and only three moves at each of the remaining four corner fields.

[2]Here, "metric" is used in a week sense, not following a strict mathematical definition, since the "nearness" of situations is not necessarily symmetric and additionally the triangle inequality does not hold.



(a) Allowed 8 movements of a king on a chess board. Eight moves are allowed on the inner 36 fields, five moves at the 24 edge-fields and only three moves and at each of the remaining four corner fields.



(b) Sample walk of a king starting at an arbitrarily chosen initial situation or field (here: $\{(d, 3)\}$).

Figure 2.4: *Example: A King's random walk on a chessboard. The movements of the king form a random process which is used as an example of a movement through a situation space. The situation space is formed of its two aspects* Γ_1 *("column") and* Γ_2 *("row").*

The 64 possible positions are conveniently represented as elementary situations of a situation space constructed by the Cartesian product of its two aspects Γ_1 ("column") and Γ_2 ("row"):

$$\Gamma_1 = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}, \mathrm{f}, \mathrm{g}, \mathrm{h}\} ,$$

$$\Gamma_2 = \{1, 2, 3, 4, 5, 6, 7, 8\} .$$

The aspects chosen above are just one possibility among arbitrarily many. Numbering the fields from 1 to 64 and assigning them as components to just one aspect would result in an equally correct situation space, constructed of $\Gamma_1 = \{1, 2, \ldots, 64\}$. Fig. 2.5 shows how the elementary situations are related to each other by the possible transitions among them. For each move of the king the future situation is randomly chosen out of the N allowed future situations with equal probability $1/N$. It is apparent that the dynamic situation model is an instance of a homogenous first order Markov chain.

As mentioned earlier, we desire to achieve the ability to pro-actively perform actions that will become necessary in situations lying in the future. Since the future situations are chosen by a random process, the future contains an inherent degree of uncertainty. It is usually impossible or in-efficient to perform all possible future actions that might become necessary in all possible future situations, as most actions result in costs caused by the consumption of some limited resource such as time, money

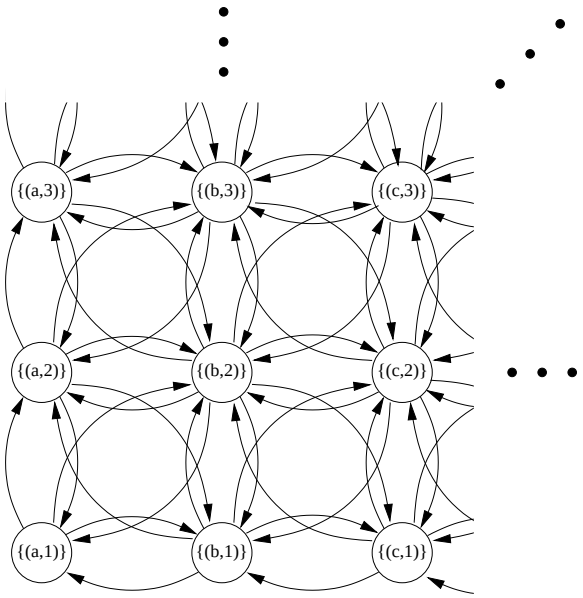


Figure 2.5: *Lower left section of transition graph with allowed transitions between situations for a king's movements (see also Fig. 2.4(a)). The situations and transitions form a homogenous first order Markov chain.*

or energy. Therefore it is attempted to identify a subset of actions that minimizes the overall cost. To choose a proper subset of actions we first have to determine the probability of the future situations given the knowledge about the situation. Fig. 2.6 shows how the probabilities of future situations depend on the temporal horizon. We see that after the first step 8 elementary situations are possibly reached, and after the second step 25 elementary situations are possible. After the first step all 8 situations are equally probable (Fig. 2.6(b)), whereas the 25 possible elementary situations after the second step are not equally probable (Fig. 2.6(c)). Intuitively we conjecture that the uncertainty increases with each step.

The entropy $H(\mathbf{x})$ of a random variable $\mathbf{x}$ is a well established metric for quantifying uncertainty [CT91] which we intend to apply to our field of interest.

If the following abbreviated notation is used for the probability mass function (PMF)

$$p_{\boldsymbol{x}}(x_i) = \Pr\{\boldsymbol{x} = x_i\}, \quad i = 1 \ldots N, \tag{2.7}$$

the entropy of a discrete random variable is defined as follows:

Definition 2.8 (Entropy of a Discrete Random Variable) *The* entropy $H(\boldsymbol{x})$ *of a discrete random variable* $\boldsymbol{x}$ *is defined by*

$$H(\boldsymbol{x}) = -\sum_{i=1}^{N} p_{\boldsymbol{x}}(x_i) \cdot \mathrm{ld}\, p_{\boldsymbol{x}}(x_i) \quad [\mathrm{bit}] \tag{2.8}$$

Since the elementary situation at step k is a random variable $\boldsymbol{\sigma}^*(k)$ with probability mass function $p_{\boldsymbol{\sigma}^*(k)}(\sigma_i^*)$

$$p_{\boldsymbol{\sigma}^*(k)}(\sigma_i^*) = \Pr\left\{\boldsymbol{\sigma}^*(k) = \sigma_i^*\right\}, \quad i = 1 \ldots N_{\sigma^*}, \tag{2.9}$$

we express our degree of uncertainty about the situation after k steps into the future by its entropy $H(\boldsymbol{\sigma}^*(k))$:

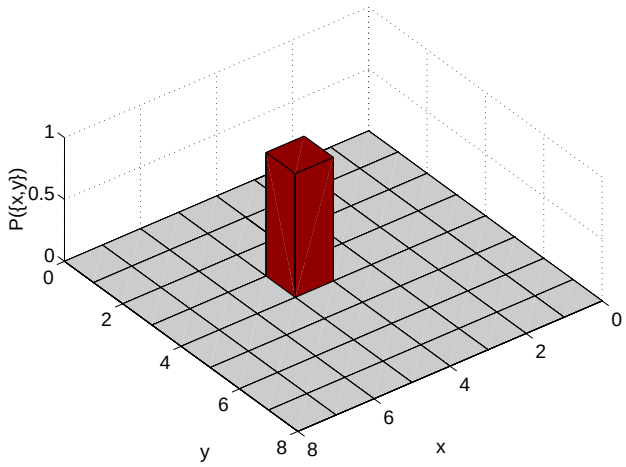
$$H(\boldsymbol{\sigma}^*(k)) = -\sum_{i=1}^{N^*} p_{\boldsymbol{\sigma}^*(k)}(\sigma_i^*) \cdot \mathrm{ld}\, p_{\boldsymbol{\sigma}^*(k)}(\sigma_i^*) \quad [\mathrm{bit}] \tag{2.10}$$

If the situation $\boldsymbol{\sigma}^*(k)$ at time instant k is perfectly known, the uncertainty is 0 bit. The uncertainty about the situation $\boldsymbol{\sigma}^*(k+1)$ after the first step in the chess example is $-8 \cdot 1/8 \cdot \mathrm{ld}\ 1/8 = 3$ bit and rises to 4.4528 bit for $\boldsymbol{\sigma}^*(k+2)$.

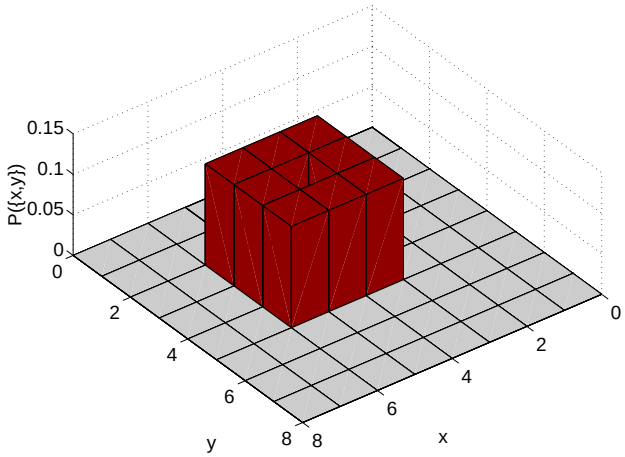
From the fact that the dynamic situation model is a first-order Markov chain, arises the interesting property that the knowledge of the transition probabilities results in a priori information that can be used to reduce the uncertainty even if no particular data about the current or future situation is available. For a long unobserved sequence of steps, the Markov chain becomes quasi-stationary, i.e. the probability to be in a particular situation approaches a stationary probability that can be computed under that assumption of a stationary Markov chain. For the random walk example these stationary probabilities are computed to be 8/420, 5/420, and 3/420 for the center, edge and corner fields, respectively. Fig. 2.7 illustrates how the relative frequencies of visits to the 64 fields approach these stationary probabilities.

Even if no explicit information on the king's current position is available, the uncertainty is only 5.9484 bit, assuming the transition probabilities are known, instead of 6 bit if the transition probabilities were unknown. Fig. 2.8 shows that the uncertainty does not rise any further but remains constant at this maximum value for the following steps.

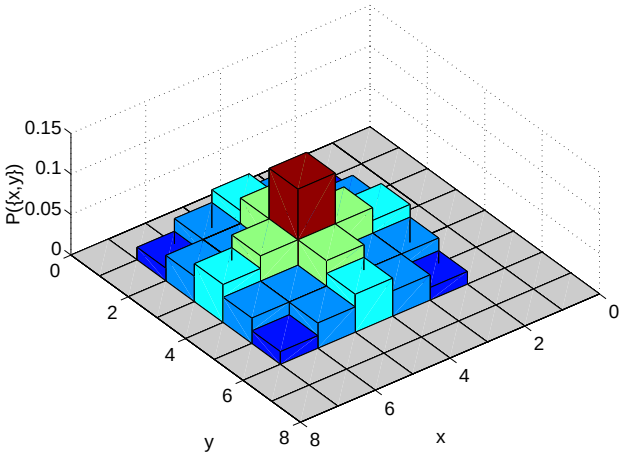
Fig.2.9(a) shows how partial knowledge and the knowledge about the stationary probabilities are combined. In the example only the aspect "row" is assumed to be known ("4"), whereas no explicit information is available on the aspect "column". Hence, the king occupies one of 8 possible fields, which would result in an uncertainty of 3 bits, if the transition probabilities were unknown. If the transition probabilities are assumed to be known, the uncertainty is reduced to 2.9749 bit (see Fig. 2.9(a)). The increase in uncertainty for the following steps is depicted in Figs. 2.9(b) and 2.9(c).



(a) Initial situation $\sigma^*(k=0)$ known, $H(\boldsymbol{\sigma}^*(k=0)) = 0$ bit

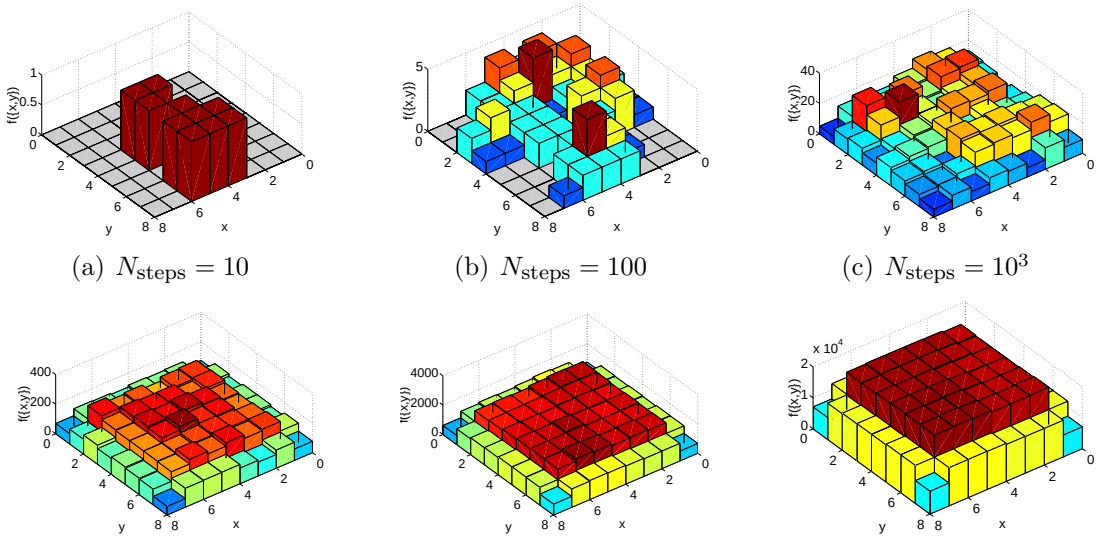


(b) Uncertainty about (future) situation $\sigma^*(k=1)$, $H(\boldsymbol{\sigma}^*(k=1)) = 3$ bit



(c) Uncertainty about (future) situation $\sigma^*(k=2)$, $H(\boldsymbol{\sigma}^*(k=2)) = 4.4528$ bit

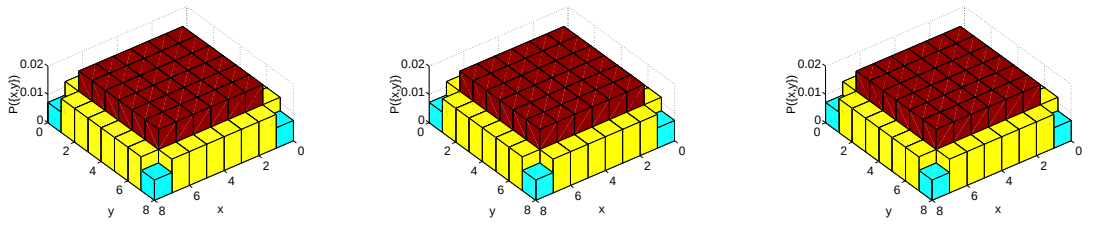
Figure 2.6: *Initially, the king's current field is known, the uncertainty is 0 bit. In the sequence of steps the uncertainty on the king's position increases. First 8 fields are equally probable, resulting in an uncertainty of* 3 *bit, then the uncertainty increases to* 4.4528 *bit, when* 25 *fields are possible with different probabilities.*



(a) $N_{\text{steps}} = 10$ (b) $N_{\text{steps}} = 100$ (c) $N_{\text{steps}} = 10^3$

(d) $N_{\text{steps}} = 10^4$ (e) $N_{\text{steps}} = 10^5$ (f) $N_{\text{steps}} = 10^6$

Figure 2.7: *With increasing number of steps N_{steps}, the relative frequency of visits converges towards the stationary probabilities of* 8/420, 5/420, *and* 3/420 *for the center, edge and corner fields, respectively.*



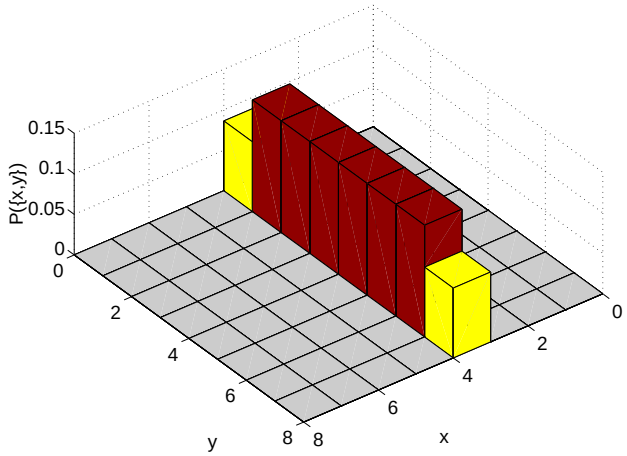
(a) (b) (c)

Initial situation $\sigma^*(k=0)$ unknown, $H(\boldsymbol{\sigma}^*(k=0)) = 5.9484\,\text{bit}$

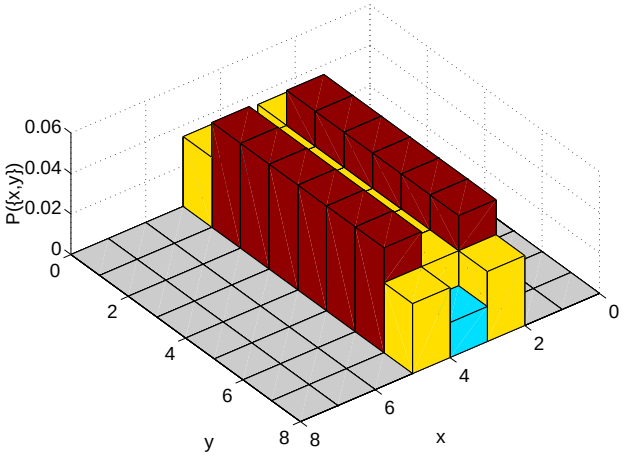
Uncertainty about (future) situation $\sigma^*(k=1)$, $H(\boldsymbol{\sigma}^*(k=1)) = 5.9484\,\text{bit}$

Uncertainty about (future) situation $\sigma^*(k=2)$, $H(\boldsymbol{\sigma}^*(k=2)) = 5.9484\,\text{bit}$

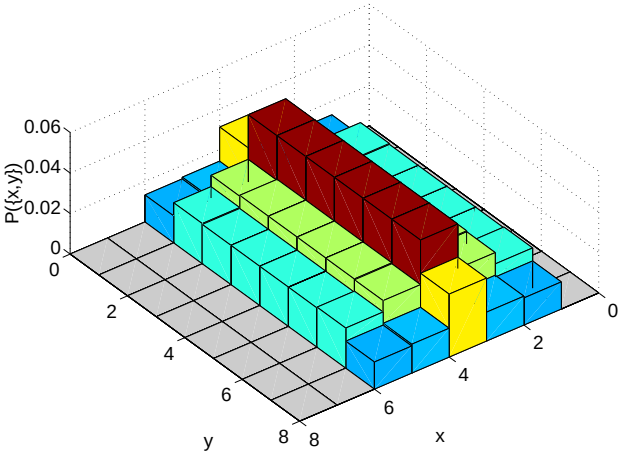
Figure 2.8: *Maximum uncertainty on initial situation.*



(a) Initial situation $\sigma^*(k=0)$, $H(\boldsymbol{\sigma}^*(k=0)) = 2.9749\,\text{bit}$



(b) Uncertainty about (future) situation $\sigma^*(k=1)$, $H(\boldsymbol{\sigma}^*(k=1)) = 4.5294\,\text{bit}$ after first step



(c) Uncertainty about (future) situation $\sigma^*(k=2)$, $H(\boldsymbol{\sigma}^*(k=2)) = 5.2052\,\text{bit}$ after second step

Figure 2.9: *Only the aspect "row" is assumed to be known ("4") at the initial stage. If the transition probabilities are assumed to be known, the uncertainty is 2.9749 bit. The uncertainty increases with the following steps.*

2.1.3 Symptoms and Consequences

2.1.3.1 Symptoms

The model that has been presented, so far, is well suited for representing the abstract concept of a subject, typically a human being, traversing through a space of situations. We can interpret the model as a finite state machine, whose state transitions form a random process. In order to connect the model to the physical world, it is necessary to introduce a representation for *sensor data* to the model (see also [Dey00, Sch02]). For this purpose we interpret sensor data that occurs in conjunction with the transitions between situations as *symptoms* that contain information about the occurred transitions.

Definition 2.9 (Symptom) *A detectable event or data that occurs in conjunction with a situation transition is termed symptom.*

The continuous or discrete values of sensor data are mapped onto symptoms. A typical example would be the mapping of a range of WGS 84 coordinates[3] onto an extended topographical area e.g. a country, city, or place of interest. In this case a GPS receiver might signal the crossing of a particular section of the boundary that defines a place called "Trafalgar Square, London, UK" would be the event "entered Trafalgar Square, London, UK from Whitehall, London, UK".
Detecting a short-ranging wireless access node at Trafalgar Square after leaving the coverage area of another access node at Whitehall would be mapped onto the same symptom.

We distinguish between two subtypes of symptoms.

Definition 2.10 (Situation Specific Symptom) *A symptom that indicates the transition from a situation σ_i to a situation σ_j, with $i \neq j$, is termed situation specific symptom $\alpha_{\sigma_i,\sigma_j}$.*

Definition 2.11 (Component Specific Symptom) *A symptom that indicates the transition from a component $\gamma_{h,i}$ to another component $\gamma_{h,j}$ of the same aspect Γ_h, with $i \neq j$, is termed component specific symptom $\alpha_{\gamma_{h,i},\gamma_{h,j}}$.*

Situation specific symptoms directly indicate the transition to a specific situation, independently of the preceding situation. Therefore, complete knowledge about the current situation is available if a situation specific symptom is observed.

Component specific symptoms indicate the change to another component of an aspect. Given the preceding situation, the new situation is determined unambiguously.

[3]World Geodetic System 1984, WGS 84 is a widely accepted earth fixed global reference frame for geodetic and navigational purposes

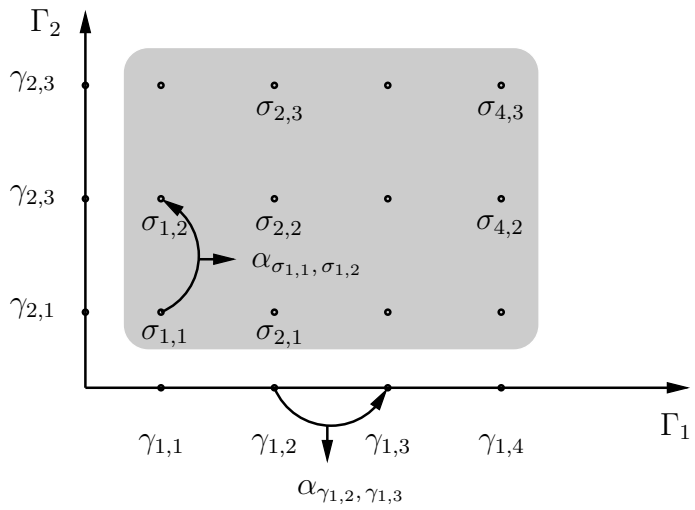


Figure 2.10: *Illustration of situation specific symptoms and component specific symptoms. After the observation of a situation specific symptom $\alpha_{\sigma_{1,1},\sigma_{1,2}}$ the current situation $\sigma_{1,2}$ is known. In contrast, the observation of a component specific symptom $\alpha_{\gamma_{1,2},\gamma_{1,3}}$ only leads to complete knowledge if the preceding situation ($\sigma_{2,1}, \sigma_{2,2}$ or $\sigma_{2,3}$) is known.*

Several practical considerations, such as the dimensionality of the situation space and the separation of independent sensors, suggest the use of component specific symptoms for most applications.

In a more general model the occurrence of symptoms at a transition is itself a stochastic event. None, one or more symptoms may occur at one transition between events. It is also possible that the same symptom is associated with transitions leading to distinct situations. If a transition occurs without an observable symptom or the observed symptom is ambiguous, the assumed current situation is also ambiguous. In some cases the ambiguity may be reduced or resolved by observing symptoms of further transitions that allow to infer the preceding situation. This is an interesting problem for which the Bayesian methods should be well applicable. However, since we are trying to maintain a balance between generality, descriptive simplicity and practical applicability we refrain from complicating our model with the concept of stochastic symptoms. Instead we will assume that each transition between situations is always observable by at least one of the above defined two types of symptoms.

2.1.3.2 Consequences

Our interest in situation awareness is motivated by our intention to employ it for improving the perceived performance services provided to mobile users. For this purpose, we consider the pro-active execution of actions, especially the pro-active transport of information over the network a suitable approach. Depending on the current situation of a user the probabilities of possible actions vary. Usually it will be impossible or too costly to pro-actively perform *all* possible actions. Instead, it

is conjectured to be reasonable to favor the pro-active execution of the more likely actions over the less likely actions. For the application of prefetching this will be analyzed quantitatively and verified in Section 2.2.

We model the decision of the user to take a particular action as a random process, similar to the situation transition process.

Definition 2.12 (Consequence) *An action a user may decide to take or require in a particular situation σ_j is termed consequence $\beta_{\sigma_j,i}$.*

A user in situation σ_j selects one consequence $\beta_{\sigma_j,i}$ out of the set of possible consequences $B_{\sigma_j,i}$. The user's decision is assumed to be the result of a random process. The user's decision process, given a situation σ_j, results in consequence $\beta_{\sigma_j,i}$, with probability $\Pr\{\boldsymbol{\beta} = \beta_{\sigma_j} | \boldsymbol{\sigma} = \sigma_j\}$. For our intended application in prefetching we will later model the user's selection of a particular hyper-media document and the retrieval of the necessary elements over the network as consequence.

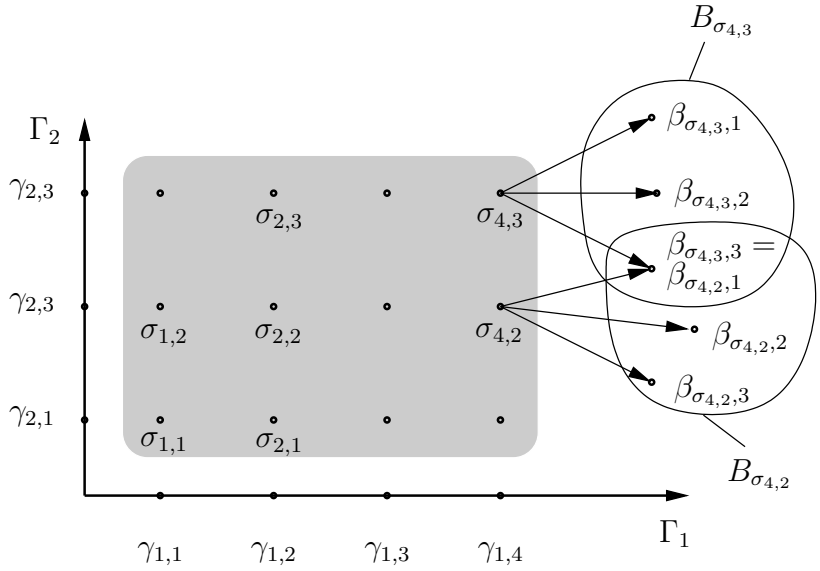


Figure 2.11: *Illustration of consequences. Given that the user is in a particular situation $\sigma_{4,3}$, three consequences $\beta_{\sigma_{4,3},1}, \beta_{\sigma_{4,3},2}$ and $\beta_{\sigma_{4,3},3}$ are possible. Identical consequences $\beta_{\sigma_{4,3},3} = \beta_{\sigma_{4,2},1}$ may be possible in distinct situation, but with distinct probabilities $Pr\{\boldsymbol{\beta} = \beta_{\sigma_{4,3},3} | \boldsymbol{\sigma} = \sigma_{4,3}\} \neq Pr\{\boldsymbol{\beta} = \beta_{\sigma_{4,2},1} | \boldsymbol{\sigma} = \sigma_{4,2}\}$*

.

2.1.4 Selection Criteria for Aspects and Components

The more aspects a situation space is constructed of, the more information can be derived from the knowledge about the actual situation. However, a situation space should not have more aspects than necessary, since the acquisition of sensor data as well as the memory needed for representing the space in a computer grows with the

number of aspect and components. It is therefore necessary to select the aspects and components that contain the information most relevant for the particular purpose.

We will use some well-known metrics from information theory, namely *joint entropy*, *conditional entropy* and *mutual information* to quantify the amount of information we obtain by knowledge of the current situation. For the convenience of the reader, the definitions of these metrics are briefly stated for two random variables. For the straightforward extension to more than two random variables, using chain rules, see [CT91]. We will further use these metrics for comparison of different options in constructing a situation space, i.e. as selection criteria for the aspects and components of a situation space. Again an airport scenario is presented for illustration purposes.

The entropy of a random variable has been defined in eq. 2.8. Extending this definition towards the *joint entropy* of a pair of random variables is straightforward, since the pair $(\boldsymbol{x}, \boldsymbol{y})$ can be interpreted as one single vector-valued random variable:

For two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ with probability mass functions $p_{\boldsymbol{x}}(x_i) = \Pr\{\boldsymbol{x} = x_i\}$ and $p_{\boldsymbol{y}}(y_j) = \Pr\{\boldsymbol{y} = y_j\}$ and *joint probability mass function* $p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j)$, with

$$p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j) = \Pr\{\boldsymbol{x} = x_i, \boldsymbol{y} = y_j\}, \quad i = 1 \ldots M,\ j = 1 \ldots N \tag{2.11}$$

the *joint entropy* of these two random variables is defined:

Definition 2.13 (Joint entropy) *The joint entropy $H(\boldsymbol{x}, \boldsymbol{y})$ of two probability mass functions $p_{\boldsymbol{x}}(x_i)$ and $p_{\boldsymbol{y}}(y_i)$ is defined as*

$$H(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{i=1}^{M}\sum_{j=1}^{N} p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j) \cdot \operatorname{ld} p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j) \quad \text{[bit]} \tag{2.12}$$

If the value of one random variable $\boldsymbol{y}$ is known a priori, the *conditional entropy* quantifies the uncertainty about the remaining unknown random variable $\boldsymbol{x}$.

Definition 2.14 (Conditional entropy) *Given the random variable $\boldsymbol{y}$, the conditional entropy $H(\boldsymbol{x}|\ \boldsymbol{y})$ of another random variable $\boldsymbol{x}$ is defined as*

$$H(\boldsymbol{x}|\ \boldsymbol{y}) = -\sum_{i=1}^{M}\sum_{j=1}^{N} p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j) \cdot \operatorname{ld} p_{\boldsymbol{x}|\ \boldsymbol{y}}(x_i|\ y_j) \quad \text{[bit]} \tag{2.13}$$

The conditional entropy $H(\boldsymbol{x}|\ \boldsymbol{y})$ is always equal or less than the "unconditional" entropy $H(\boldsymbol{x})$. *Mutual information* quantifies this reduction in the uncertainty of one random variable due to the knowledge of the other. Hence, mutual information is a measure of the amount of information that one random variable contains about another random variable.

For two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ with probability mass functions $p_{\boldsymbol{x}}(x_i)$, $p_{\boldsymbol{y}}(y_j)$ and joint probability mass function $p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j)$ with $i = 1 \ldots M, j = 1 \ldots N$, the *mutual information* between these two random variables is defined.

Definition 2.15 (Mutual information) *The mutual information $I(\boldsymbol{x};\boldsymbol{y})$ between two discrete random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined by*

$$I(\boldsymbol{x};\boldsymbol{y}) = \sum_{i=1}^{M}\sum_{j=1}^{N} p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j) \cdot \mathrm{ld}\, \frac{p_{\boldsymbol{x},\boldsymbol{y}}(x_i, y_j)}{p_{\boldsymbol{x}}(x_i) \cdot p_{\boldsymbol{y}}(y_j)} \quad \text{[bit]} \tag{2.14}$$

The following equations link the quantities entropy $H(\boldsymbol{x})$, joint entropy $H(\boldsymbol{x},\boldsymbol{y})$, conditional entropy $H(\boldsymbol{x}|\,\boldsymbol{y})$, and mutual information $I(\boldsymbol{x};\boldsymbol{y})$ and will be used in the following (see [CT91] for proofs):

$$I(\boldsymbol{x};\boldsymbol{y}) = H(\boldsymbol{x}) - H(\boldsymbol{x}|\,\boldsymbol{y}) \tag{2.15}$$

$$I(\boldsymbol{x};\boldsymbol{y}) = H(\boldsymbol{y}) - H(\boldsymbol{y}|\,\boldsymbol{x}) \tag{2.16}$$

$$I(\boldsymbol{x};\boldsymbol{y}) = H(\boldsymbol{x}) + H(\boldsymbol{y}) - H(\boldsymbol{x},\boldsymbol{y}) \tag{2.17}$$

In the following illustrative example we will demonstrate how the metrics are employed for our purposes.

Consider a situation space in a simple airport scenario with two aspects Γ_1 ("gender") and Γ_2 ("flight status")

$$\Gamma_1 = \{\text{male}, \text{female}\},$$

$$\Gamma_2 = \{\text{arriving}, \text{departing}\},$$

resulting in four elementary situations

$$\begin{aligned}
\sigma_1^* &= \{(\text{male}, \text{arriving})\},\\
\sigma_2^* &= \{(\text{male}, \text{departing})\},\\
\sigma_3^* &= \{(\text{female}, \text{arriving})\},\\
\sigma_4^* &= \{(\text{female}, \text{departing})\}.
\end{aligned}$$

In this example we can envision that the symptoms indicating the current situation will be *situation specific* as defined in Section 2.1.3, and derived from the electronic ticket of the passenger. The raw sensor data would be the "Mr./Mrs."-field, the "arrival"- and "departure"-field as well as the current time[4]. Possible consequences that may occur in the situations shall be

$$\begin{aligned}
\beta_1 &= \{\text{"request gate information"}\},\\
\beta_2 &= \{\text{"request shopping information"}\},\\
\beta_3 &= \{\text{"request public transportation information"}\}.
\end{aligned}$$

[4]No location information is necessary for obtaining and using this data, since all necessary information is available in the airline flight databases for every passenger.

Furthermore, we assume that a fictitious trial with airline passengers has been performed and resulted in an (estimated) joint probability mass function $p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i, \beta_i)$ as listed in Table 2.1. Marginalization yields the probability mass functions $p_{\boldsymbol{\sigma}}(\sigma_i) = \Pr\{\boldsymbol{\sigma} = \sigma_i\}$ and $p_{\boldsymbol{\beta}}(\beta_j) = \Pr\{\boldsymbol{\beta} = \beta_j\}$.

$p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i, \beta_i)$	σ_1^*	σ_2^*	σ_3^*	σ_4^*	$p_{\boldsymbol{\beta}}(\beta_i)$
β_1	2/100	21/100	2/100	19/100	44/100
β_2	6/100	3/100	4/100	3/100	16/100
β_3	19/100	3/100	17/100	1/100	40/100
$p_{\boldsymbol{\sigma}}(\sigma_i)$	27/100	27/100	23/100	23/100	$\sum = 1$

Table 2.1: *Joint probability mass function $p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i, \beta_j)$, with $p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i, \beta_j) = Pr\{\boldsymbol{\sigma} = \sigma_i, \boldsymbol{\beta} = \beta_j\}$ and marginalized probability mass functions $p_{\boldsymbol{\sigma}}(\sigma_i) = Pr\{\boldsymbol{\sigma} = \sigma_i\}$ and $p_{\boldsymbol{\beta}}(\beta_j) = Pr\{\boldsymbol{\beta} = \beta_j\}$*

All relevant metrics that have been computed for this case are listed in Table 2.2 and discussed in the following.

	Γ_1, Γ_2	Γ_1	Γ_2	
$H(\boldsymbol{\sigma}, \boldsymbol{\beta})$	2.9946	2.4676	2.066	[bit]
$H(\boldsymbol{\sigma})$	1.9954	0.9953	1.0000	[bit]
$H(\boldsymbol{\beta})$	1.4729	1.4729	1.4729	[bit]
$H(\boldsymbol{\beta}\mid \boldsymbol{\sigma})$	0.9992	1.4722	1.0066	[bit]
$H(\boldsymbol{\sigma}\mid \boldsymbol{\beta})$	1.5217	0.9946	0.5337	[bit]
$I(\boldsymbol{\sigma}; \boldsymbol{\beta})$	0.4737	0.0007	0.4663	[bit]

Table 2.2: *Joint entropy $H(\boldsymbol{\sigma}, \boldsymbol{\beta})$, entropies $H(\boldsymbol{\sigma})$, $H(\boldsymbol{\beta})$, conditional entropies $H(\boldsymbol{\sigma}\mid \boldsymbol{\beta})$, $H(\boldsymbol{\beta}\mid \boldsymbol{\sigma})$ and mutual information $I(\boldsymbol{\sigma}; \boldsymbol{\beta})$ for full situation space (Γ_1, Γ_2) and reduced situation spaces (Γ_1) and (Γ_2).*

The first metric we are interpreting is the joint entropy of the situation transitions and the consequences $H(\boldsymbol{\sigma}, \boldsymbol{\beta})$ which is 2.9946 bit for the original situation space consisting of Γ_1 and Γ_2. This metric quantifies the overall uncertainty we have about the situation a participating user is in and which consequence will result from this situation *in the absence* of any sensor data.

The "unconditional" entropies $H(\boldsymbol{\sigma}) = 1.995\,\text{bit}$ and $H(\boldsymbol{\beta}) = 1.4729\,\text{bit}$ quantify our uncertainty if we are only interested in the situation or the consequences in

the absence of sensor data. Since it is our intention to apply our concept to proactive retrieval of information that will be requested by the user, we would model the requests for particular documents as consequences. In this case the uncertainty about the user's next request would be $H(\boldsymbol{\beta}) = 1.4729$ bit.

This uncertainty is reduced to $H(\boldsymbol{\beta}|\ \boldsymbol{\sigma}) = 0.9992$ bit, in a system which knows and uses the current situation. This increase in certainty or knowledge about the next request is equivalent with the mutual information $H(\boldsymbol{\beta}) - H(\boldsymbol{\beta}|\ \boldsymbol{\sigma}) = I(\boldsymbol{\sigma}; \boldsymbol{\beta}) = 0.4737$. Fig. 2.12 illustrates this effect. The mutual information also quantifies the amount of

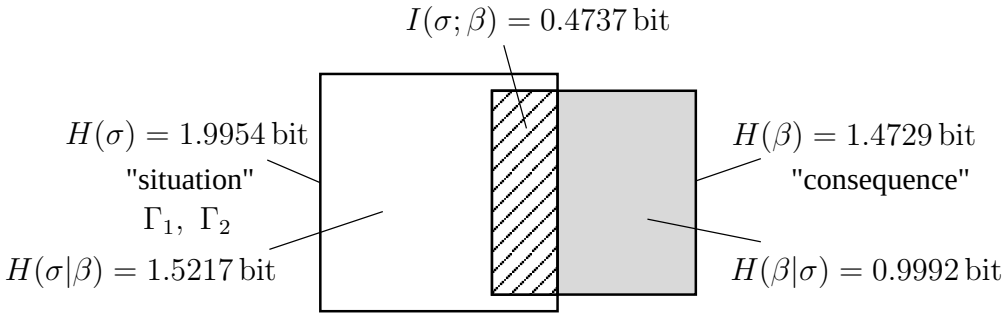


Figure 2.12: *For the full situation space, consisting of* Γ_1 *and* Γ_2*, the uncertainty about the next consequence* β *is reduced from* $H(\boldsymbol{\beta}) = 1.4729$ *bit to* $H(\boldsymbol{\beta}|\ \boldsymbol{\sigma}) = 0.9992$ *bit, by applying the knowledge about the current situation* σ*. This increase in certainty is equivalent with the mutual information* $I(\boldsymbol{\sigma}; \boldsymbol{\beta}) = 0.4737$ *bit.*

information about the current situation contained in the observation of the requests for information. By observing the consequences we can reduce the uncertainty about the situation to $H(\boldsymbol{\sigma}|\ \boldsymbol{\beta}) = 1.5217$, compared to $H(\boldsymbol{\sigma}) = 1.9954$, which, as stated before, would be the uncertainty if no sensor data was available or used. The increase in certainty is again the mutual information.

After the fictitious trial phase, it shall now be determined whether sensor data for all aspects is necessary, and what degradations are caused if aspects are neglected. We will demonstrate that the previously applied metrics are also applicable in selecting the components and aspects for a situation space.

The first suggestion for a reduction of the situation space shall be to neglect whether the person is arriving or departing. Table 2.3 lists the joint probability mass function for the configuration in which only the gender is considered. The metrics for the evaluation of this suggestion are found in the third column ("Γ_1") of Table 2.2. Since for this case flight status is not of interest, the situation is only determined by the gender. Hence, the uncertainty regarding the situation $H(\boldsymbol{\sigma})$ is reduced to 0.9953 bit compared to the full situation space. However, this reduction is not caused by increase in knowledge but by limiting the scope of interest or "ignorance".

If we use the knowledge of the current situation, in this configuration the user's gender, the uncertainty regarding the consequences is only marginally reduced by

$p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i,\beta_i)$	$\sigma_1^* \cup \sigma_2^*$	$\sigma_3^* \cup \sigma_4^*$	$p_{\boldsymbol{\beta}}(\beta_i)$
β_1	23/100	21/100	44/100
β_2	9/100	7/100	16/100
β_3	22/100	18/100	40/100
$p_{\boldsymbol{\sigma}}(\sigma_i)$	54/100	46/100	$\sum = 1$

Table 2.3: *Joint probability mass functions $p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i,\beta_j)$ for incomplete situation information, only aspect Γ_1 ("gender") known*

$I(\boldsymbol{\sigma};\boldsymbol{\beta}) = 0.0007$ bit, to $H(\boldsymbol{\beta}|\ \boldsymbol{\sigma}) = 1.4722$ bit. Obviously, the behavior of male and female users is very similar according to this metric. We see clearly, that the gender of the subject contains only a small amount of information about the likely requests. Fig. 2.13 illustrates this effect.

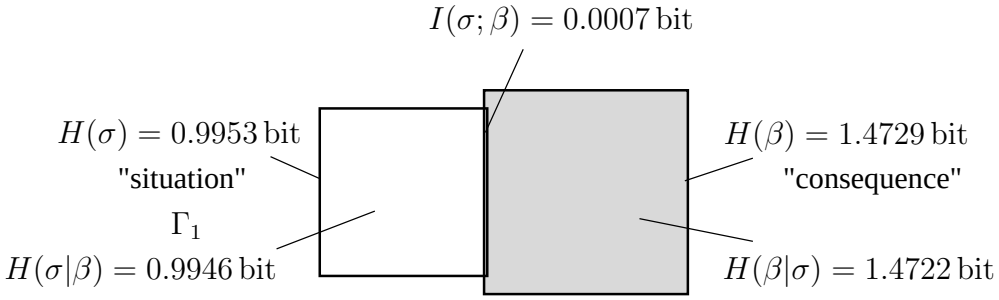


Figure 2.13: *For the first configuration of a reduced situation space, consisting only of Γ_1, i.e. only considering the "gender" aspect, the uncertainty about the next consequence β is marginally reduced from $H(\boldsymbol{\beta}) = 1.4729$ bit to $H(\boldsymbol{\beta}|\ \boldsymbol{\sigma}) = 1.4722$ bit, by applying the knowledge about the current situation σ. This increase in certainty is equivalent with the (almost negligible) mutual information $I(\boldsymbol{\sigma};\boldsymbol{\beta}) = 0.0007$ bit. Hence, this configuration is not preferable.*

The suggestion to take only the aspect "gender" into account does not promise significant benefits over using no situation information at all.

The second proposed configuration shall be to neglect the gender completely and to use only sensor data on the aspect "flight status".
The resulting joint probability mass function for this case is listed in Table 2.4. Inspection of the metrics in the fourth column of Table 2.2, immediately shows that almost the same reduction in uncertainty is achieved with this suggestion as in the trial configuration with the complete situation space. The mutual information

$p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i,\beta_i)$	$\sigma_1^* \cup \sigma_3^*$	$\sigma_2^* \cup \sigma_4^*$	$p_{\boldsymbol{\beta}}(\beta_i)$
β_1	4/100	40/100	44/100
β_2	10/100	6/100	16/100
β_3	36/100	4/100	40/100
$p_{\boldsymbol{\sigma}}(\sigma_i)$	50/100	50/100	$\sum = 1$

Table 2.4: *Joint probability mass functions $p_{\boldsymbol{\sigma},\boldsymbol{\beta}}(\sigma_i,\beta_j)$ for incomplete situation information, only aspect Γ_2 ("flight status") known*

$I(\boldsymbol{\sigma};\boldsymbol{\beta}) = 0.4663$ bit is close to the mutual information of 0.4737 bit for the complete information space. Hence the uncertainty about the consequences is reduced to $H(\boldsymbol{\beta}|\,\boldsymbol{\sigma}) = 1.0066\,\text{bit}$ as depicted in Fig. 2.14. If the situation space has to be reduced in this example it is obviously recommendable to drop the "gender" aspect in favor of the "flight status" aspect.

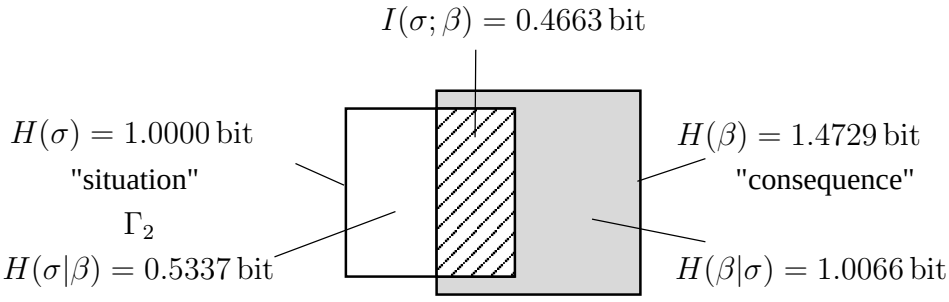


Figure 2.14: *For the second configuration of a reduced situation space, consisting only of Γ_2, i.e. only considering the "flight status" aspect, the uncertainty about the next consequence β is significantly reduced from $H(\boldsymbol{\beta}) = 1.4729$ bit to $H(\boldsymbol{\beta}|\,\boldsymbol{\sigma}) = 1.0066$ bit, by applying the knowledge about the current situation σ. The increase in certainty is equivalent with the mutual information $I(\boldsymbol{\sigma};\boldsymbol{\beta}) = 0.4663$ bit. This configuration achieves almost the same performance as the full situation space configuration. Therefore, this configuration would be recommendable.*

Apart from their application in the design phase of a situation space, the previously discussed metrics provide valuable monitoring information for a situation aware system, since they are continuously computable during the operation of a system.

2.1.4.1 Interpretation for Ignorance of Situation as False Probability Assumption

We want to conclude our discussion of the probabilities for situations and consequences from an information theoretic viewpoint by giving an interpretation of a decision that does *not* make use of all available situation or context information.

If current situation and succeeding situations or current situation and consequence are not independent, not considering the information about the current situation inherently leads to an assumed probability mass function $q(\cdot)$ for the situation transitions or consequences that differs from the true probability mass function $p(\cdot)$ in the particular situation.

Whenever assumed probabilities differ from the true probabilities a degradation of performance of a following decision process occurs. This phenomenon is well-known e.g. from source coding. If, instead of the true probability mass function $p_{\boldsymbol{x}}(x_i)$ a different probability mass function $q_{\boldsymbol{x}}(x_i)$ is wrongly assumed, we have to invest not only the bits for the entropy $H(\boldsymbol{x})$, but an extra penalty. The relative entropy or Kullback-Leibler "distance" quantifies the amount of error caused by a wrong probability assumption[5].

Definition 2.16 (Relative entropy or Kullback-Leibler distance) *The relative entropy or Kullback-Leibler distance between two probability mass functions $p_{\boldsymbol{x}}(x_i)$ and $q_{\boldsymbol{x}}(x_i)$ is defined as*

$$D(p_{\boldsymbol{x}}(x_i) \parallel q_{\boldsymbol{x}}(x_i)) = \sum_{i=1}^{N} p_{\boldsymbol{x}}(x_i) \cdot \mathrm{ld}\frac{p_{\boldsymbol{x}}(x_i)}{q_{\boldsymbol{x}}(x_i)} \tag{2.18}$$

In source coding the total amount of bits needed is then $H(\boldsymbol{x}) + D(p_{\boldsymbol{x}}(x_i) \parallel q_{\boldsymbol{x}}(x_i))$ instead of $H(X)$. The relative entropy does not have a similarly direct equivalence in our field of application, however, it is well suited for comparing different options of a situation space design.

Table 2.5 lists the relative entropies $D(p_{\boldsymbol{\beta}}(\beta_i|\ \sigma_i) \parallel\ q_{\boldsymbol{\beta}}(\beta_i|\ \cdot\ ))$ that will occur for all four situations and all situation space configurations of our situation example.

If the complete situation information is taken into account (Γ_1, Γ_2 known), no error occurs and the relative entropy is zero (second column).

If only the gender (Γ_1) is taken into account (third column), the relative entropy is significantly higher compared to the case in which the flight status (Γ_2) is considered (fourth column). The error is maximized if the system is "situation-blind", i.e. no situation information is used (fifth column).

We have laid out the structure and properties of our concept of situation and situation space and have discussed the role and importance of probabilities in this concept. In the following section we will discuss the process of acquiring sensible numerical values for these probabilities during the operation of a system.

[5] The convention that $0 \cdot \mathrm{ld}\frac{0}{q} = 0$ and $p \cdot \mathrm{ld}\frac{p}{0} = \infty$ is used.

$D(p_{\boldsymbol{\beta}}(\beta_i \mid \sigma_i) \parallel q_{\boldsymbol{\beta}}(\beta_i \mid \cdot))$	Γ_1, Γ_2	Γ_1	Γ_2	blind	
σ_1^*	0	0.4602	0.0023	0.4884	[bit]
σ_2^*	0	0.4024	0.0087	0.3754	[bit]
σ_3^*	0	0.5036	0.0033	0.4722	[bit]
σ_4^*	0	0.5400	0.0157	0.5731	[bit]

Table 2.5: *Relative entropy $D(p_{\boldsymbol{\beta}}(\beta_i \mid \sigma_i) \| q_{\boldsymbol{\beta}}(\beta_i \mid \cdot))$ quantifying for each situation the error in assumed probability mass functions for various levels of situation knowledge. If the complete situation information is taken into account (Γ_1, Γ_2 known), no error occurs and the relative entropy is zero ("Γ_1, Γ_2"). If only the aspect "gender" (Γ_1) is taken into account ("Γ_1"), the relative entropy is significantly higher compared to the case in which the aspect "flight status" (Γ_2) is considered ("Γ_2"). The error is maximized if the system is "situation-blind", i.e. no situation information is used ("blind").*

2.1.5 Estimation of Model Probabilities

In the previous section we have developed a dynamic model for the transitions between situations and the event of consequences. We have defined random processes based on the probabilities for the transitions and consequences. So far, we have been assuming perfect a priori knowledge of the numerical values of these probabilities. We might assume that the probabilities are determined by domain experts based on previous experience, physical conditions or logical reasoning. In many cases, though, this assumption would be not realistic, since either no previous experience is available, or manual acquisition of the probabilities would be tedious due to the large number of possible transitions and consequences. Therefore, in most applications a procedure is necessary which enables to learn and adapt to the actual probabilities during operation.

In the following, we will show that parameter estimation is well-applicable for this task. The nature and performance of different estimators for the probabilities is analyzed first from a classical perspective, followed by a Bayesian perspective.

The problems of estimating the transition probabilities from one situation to a possible succeeding situation, and estimating the probabilities of the possible consequences, given the current situation, are identical. For each situation, two tables are maintained. A first table contains all possible succeeding situations and their frequency of occurrence; a second table contains all possible consequences.

The number of occurrences x_i of the situations σ_i, $i = 1 \dots N$ are represented by a vector $\mathbf{x}$, where N is the number of elements in the table and n is the total number of observations made.

$$\mathbf{x} = (x_1, \dots, x_N)^T, \tag{2.19}$$

$$\text{where} \quad x_i \in \{0, 1, 2, \dots, n\} \tag{2.20}$$

$$\text{and} \quad \sum_{i=1}^{N} x_i = n \tag{2.21}$$

It is our objective to estimate the values of the N probabilities p_i from the observed frequencies of occurrence x_i. Later, these estimated values will be used as a basis for the subsequent decision processes.

We have to be aware that the problem at hand is in fact only a $N-1$ – dimensional estimation problem, since, of course

$$\sum_{i=1}^{N} p_i = 1. \tag{2.22}$$

For easier reference, we will adopt the standard notation used in estimation literature. The parameter to be estimated is denoted Θ, if it is scalar, and $\mathbf{\Theta}$ if it is vector-valued.

$$\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_N)^T = (p_1, \ldots, p_N)^T, \tag{2.23}$$

$$\text{where} \quad 0 \leq \Theta_i \leq 1 \tag{2.24}$$

$$\text{and} \quad \sum_{i=1}^{N} \Theta_i = 1 \tag{2.25}$$

Fig. 2.15 illustrates these boundary conditions (eqs. 2.24, 2.25). Due to eq. 2.24 the parameter $\boldsymbol{\Theta}$ is restricted to the plane passing through $(1,0,0)^T$, $(0,1,0)^T$ and $(0,0,1)^T$, due to eq. 2.25 it is restricted to the triangle $\mathcal{H}$ on this plane.

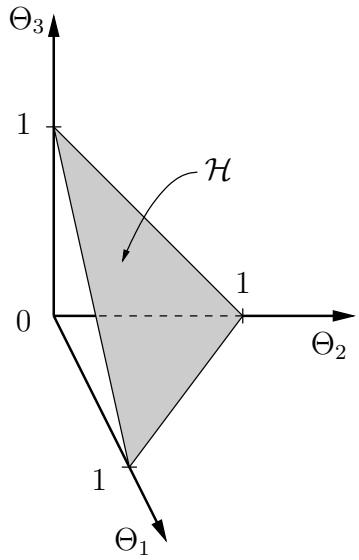


Figure 2.15: *Illustration of parameter space for $N = 3$. The parameter $\boldsymbol{\Theta}$ is restricted to $\mathcal{H}$, since $\sum_{i=1}^{N} p_i = 1$.*

From the very beginning of a situation aware system's operation, the choices made by the users are observed. The successive observations are logged into the appropriate tables. Fig. 2.16 illustrates how the resulting changes of the observed frequencies of occurrence x_i, of a particular situation's tables, create a path through the observation space.

The vector $\boldsymbol{\Theta}$ is a random variable, distributed according to the multinomial distribution.

$$f(\mathbf{x}|n, \boldsymbol{\Theta}) = \frac{n!}{\prod_{i=1}^{N} x_i!} \prod_{i=1}^{N} \Theta_i^{x_i}. \tag{2.26}$$

A number of classical estimators are known for this type of problem. We will first introduce them by their equations and will later take a Bayesian perspective to point out the implicit assumptions in these classical estimators.

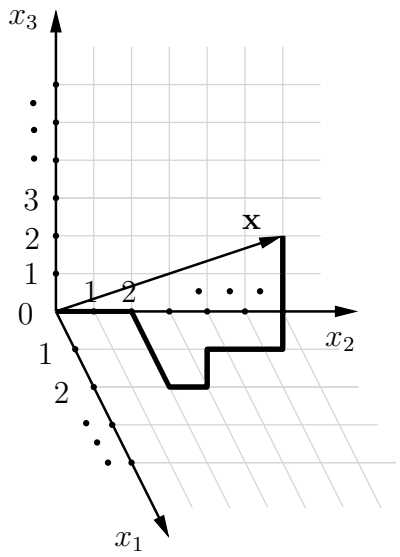


Figure 2.16: *Illustration of observation space for $N = 3$. The successive observations* $\mathbf{x}$ *create a path through the observation space*

2.1.5.1 Frequentist's Estimator

The most straightforward estimator is based on the frequentist's definition of probability. It uses the vector of relative frequencies of the observed transitions as an estimate:

$$\hat{\boldsymbol{\Theta}} = \left(\frac{x_1}{n}, \frac{x_2}{n}, \ldots, \frac{x_{N-1}}{n}, \frac{x_N}{n}\right)^T \tag{2.27}$$

As stated previously, we are especially interested in a system that is capable of providing information from the very beginning of its operation. This particularly requires the generation of estimates for small values of n. Unfortunately, the frequentist's definition of probability specifically demands large values of n. With only few observations, some recorded relative frequencies x_i/n will still be zero, thus leading to an assigned probability p_i of zero. Depending on the application this proposition may be far too drastic, since it informs all subsequent deciding instances, that the particular event will *never* occur. Similarly drastic is the assignment of $p_i = 1$ if $x_i = n$, which is, of course, always the case for the first observed event.

Before any observation is made ($n = 0$), eq. 2.27 is not defined, since the expressions for the vector's elements become $0/0$.

Assigning a probability without any observation would be justified by what Bernoulli called the "principle of insufficient reason". Keynes later used the term "principle of indifference" for the same idea: assuming that for a set of mutually exclusive events, for which we have no reason to believe that one of these events is more likely to occur than another, we should assign the same probability to all of them. Applying this

principle leads to the following equation for the estimation before any observation has been made:

$$\hat{\mathbf{\Theta}} = \left(\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}, \frac{1}{N}\right)^T \quad \text{for } n = 0 \tag{2.28}$$

While eq. 2.28 closes the gap for $n = 0$, the problem of the drastic propositions remains. Fortunately, Laplace's law of succession provides another estimator for the multinomial problem that treats these cases elegantly without any gaps.

2.1.5.2 Laplace's Law of Succession

Laplace's law of succession was initially stated for the binomial distribution which is the special case of the multinomial distribution with $N = 2$. In the typical example the events are "success" and "failure", with x_1 =number of "successes" and p_1 the respective probability.

$$f(x_1|n, \Theta) = \binom{n}{x_1} \cdot \Theta^{x_1} \cdot (1 - \Theta)^{n-x_1}, \quad \Theta = p_1. \tag{2.29}$$

For this binomial case the law of succession states that after the observation of n events, the probability that the next transition (event) will be "success" is:

$$\hat{\Theta} = \frac{x_1 + 1}{n + 2} \quad \text{(Laplace's law of succession)} \tag{2.30}$$

Applying eq. 2.30 for the case without any previous observations, i.e. $x_1 = 0, n = 0$, we see that $p_1 = p_2 = 1/2$, thus satisfying the previously stated principle of indifference.

The natural generalization of Laplace's law of succession for the multinomial distribution is

$$\hat{\mathbf{\Theta}} = (\frac{x_1 + 1}{n + N}, \frac{x_2 + 1}{n + N}, \ldots, \frac{x_{N-1} + 1}{n + N}, \frac{x_N + 1}{n + N}) \tag{2.31}$$

(Generalized Laplace's law of succession)

2.1.5.3 Derivation of Maximum Likelihood Estimator for Multinomial Distributions

We will now employ the maximum-likelihood principle to derive an estimator which is *optimal* in a certain sense.

Maximum likelihood (ML) estimation is known to have certain desirable properties. If an ML estimator exists, it attains the Cramer-Rao Lower Bound (CRLB), and does so faster than any other estimator [Kay93]. The general principle of maximum likelihood estimation is to use the value Θ as an estimate, that maximizes a so-called

likelihood function $\mathrm{L}(x;\Theta)$. The likelihood function is derived from the probability density function by using the parameter instead of the random variable as variable. The estimate $\hat{\Theta}_{\mathrm{ML}}$ is then

$$\hat{\Theta}_{\mathrm{ML}} = \arg\max_{\Theta} \mathrm{L}(x;\Theta). \tag{2.32}$$

Applied to the multinomial distribution (eq. 2.26) this results in

$$\hat{\boldsymbol{\Theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\Theta}} \frac{n!}{\prod_{i=1}^{N} x_i!} \prod_{i=1}^{N} \Theta_i^{x_i}. \tag{2.33}$$

Since we have to find only the position of the maximum, i.e. the mode of $\mathrm{L}(x;\Theta)$, the normalization term in eq. 2.33 is irrelevant for finding the maximum and we only have to consider

$$\hat{\boldsymbol{\Theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\Theta}} \prod_{i=1}^{N} \Theta_i^{x_i}. \tag{2.34}$$

Using the fact that the log-function is strictly monotone we transform eq. 2.34 into a sum.

$$\hat{\boldsymbol{\Theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\Theta}} \log \prod_{i=1}^{N} \Theta_i^{x_i} = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{N} x_i \cdot \log \Theta_i \tag{2.35}$$

Now the mode should be easily determined by setting the first partial derivative with respect to $\boldsymbol{\Theta}$ to zero.

$$\frac{\partial \sum_{i=1}^{N} x_i \cdot \log \Theta_i}{\partial \boldsymbol{\Theta}} \to 0, \tag{2.36}$$

which corresponds to setting the first partial derivatives with respect to $\boldsymbol{\Theta}$'s elements to zero.

$$\frac{\partial \sum_{i=1}^{N} x_i \cdot \log \Theta_i}{\partial \Theta_i} \to 0, \tag{2.37}$$

Since the multinomial distribution as defined in eq. 2.26 is only $N-1$ dimensional, we obtain

$$\Theta_N = 1 - \sum_{i=1}^{N-1} \Theta_i. \tag{2.38}$$

We use eq. 2.37 and 2.38 to obtain

$$x_i \cdot \frac{1}{\Theta_i} + x_N \cdot \frac{1}{\Theta_N} \cdot (-1) = 0 \quad \forall\, i \in \{1, \ldots, N-1, N\} \tag{2.39}$$

from which follows

$$\frac{\Theta_j}{\Theta_i} = \frac{x_j}{x_i} \Rightarrow \frac{\Theta_h}{\Theta_i} + \frac{\Theta_j}{\Theta_i} = \frac{x_j}{x_i} + \frac{x_h}{x_i} \quad \forall\, h, i, j \in \{1, \ldots, N-1, N\}. \tag{2.40}$$

If we continue adding to both sides ratios with Θ_i and x_i in the denominator, we end up with

$$\frac{\sum_{j=1}^{N} \Theta_j}{\Theta_i} = \frac{\sum_{j=1}^{N} x_j}{x_i} \Rightarrow \frac{1}{\Theta_i} = \frac{n}{x_i} \tag{2.41}$$

and finally

$$\hat{\boldsymbol{\Theta}}_{\mathrm{ML}} = \left(\frac{x_1}{n}, \frac{x_2}{n}, \ldots, \frac{x_{N-1}}{n}, \frac{x_N}{n}\right)^T \tag{2.42}$$

which is exactly the frequentist's estimate as defined in eq. 2.27. Hence, we have shown that the frequentist's estimator is optimum in the maximum likelihood sense. We will now compare the performance of the frequentist's (or maximum likelihood) estimator with Laplace's law of succession.

2.1.5.4 Analysis and Comparison of Estimator Errors

The previous derivation of the maximum likelihood estimator (MLE) has shown that the frequentist's estimator is optimal in the maximum likelihood sense. Consequently Laplace's law of succession is suboptimal in the maximum likelihood sense. However, we wish to better understand the nature of the errors that both estimators result in.

The error ϵ is defined as the difference between the true value of the parameter Θ and the estimated value $\hat{\Theta}$.

$$\epsilon = \Theta - \hat{\Theta} \tag{2.43}$$

Analogously, the error for the vector case is defined as

$$\boldsymbol{\epsilon} = \boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}. \tag{2.44}$$

A simulation experiment consisting of a binomial random source that generates two different possible events (symbols) with probabilities p_1 and $p_2 = 1 - p_1$ is performed. Both estimators use the resulting sequence of events to compute their estimates about the symbol probabilities. Supposedly the estimates will become more accurate with increasing number of observations.
Fig. 2.17 shows the *average* error $\bar{\epsilon}$ for 10000 repetitions. The results for different probabilities $p_1 \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and increasing numbers of observed events n are plotted.

Clearly visible is the outstanding property of *unbiasedness* of the maximum likelihood estimator. For all values of p_1 the average error $\bar{\epsilon}$ is zero. In contrast, Laplace's law of succession results in a bias for all values of p_1, with the exception $p_1 = 0.5$. The "skeptic" nature of this estimator is nicely observable. It tends towards the balanced estimate of $p_1 = p_2$ and has to be "convinced" by multiple observations to assume unbalanced probabilities such as $p_1 = 0.0$ or $p_1 = 1.0$.

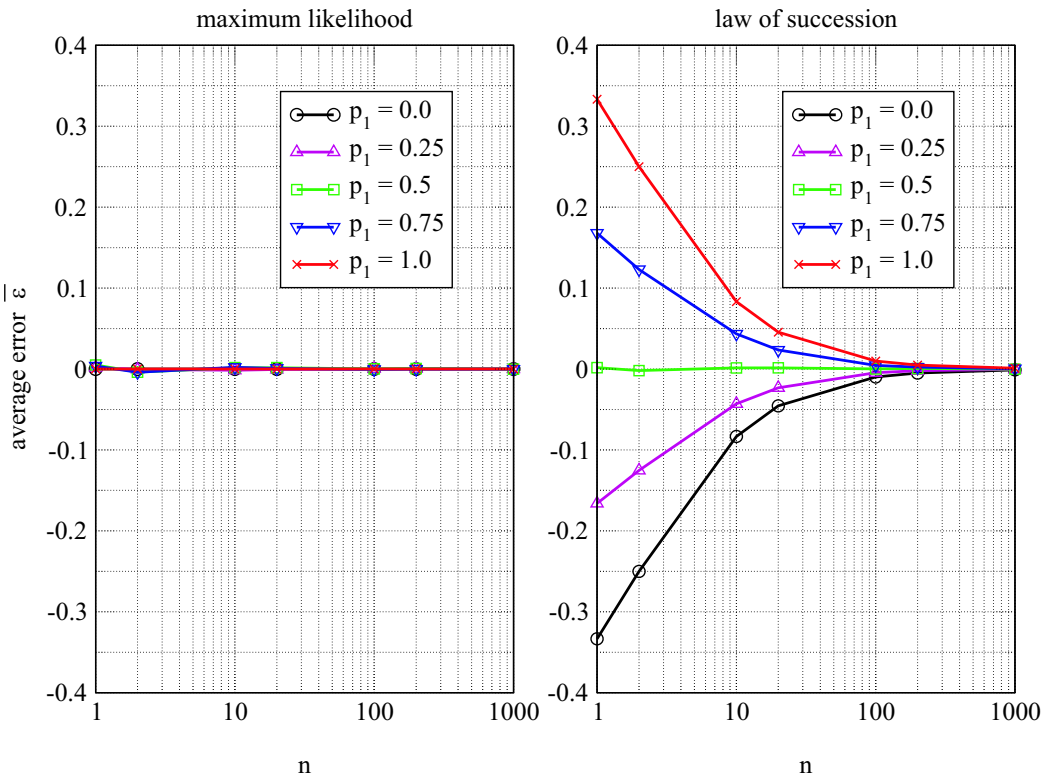


Figure 2.17: *Comparison of average error $\bar{\epsilon}$ caused by the maximum likelihood estimator (left) and Laplace's law of succession (right) for probabilities $p_1 \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and increasing numbers of observed events n. Each data point is averaged over 10000 repetitions. The estimates of the MLE show zero bias for all probabilities. The estimates of the law of succession tend towards the balanced estimate of $p_1 = p_2$*

A superficial look on Fig. 2.17 might lead to the conclusion that the MLE results in smaller errors than Laplace's law of succession. However, this conclusion is potentially erroneous. The fact that the MLE is unbiased, does not necessarily imply that its errors are small. The unbiasedness only assures that the positive and negative aberration compensate each other. Fig. 2.18 reveals this effect:

We see that for moderate probabilities $p_1 \in \{0.25, 0.5, 0.75\}$, the average *absolute* error $\overline{|\epsilon|}$ of the MLE is especially in the early phases ($n = 1 \ldots 10$) significantly higher than the error caused by the law of succession.

The previously mentioned "drastic" nature of the MLE explains the large error for $p_1 \in \{0.25, 0.5, 0.75\}$ at $n = 1$. Depending on the first observation, the MLE takes the supposition $p_1 = 0.0$ or $p_1 = 1.0$, hence $|\epsilon| = |0.5 - 0.0|$ or $|0.5 - 1.0|$ and therefore the average error $\overline{|\epsilon|}$ is 0.5. For the extreme values $p_1 = 0.0$ and $p_1 = 1.0$ the MLE's drastic supposition results of course in $\overline{|\epsilon|} = 0.0$.

Depending on the subsequent decision process it may be appropriate to assume that the negative influence of an erroneous estimate increases super-proportionally

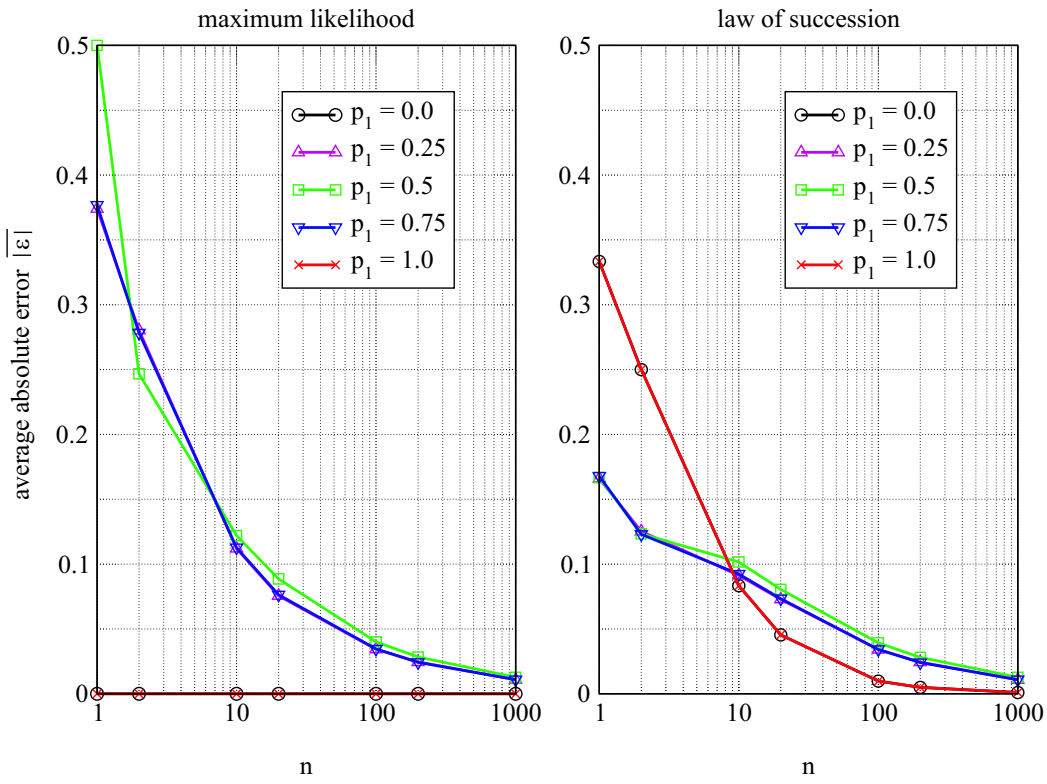


Figure 2.18: *Comparison of average absolute error $\overline{|\epsilon|}$ caused by the maximum likelihood estimator (left) and Laplace's law of succession (right) for probabilities $p_1 \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and increasing numbers of observed events n. Each data point is averaged over 10000 repetitions. For moderate probabilities $p_1 \in \{0.25, 0.5, 0.75\}$ the average absolute error $\overline{|\epsilon|}$ of the MLE is especially in the early phases ($n = 1 \ldots 10$) significantly higher than the error caused by the law of succession*

with the absolute error $|\epsilon|$. The most common metric for quantifying the performance of an estimator is the average squared error $\overline{\epsilon^2}$. Fig. 2.19 shows the resulting curves for the MLE and the law of succession:

We see that also for this metric the law of succession performs better for moderate probabilities than the MLE. The comparison has demonstrated that for some applications the law of succession may achieve even better results than the maximum likelihood estimator. How is this possible, when the MLE is supposed to be the *optimum* estimator? We have to remember the fact, that the MLE is the optimum *unbiased* estimator. Better estimators exist if unbiasedness is not required, which is the case for many applications.

Both estimators converge towards the same estimates for large numbers of observations n, but result in considerable difference for small n. Furthermore, we have observed the effect that both estimators perform differently, depending on the true value of the parameter Θ (here p_1).

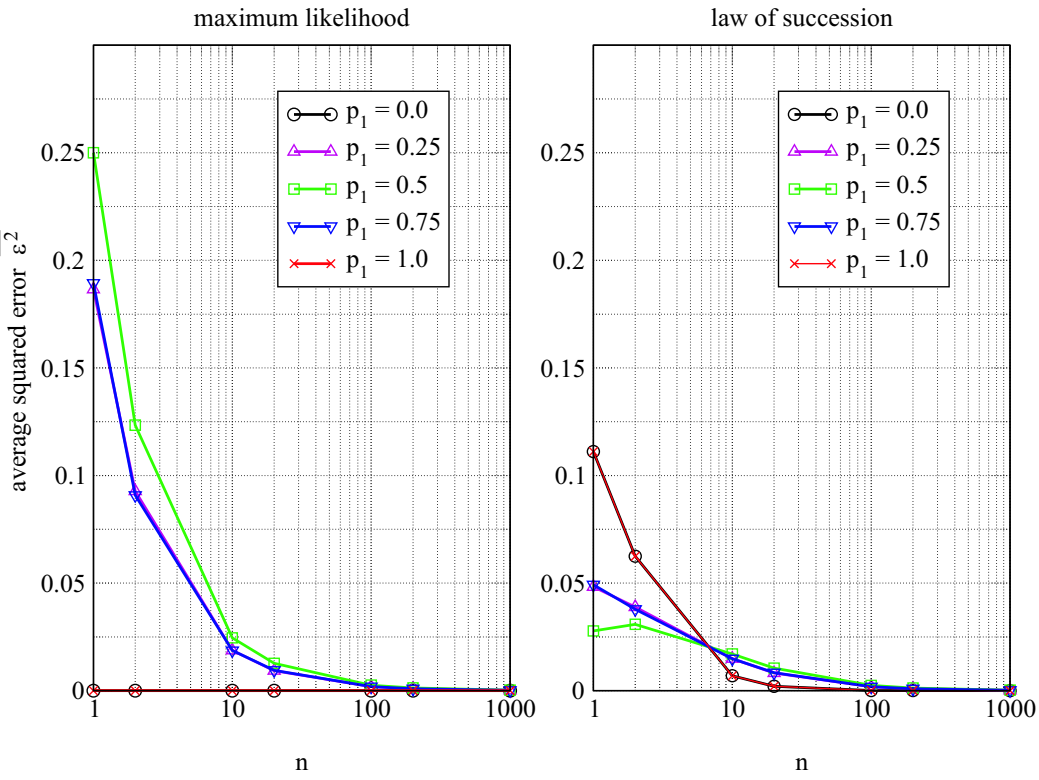


Figure 2.19: *Comparison of average squared error $\overline{\epsilon^2}$ caused by the maximum likelihood estimator (left) and Laplace's law of succession (right) for probabilities $p_1 \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and increasing numbers of observed events n. Each data point is averaged over 10000 repetitions. For moderate probabilities $p_1 \in \{0.25, 0.5, 0.75\}$ the average squared error $\overline{\epsilon^2}$ of the MLE is especially in the early phases ($n = 1 \ldots 10$) significantly higher than the error caused by the law of succession*

Depending on the application it may become necessary to obey a *threshold* for the estimated probabilities. In an example pertaining to prefetching this threshold may specify the minimum probability of occurrence for a document in order to invest the necessary network resources for retrieval of the document. Especially for the case where only a few observations are available, the use of the ML estimate is problematic. If only a single event has been observed the MLE immediately assigns a probability of one to this particular event, which exceeds any minimum probability threshold. Therefore, the "over-confident" estimate of the MLE is not well-suited for this purpose. Instead, the estimate provided by Laplace's law of succession for the case of a uniform a priori probability density function, yields a suitable probability to compare against the threshold.

If we accept the fact that the true parameter Θ is itself a random variable and distributed according to a probability density function $f_\Theta(\Theta)$, we can now adopt a Bayesian perspective to better understand the fundamentals of the estimation process.

2.1.5.5 Bayesian Analysis of Estimation for Multinomial Distributions

The maximum likelihood estimator and Laplace's law of succession both generate *point estimates*, i.e. single values $\hat{\Theta}$ or vectors $\hat{\mathbf{\Theta}}$. These point estimates do not carry any information regarding the estimators' confidence in the estimate. We have already pointed out the problems that the maximum likelihood estimator's tendency to claim sometimes drastic suppositions , e.g. telling an application that a transition from one situation to another can *never* occur, because it hasn't been observed previously. Desirable would be an estimator that conveys all its information, including its uncertainty, towards all subsequent decision instances. Fortunately, Bayes' formula facilitates the computation of the a posteriori probability density function $f_{\Theta|x}(\Theta|x)$ for the parameter Θ, given the observed data x [Kay93, vT68, Siv96] . Therefore, we will briefly introduce and then adopt the Bayesian perspective in order to gain further insights into the problem.

The success of probability theory about 300 years ago has its main reason in the theories applicability to games of chance. For this application *deductive* logic and combinatorics were able to solve most problems, usually with the purpose of determining the odds for events, such as drawing a specific combination of playing cards or shooting a particular constellation of dice. In many fields of more scientific interest a somewhat inverted problem has to be solved, i.e. to *infer* on the properties of some process, given observations or measured data. Reverend Thomas Bayes was the first to ponder this problem; followed by Laplace to whom the present-day form of the Bayes-theorem is accredited and who successfully applied it to many physical and medical problems. Sivia [Siv96] states that "to the pioneers such as the Bernoullis, Bayes and Laplace, probability represented a *degree-of-belief* or plausibility: how much they thought that something was true, based on the evidence at hand. To the nineteenth-century scholars, however, this seemed too vague and subjective an idea to be the basis of a rigorous mathematical theory." The ideas of Bayes and Laplace were not rediscovered before the last century, when Jeffreys and especially Jaynes [Jay89] refined them and demonstrated their applicability to a multitude of modern problems. Strongly influenced by them is Good [Goo65], who specifically applies Bayesian methods to problems pertaining to the estimation of probabilities. Both classical and Bayesian estimation techniques and their application to problems in communications and statistical signal processing are comprehensively presented by van Trees [vT68] and in the more recent book by Kay [Kay93], while Gelman et al [GCSR65] and Carlin and Louis [CL00] apply Bayesian methods to general data analysis. It should not be left unmentioned that the various schools of probability theory and statistics still lead controversial discussions. None of the cited authors misses to point out the deficiencies of a purely frequentistic perspective. For a highly interesting discussion and systematic account of the various interpretations and schools of probability the inclined reader is referred to [Gil00].

The foundation of our Bayesian analysis is Bayes' formula which allows to compute the a posteriori probability density function (PDF) $f_{\Theta|x}(\Theta|x)$ from the a priori

PDF, or "prior", $f_\Theta(\Theta)$ and the conditional PDF that describes the random process $f_{x|\Theta}(x|\Theta)$:

$$f_{\Theta|x}(\Theta|x) = \frac{f_{x|\Theta}(x|\Theta) \cdot f_\Theta(\Theta)}{\int f_{x|\Theta}(x|\Theta) \cdot f_\Theta(\Theta) d\Theta} \tag{2.45}$$

Accordingly, the a posteriori PDF $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$ for the vector case is computed:

$$f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x}) = \frac{f_{\mathbf{x}|\mathbf{\Theta}}(\mathbf{x}|\mathbf{\Theta}) \cdot f_{\mathbf{\Theta}}(\mathbf{\Theta})}{\int f_{\mathbf{x}|\mathbf{\Theta}}(\mathbf{x}|\mathbf{\Theta}) \cdot f_{\mathbf{\Theta}}(\mathbf{\Theta}) d\mathbf{\Theta}} \tag{2.46}$$

For illustration purposes we consider an example with a parameter space of dimension $N = 3$, with probabilities p_1, p_2 and $p_3 = 1 - p_1 - p_2$ (see also Fig. 2.15). Hence, the parameter $\mathbf{\Theta}$ is restricted to the triangle $\mathcal{H}$. Since the actual estimation problem is only of dimension $N - 1 = 2$, the a posteriori PDF $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$ is defined on an arbitrary subset of two elements of the vector $\mathbf{\Theta}$. If we choose elements Θ_1 and Θ_2, this results in a projection onto the triangle $\mathcal{H}'$ (see Fig. 2.20(a)). Furthermore, we assume that no a priori knowledge, except $\sum_{i=1}^{N} p_i = 1$, is available. Hence, the following a priori PDF $f_{\mathbf{\Theta}}(\mathbf{\Theta})$ is assigned:

$$f_{\mathbf{\Theta}}(\mathbf{\Theta}) := \begin{cases} 2 & \text{where} \quad \Theta_1 + \Theta_2 \leq 1,\ \Theta_1 > 0,\ \Theta_2 > 0 \\ 0 & \text{elsewhere} \end{cases} \tag{2.47}$$

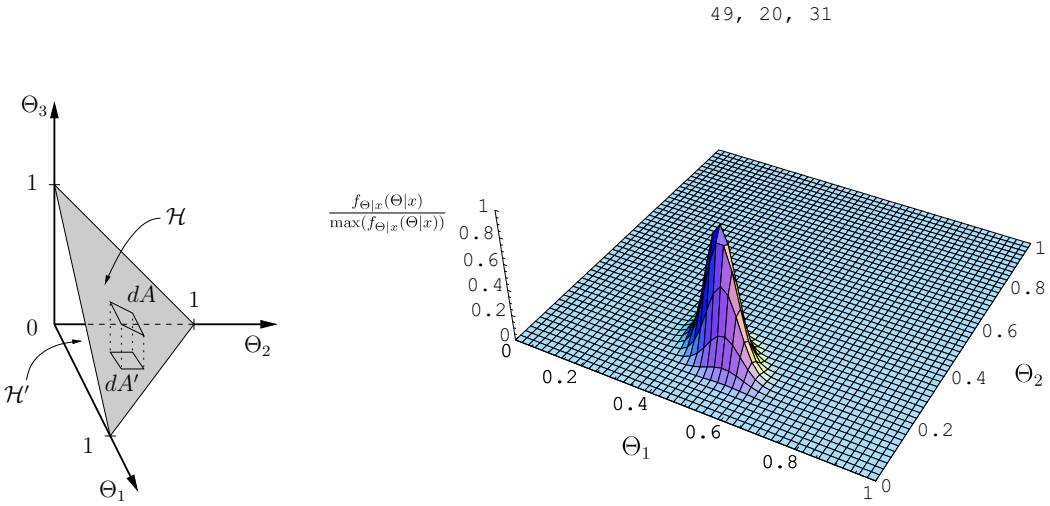
Fig. 2.20(b) shows the computed a posteriori PDF $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$ after 100 trials, with (true) probabilities $p_1 = 0.5, p_2 = 0.2, p_3 = 0.3$, resulting in an observation vector $\mathbf{x} = (49, 20, 31)$. The depicted surface is normalized with respect to the maximum value of the PDF. The mode (location of the maximum) is at $\Theta_1 = 0.49, \Theta_2 = 0.2$. This PDF contains the complete information that can be inferred about $\mathbf{\Theta}$, given the observed data $\mathbf{x}$. Fig. 2.21 shows the actual "learning process" of the estimator, i.e. how, from the very beginning, the knowledge about the parameter $\mathbf{\Theta}$ increases with each observation.

Before the first trial ($\mathbf{\Theta} = (0, 0, 0)^T$), only the knowledge about the parameter $\mathbf{\Theta}$ given by the a priori PDF $f_{\mathbf{\Theta}}(\mathbf{\Theta})$ is available (see eq. 2.47; note that all surfaces are normalized with respect to the maximum of $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$).

After the first trial ($\mathbf{\Theta} = (1, 0, 0)^T$), it is a logical consequence to assign a zero probability to all candidate parameters $\mathbf{\Theta}$ whose element Θ_1 is zero. The a posteriori PDF expresses exactly this knowledge.

After the fourth trial ($\mathbf{\Theta} = (3, 1, 0)^T$), the a posteriori PDF expresses exactly the fact that the element Θ_2 cannot become zero. The same effect occurs after the seventh trial ($\mathbf{\Theta} = (5, 1, 1)^T$), when $\Theta_3 = 0.0$ becomes impossible.

In the subsequent trials the mode of the a posteriori PDF moves according to the observations and the peak becomes narrower, thus expressing increasing confidence in the knowledge.



(a) projection $\mathcal{H} \mapsto \mathcal{H}'$ (b) (normalized) a posteriori PDF $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$

Figure 2.20: *Resulting a posteriori PDF $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$ after 100 trials, with (true) probabilities $p_1 = 0.5, p_2 = 0.2, p_3 = 0.3$, and observation vector $\mathbf{x} = (49, 20, 31)$. The depicted surface is normalized with respect to the maximum value of the PDF. The mode (location of the maximum) is at $\Theta_1 = 0.49, \Theta_2 = 0.2$.*

Since the a posteriori PDF contains the complete knowledge that can be accumulated by observing the data it is possible to derive laws that generate point estimates from the a posteriori PDF as well.
The most obvious choice is the *mode* (location of the maximum) of the a posteriori PDF, which results in the definition of the *maximum a posteriori (MAP)* estimator:

Definition 2.17 (Maximum a posteriori (MAP) estimator) *The estimator that chooses the mode of the a posteriori probability density function $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta} \,|\, \mathbf{x})$ is termed maximum a posteriori (MAP) estimator.*

$$\hat{\mathbf{\Theta}}_{\mathrm{MAP}} = \arg\max_{\mathbf{\Theta}} f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta} : | \, \mathbf{x}) \tag{2.48}$$

For a *uniform* a priori probability density function $f_{\mathbf{\Theta}}(\mathbf{\Theta})$ the mode (location of the maximum) of the a posteriori probability density function $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta} : | \, \mathbf{x})$ coincides with the mode of the likelihood function $\mathrm{L}(\mathbf{x}; \mathbf{\Theta})$. Hence, in the case of a uniform a priori probability density function, the MAP estimator is equivalent to the maximum likelihood estimator.

Another obvious choice is the *expected value* or "mean" of the a posteriori PDF, which results in the definition of the *minimum mean square error (MMSE)* estimator:

Definition 2.18 (Minimum mean square error (MMSE) estimator) *The estimator that chooses the expected value of the a posteriori probability density function $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta} : | \, \mathbf{x})$ is termed minimum mean square error (MMSE) estimator.*

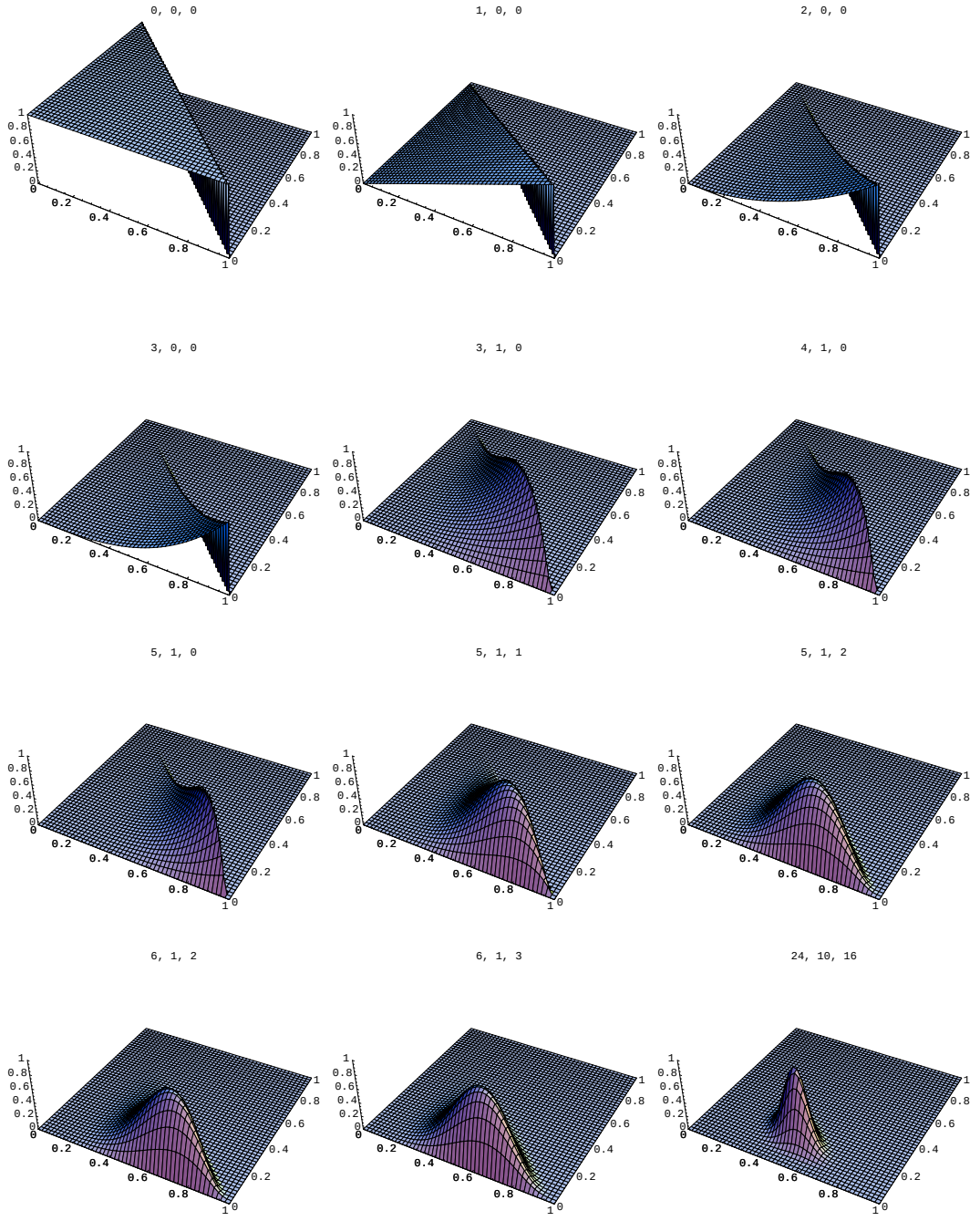


Figure 2.21: *The actual "learning process" of the estimator is captured in the evolution of the a posteriori PDF* $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$. *The sequence starts before any observation is made* $n = 0,\ \mathbf{x} = (0,0,0)^T$. *The observation vector* $\mathbf{x}$ *is noted above the corresponding surfaces. (note: all surfaces are normalized with respect to the maximum of* $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$*), please see Fig. 2.20(b) for axis labels)*

$$\hat{\boldsymbol{\Theta}}_{\mathrm{MMSE}} = \int \boldsymbol{\Theta} \cdot f_{\boldsymbol{\Theta}|\mathbf{x}}(\boldsymbol{\Theta} \mid \mathbf{x})\, d\boldsymbol{\Theta} \tag{2.49}$$

We apply eq. 2.49 for the estimation of the parameter of the binomial distribution:

$$\hat{\Theta}_{\mathrm{MMSE}} = \int \Theta \cdot f_{\Theta \mid x}(\Theta \mid x)\, d\Theta \tag{2.50}$$

Applying Bayes' formula (eq. 2.45) yields

$$\hat{\Theta}_{\mathrm{MMSE}} = \int \Theta \cdot \frac{f_{x|\Theta}(x|\Theta) \cdot f_{\Theta}(\Theta)}{\int f_{x|\Theta}(x|\Theta) \cdot f_{\Theta}(\Theta) d\Theta}\, d\Theta. \tag{2.51}$$

The assumption of absent a priori information is represented by a uniform a priori PDF:

$$f_{\Theta}(\Theta) = \begin{cases} 1 & \text{where} \quad 0 \leq \Theta \leq 1 \\ 0 & \text{elsewhere} \end{cases} \tag{2.52}$$

For the binomial distribution this yields

$$\hat{\Theta}_{\mathrm{MMSE}} = \int_0^1 \Theta \cdot \frac{\binom{n}{x_1} \cdot \Theta^{x_1} \cdot (1-\Theta)^{n-x_1}}{\int_0^1 \binom{n}{x_1} \cdot \Theta^{x_1} \cdot (1-\Theta)^{n-x_1}\, d\Theta}\, d\Theta. \tag{2.53}$$

$$\text{With} \quad \int_0^1 x^{\alpha} \cdot (1-x)^{\beta}\, dx = \frac{\Gamma(\alpha+1) \cdot \Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}, \quad \text{[GR65]} \tag{2.54}$$

(Note: The integral in eq. 2.54 is actually the definition of the beta function)

where $\Gamma(x+1) = x!,\ x \in \mathbb{N}$,

we simplify to

$$\begin{aligned} \hat{\Theta}_{\mathrm{MMSE}} &= \frac{\dfrac{\Gamma(x_1+2) \cdot \Gamma(n-x_1+1)}{\Gamma(x_1+1+n-x_1+2)}}{\dfrac{\Gamma(x_1+1) \cdot \Gamma(n-x_1+1)}{\Gamma(x_1+n-x_1+2)}} = \frac{\Gamma(x_1+2) \cdot \Gamma(n+2)}{\Gamma(n+3) \cdot \Gamma(x_1+1)} \\ &= \frac{(x_1+1)! \cdot (n+1)!}{(n+2)! \cdot x!}. \end{aligned} \tag{2.55}$$

Since $x! = (x-1)! \cdot x,$ we finally obtain the MMSE estimator:

$$\hat{\Theta}_{\mathrm{MMSE}} = \frac{x+1}{n+2} \tag{2.56}$$

A comparison with eq. 2.30 shows the equivalence with Laplace's law of succession. Hence, we have shown that Laplace's law of succession is the MMSE estimator for the binomial distribution. (The same is true for the generalized multinomial distribution.)

We have shown that both the maximum likelihood estimator and Laplace's law of succession have direct interpretations to certain outstanding quantities of the a posteriori PDF $f_{\boldsymbol{\Theta}|\mathbf{x}}(\boldsymbol{\Theta}|\mathbf{x})$. Fig. 2.22 illustrates the different estimates for a general a posteriori PDF.

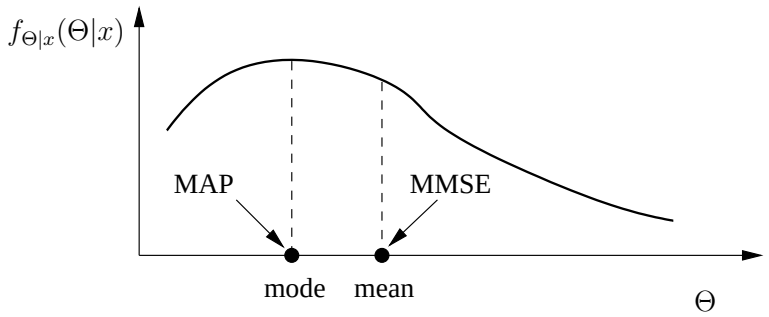


Figure 2.22: *Illustration of MAP and MMSE estimator for a general a posteriori PDF $f_{\Theta|x}(\Theta|x)$. The MAP estimator corresponds to the mode of the PDF, the MMSE estimator corresponds to the expected value or "mean" of the PDF.*

We have given this brief overview on the derivation of various estimators from a Bayesian perspective, since we strongly believe that this perspective provides additional insight into the *implicit* assumption of the estimators. Depending on the application of situation awareness it has to be decided if one of the point estimates or the complete a posteriori PDF is appropriate to convey the knowledge to subsequent decision instances.

We wish to emphasize an important relation between the structure of the situation space and the Bayesian estimation process: The structure of a situation space suggests to use the a posteriori probability density functions of situations that are similar to a particular situation for deriving reasonable a priori probability density functions for this particular situation.

For some types of resulting a priori probability density functions it may be difficult to derive closed form analytic estimators. For these cases both numerical integration as well as Monte-Carlo methods allow the efficient computation of estimates.

2.1.5.6 Estimation for Ranking and Prefetching Purposes

For certain applications the final result or "output format" of the estimation process is neither a point estimate nor a probability density function for the individual probabilities, but a sorted list, whose sorting criterion is the probability of its elements. In the absence of a priori information, the estimates of the discussed estimators (ML, MAP, MMSE) will never contradict each other regarding the *ranking* of the estimated event probabilities. If only pure ranking is the required result, no estimator has particular advantages.

Independently of the employed estimator the question of the accuracy of the resulting ranking occurs. The possible questions are: What is the probability that the chosen permutation (ranking) is *not* the true one? Or more generally: What is the probability for an arbitrary ranking to be the correct ranking? A model for this problem is depicted in Fig. 2.23:

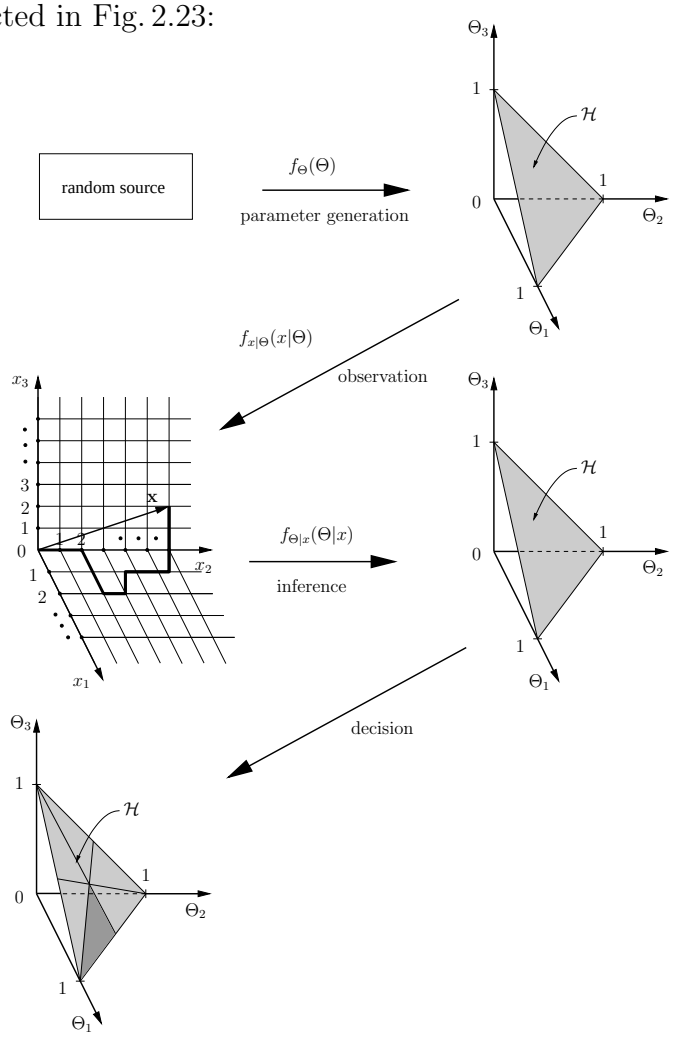


Figure 2.23: *Model for estimation and ranking process. A random source generates the true vector parameter* $\mathbf{\Theta}$ *on the triangle* $\mathcal{H}$ *of the parameter space. Successive random trials follow the conditional PDF* $f_{\mathbf{x}|\mathbf{\Theta}}(\mathbf{x}|\mathbf{\Theta})$ *and result in the observation vector* $\mathbf{x}$ *in the observation space. Bayes' rule is used to compute the a posteriori PDF* $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$*, defined on the triangle* $\mathcal{H}$ *in the parameter space. Based on this a posteriori PDF the probability of correctness of a particular hypothesis (dark-gray shaded area), i.e. a permutation, is computed.*

The true vector parameter $\mathbf{\Theta}$ is generated by a random source according to an a priori probability density function $f_{\mathbf{\Theta}}(\mathbf{\Theta})$. The generated parameter is confined to the triangle $\mathcal{H}$ of the parameter space. Successive random trials follow the conditional PDF $f_{\mathbf{x}|\mathbf{\Theta}}(\mathbf{x}|\mathbf{\Theta})$, in our case the multinomial distribution, and result in an observation

vector $\mathbf{x}$ in the observation space. Bayes' rule is used to compute the a posteriori PDF $f_{\mathbf{\Theta}|\mathbf{x}}(\mathbf{\Theta}|\mathbf{x})$, which is again defined on the triangle $\mathcal{H}$ in the parameter space. Based on this a posteriori PDF the probability of correctness of a particular hypothesis, i.e. a permutation in our case, is computed.
In general the number of possible rankings of N possible events is the number of possible permutations of an N-element set, which is $N!$. For our example $3! = 6$ different rankings exist. Each permutation corresponds to a particular subset of the points in $\mathcal{H}$.

A statement, e.g. $\Theta_1 = \Theta_2$, corresponds to a set of points on $\mathcal{H}$ that form a line splitting $\mathcal{H}$ in two subsets. For all the points in one subsets the binary statements $\Theta_1 > \Theta_2$, for the points in the other subset the statement $\Theta_1 < \Theta_2$ is true. For $N = 3$ possible events Fig. 2.24 shows the plane $\mathcal{H}$, to which the vector parameter $\mathbf{\Theta}$ is restricted from "top-view" in more detail (please compare Fig. 2.15.)

The subset that corresponds to a particular ranking, e.g. $\Theta_1 > \Theta_2 > \Theta_3$ (gray-shaded area in Fig. 2.24) is constructed by intersecting the subsets of the binary statements. The probability of a particular ranking to be the correct ranking is computed by integrating over all points in $\mathcal{H}$ for which the statement corresponding to the ranking is true.

$$\begin{aligned}\Pr\{\Theta_i \geq \Theta_j \geq \ldots\} &= \int_{\mathcal{H}} I\langle\Theta_i \geq \Theta_j \geq \ldots\rangle \cdot f(\mathbf{\Theta}|\mathbf{x}) \\ &= \int_0^1 \cdots \int_0^1 I\langle\Theta_i \geq \Theta_j \geq \ldots\rangle \cdot f(\mathbf{\Theta}|\mathbf{x})\, d\Theta_1 \ldots d\Theta_{N-1} \end{aligned} \tag{2.57}$$

where

$$I\langle\Theta_i \geq \Theta_j \geq \ldots\rangle = \begin{cases} 1 & \text{if} \quad \Theta_i \geq \Theta_j \geq \ldots \quad \text{is true} \\ 0 & \text{else} \end{cases} \tag{2.58}$$

From an overall system design perspective it may be considered desirable to give application developers, in addition to the a posteriori PDF, a very compact and easily understandable interface. Eq. 2.57 enables us to provide this kind of interface by providing a hypothesis and the corresponding probability of this hypothesis to be correct.

We now consider the situation space model and its analysis to be sufficiently advanced to proceed with our application example.

So far, we have kept the presentation and discussion of our situation space model and its features as general as possible in order to facilitate its adaptability to various forms of context aware applications. We will now pick a particularly interesting application of context awareness, namely the proactive retrieval of content, and investigate it under the assumption of a situation aware hypertext system that is able to provide probabilities for the user's document viewing activity.

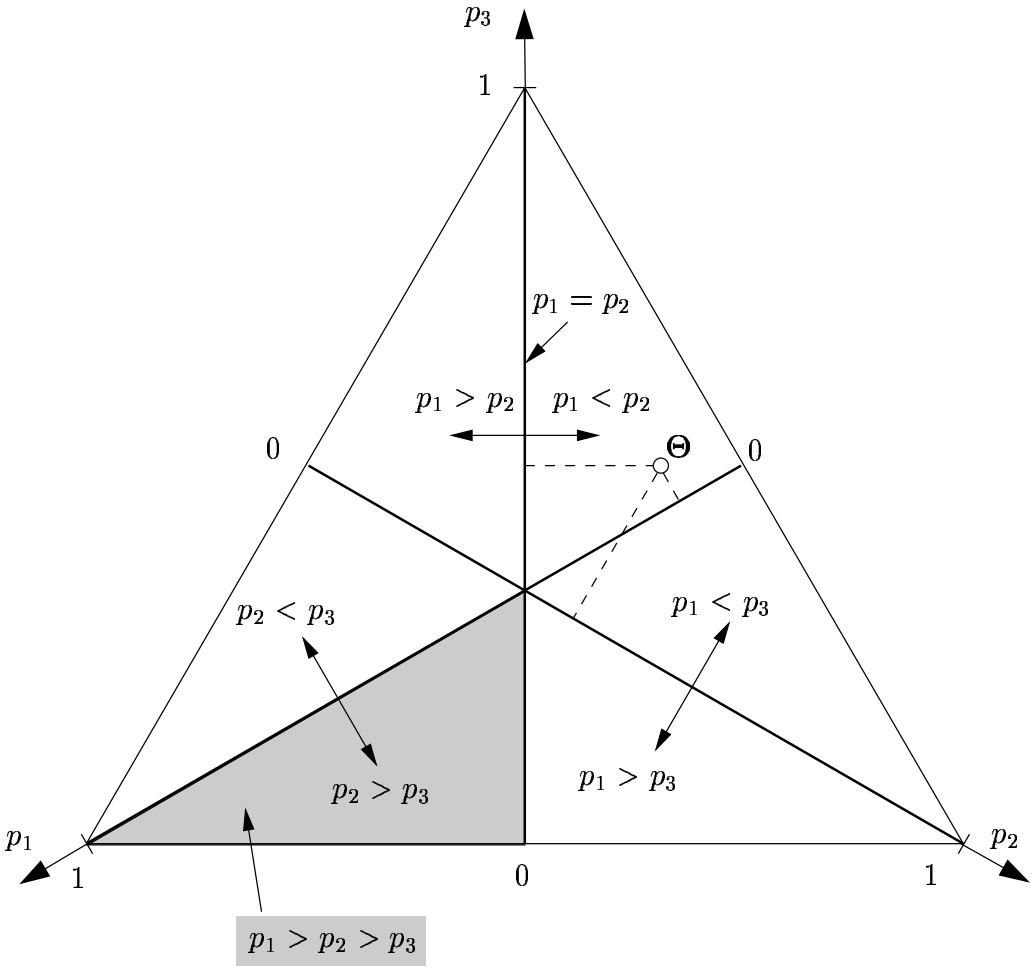


Figure 2.24: *Perpendicular view on triangle $\mathcal{H}$ (compare Fig. 2.15, note: the plane is not transparent, there is no perspective in this illustration.). For problems involving ranking, $\mathcal{H}$ is partitioned into segments corresponding to $N!$ possible permutations. A statement, e.g. $\Theta_1 = \Theta_2$, corresponds to a set of points on $\mathcal{H}$ that forms a line, splitting $\mathcal{H}$ in two subsets. For all the points in one subsets the binary statements $\Theta_1 > \Theta_2$, for the points in the other subset the statement $\Theta_1 < \Theta_2$ is true. The subset that corresponds to a particular ranking, e.g. $\Theta_1 > \Theta_2 > \Theta_3$ (gray-shaded area) is constructed by intersecting the subsets of the binary statements. Note: In the depicted case for $N = 3$ probabilities, only $N - 1 = 2$ degrees of freedom exist.*

2.2 Situation-Aware Prefetching

Our main field of application for the previously introduced situation awareness shall now be the technique of prefetching of content in hypertext systems[6]. In a conventional hypertext system a user who requested a specific document has to wait for the time it takes to transport the document over the network. As the term suggests,

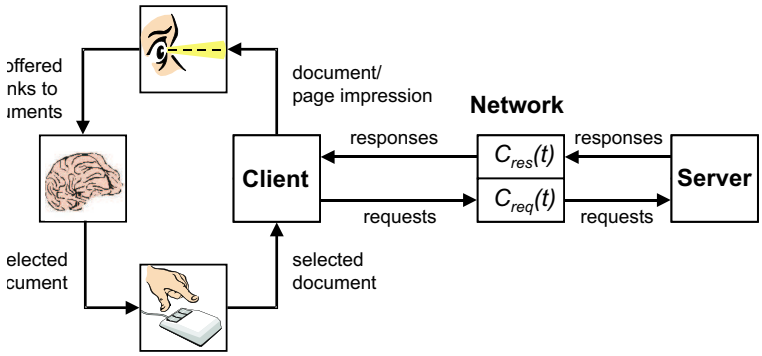


Figure 2.25: *The components of a hypertext system, namely server, network, client and user and their interactions form a closed-loop system. Upon the user's selection of a particular document the client sends a request for the content, via the network, towards the server. The server responds by sending the requested content towards the client. The duration of the transport of requests and responses depend on their volume and the actual network condition. Upon arrival of the content the client presents the document to the user, who, after contemplating the resulting page impression and extracting the desired information, selects another document.*

prefetching tries to transport the content towards the user *before* it is actually requested, thus saving the user from unwanted waiting. This property seems to be the ideal remedy for a widely prevailing problem in mobile wireless data communications: The availability of data communication in modern wireless communication networks such as GSM-(HS)CSD/GPRS or UMTS and increasing processing and display capabilities of mobile end devices has enabled the proliferation of hypertext systems to mobile applications such as WAP, i-mode, and recently also the WWW. However, in contrast to early expectations, the user acceptance of these applications has been relatively limited, compared to the astonishing success of the early World Wide Web. A major reason of user dissatisfaction is the poor perceived performance, i.e. much longer waiting times compared to what users are accustomed to from their desktop experience.

[6]We use the term "prefetching", since it is a well established label for the research topic. However, we do not wish to imply that the investigation is limited to protocols or systems in which a particular party pulls the content. In fact, the theoretical investigation in this section is completely agnostic whether the content is pushed or pulled. Therefore the term "proactive retrieval of content", or even more generally "proactive allocation of information" might be more appropriate.

The traditional approach to improve the performance by an increase of the data rate of the communication system usually results in an increase of necessary spectral bandwidth. Unfortunately, radio spectrum in suitable frequency ranges is an inherently limited resource. As a consequence of this fact the available capacities and datarates for mobile hypertext applications are and will most likely continue to be considerably lower (by orders of magnitude) than what is available for accessing a hypertext system from a fixed terminal.

We have already indicated that prefetching is a potential remedy against the resulting lack of perceived performance, since it actually "hides" the network's latency from the user. Fig. 2.26 illustrates this effect:

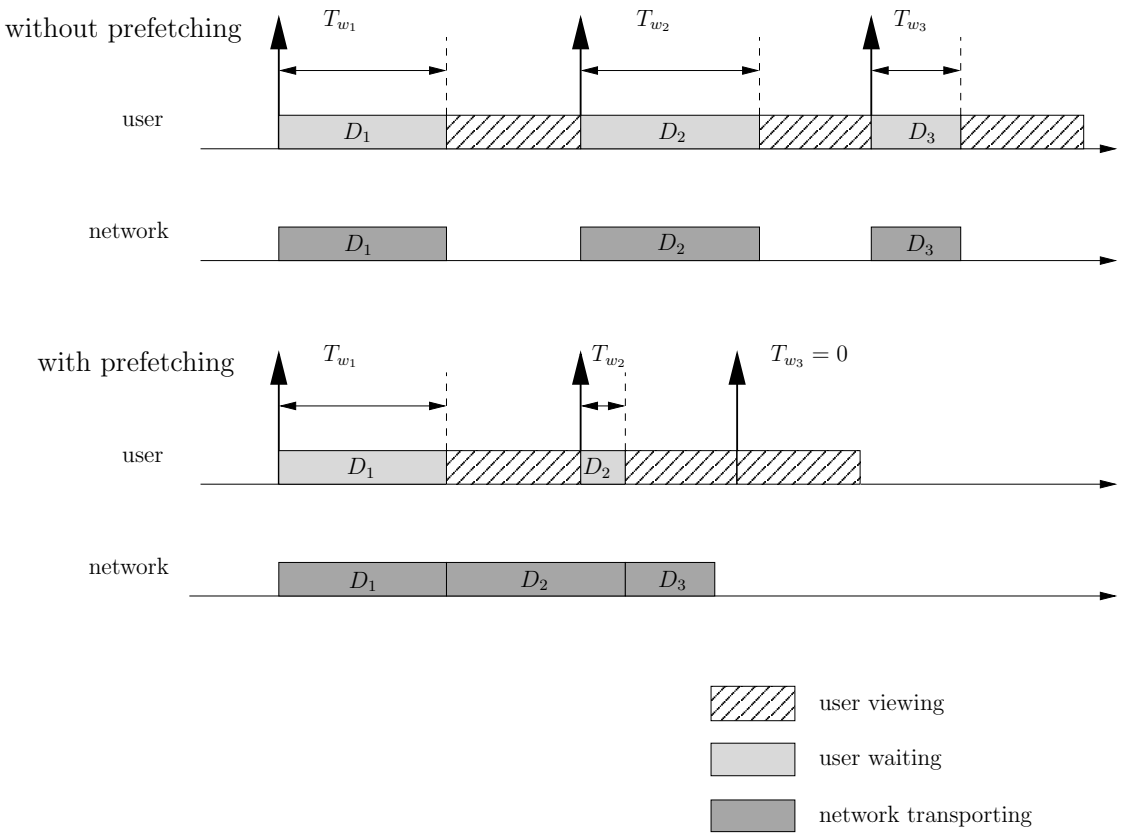


Figure 2.26: *Influence of ideal prefetching on the request-transport/wait-view cycle. If no prefetching is applied, the user has to wait during the time it takes to transport a requested document. While the user is viewing a document, the network resource is unused. If ideal prefetching is applied, the network resource is used to transport the subsequent document during the viewing time of its predecessor. Hence, the waiting time for a document is reduced by the viewing time for its predecessor or becomes zero, if the viewing time exceeds the time necessary for the transport.*

If no prefetching is applied, the closed-loop hypertext system directly results in the following request-transport/wait-view cycle: After requesting document D_1, the user has to wait a certain time (T_{w_1}) until the document is completely transported over the network. After completion of the transport the network remains unused while the user is viewing document D_1, until the same sequence is repeated when the user selects the next document D_2.
When ideal prefetching is applied, the network's capacity which was unused during the viewing times, is used to transport the next document before the user requests it. This way, the waiting time of the user is reduced by the viewing time. If the time for the transport of a document is shorter than the actual viewing time of its predecessor, no waiting is necessary (document D_3, $T_{w_3} = 0$).
Unfortunately, like any other remedy, prefetching may cause unwanted and adverse side-effects if it is applied carelessly: The choice of content a user will request is subject to his free will which is not (completely) foreseeable by anybody or anything. Instead, its nature is more or less random if observed from the outside. Therefore, many implementations of prefetching for hypertext systems decide, by examination of the documents' hyper-reference structure, to prefetch all content that will possibly be requested by the user. Consequently, this "blind" prefetching results in the transportation of many documents that will most likely not be requested by the user but congest the network resource.
This is where the cross-fertilization of the two techniques of prefetching and situation awareness generates its beneficial properties. The choice of documents a user is requesting strongly depends on his current situation. While "blind" prefetching cannot use information pertaining to the user's current situation, situation aware prefetching is able to employ all available knowledge about the situation to assign a suitable probability to each document. In our previously introduced model for situation awareness the request for a particular document maps to a consequence $\beta_{\sigma_j,i}$ (cf. 2.1.3.2) of the current situation σ_j. Among the other aspects that determine the current situation the document which is currently being viewed by the user (page impression) is likely to have a strong influence on the consequences' probabilities and is therefore a sensible choice for an aspect in the design of a situation space applied for prefetching (compare Section 2.1.4).

2.2.1 Analytical Model

We have pointed out how the previously introduced model of a situation space with symptoms and consequences is mapped to the application domain of information retrieval in a hypertext system. Since we consider it necessary to achieve a deep understanding of the application domain itself, in order to thoroughly understand the relations between the situation model and the application, we have performed an in-depth analysis of the properties of prefetching from a probabilistic viewpoint. This analysis has evolved to a contribution to the research in prefetching that is also considered valuable without the perspective of context awareness. Hence we will present

this analysis in a form that is useful also to readers only interested in the prefetching topic but not in context awareness. We will use the abstraction of document probabilities and "hide" their origin as the probabilities of consequences. However, we have to keep in mind that the knowledge about the documents' probabilities can only be achieved by a representation of the user's behavior in a situation space, continuous observation of the user's actions and estimation of the model probabilities.

We will attempt to answer several fundamental questions, such as:

1. What criterion should we apply to determine the sequence of speculatively retrieved documents?
2. Do we prefer small documents over large documents?
3. How does the distribution of probabilities influence the achievable reduction in waiting time T_w?
4. How much additional traffic is caused by prefetching?
5. Should we refrain from prefetching documents that are very unlikely to be requested?

We assume that in a certain situation, in which the user views the document D_0, he requests another document. The requested document will be randomly chosen out of a set of N documents $D_i,\ i = 1 \ldots N$, with probability $p_i = \Pr\{\boldsymbol{D} = D_i\}$, size V_i and data rate C_i for the retrieval of this document[7].

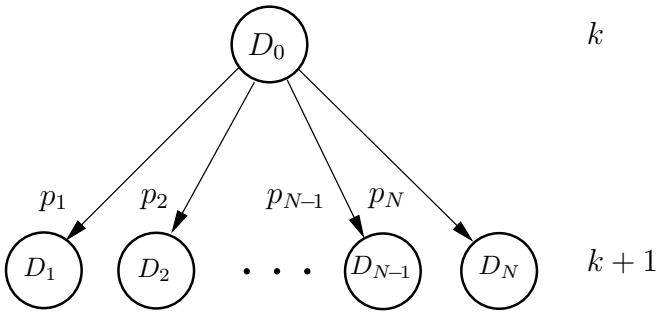


Figure 2.27: *Illustration for document selection process. The user views a document D_0. He chooses a subsequent document $D_i,\ i = 1 \ldots N$ with probability $p_i = Pr\{\boldsymbol{D} = D_i\}$.*

It is conjectured that prefetching, if performed intelligently, will reduce the average time a user has to wait for the retrieval of documents. Nevertheless the *speculative*

[7]For the classical mobile network scenario, which will be defined in Section 3.1, where the mobile link is always available and its data rate dominates the overall data rate all documents experience the same data-rate $C_i = r_M,\ \forall\, i$.

nature of prefetching will sometimes result in the transport of unnecessary documents. Hence the average transported volume will increase. In the following we will present a quantitative analysis of both effects.

2.2.2 Influence on Waiting Time

To achieve a first understanding we start with a reduced, minimalistic variant of the problem, consisting of only two distinct documents and a known time instant at which the user states his true selection. Subsequently, we will generalize this problem to an arbitrary number of documents and random time instant t_R for the end of the viewing time for the preceding document.

2.2.2.1 Two-Document Problem with Known Request Time

We consider two distinct candidate documents D_1 and D_2, with volumes V_1, V_2, and probabilities p_1, p_2. One of these documents shall be requested by the user at a known time instant t_R, the end of the viewing time for the preceding document. The duration of the viewing time shall be $t_R - t_0$. The documents will be transported over the network resource with a constant data rate C. The network resource shall be exclusively dedicated to a single user. The moment the true request from the user occurs, the prefetching activity is interrupted and the response for the user request is transported exclusively.

This minimalistic case allows to illustrate several possible strategies and will help us to understand the influence of these strategies on the expected value $E\{T_w\}$ for the time the user has to wait. The following five strategies can be distinguished:

- Do not prefetch any document at all (Strategy A).
- Prefetch only document D_1 (Strategy B).
- Prefetch only document D_2 (Strategy C).
- First prefetch document D_1, if time is left prefetch D_2 afterwards (Strategy D).
- First prefetch document D_2, if time is left prefetch D_1 afterwards (Strategy E).

Strategy A resembles the absence of any prefetching and is used as a reference. In this initial analysis we are only investigating the influence of the various strategies on the average waiting time and do not consider any cost for the amount of bytes transferred. Nevertheless it should be noted that strategy A is, of course, the one that causes the least amounts of bytes transferred over the network resource.

The network resources' data rate shall be $C_1 = C_2 = C = 1$, the documents' volumes $V_1 = 1,\ V_2 = 2$ and their probabilities $p_1 = 0.45$ and $p_2 = 0.15$ respectively (arbitrary normalization factors with units bit/s and bit may be assumed).

The first document D_1 is three time more likely to be chosen than the second document D_2. However, the second document has twice the volume of the first document. Which document should be prefetched first?

To obtain a clearer picture, we will successively compute the expected value of the waiting time $E\{T_w\}$ as a function of t_R, the end of the viewing time. The resulting curves for all five strategies are depicted in Fig. 2.28.

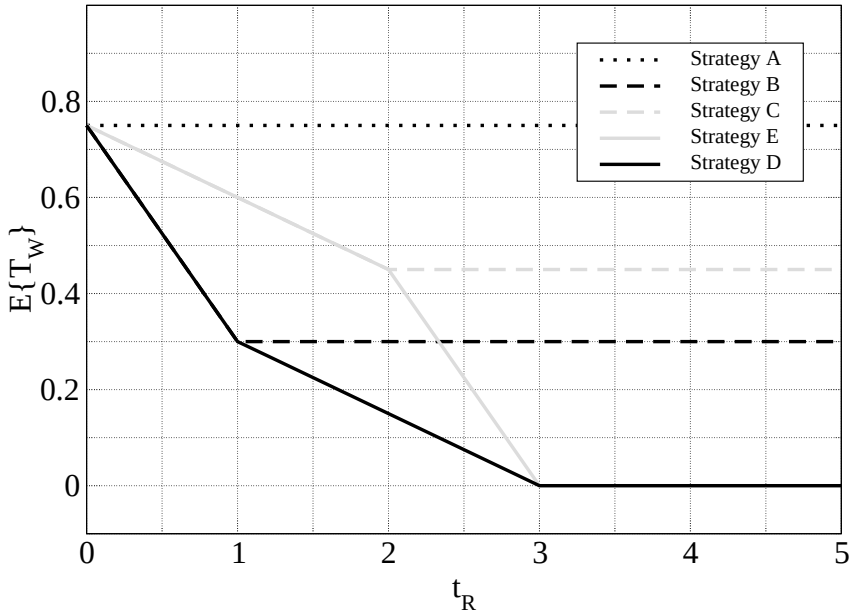


Figure 2.28: *Comparison of prefetching strategies for two candidate documents D_1, D_2. Depicted is the expected value of the waiting time $E\{T_w\}$ as a function of the end t_R of the viewing time. Without loss of generalization the begin of the viewing time shall be $t_0 = 0$ ($C = 1,\ V_1 = 1,\ V_2 = 2,\ p_1 = 0.45,\ p_2 = 0.15$).*

At $t_R = t_0$, the beginning of the viewing time, no prefetching has occurred. The expected value of the waiting time at $t_R = t_0$ is therefore

$$E\{T_w\}(t_R = t_0) = p_1 \cdot \frac{V_1}{C_1} + p_2 \cdot \frac{V_2}{C_2} \tag{2.59}$$

$$= 0.45 \cdot \frac{1}{1} + 0.15 \cdot \frac{2}{1} = 0.75, \tag{2.60}$$

independently of the chosen strategy.

For strategy A no document is prefetched at any time instant. Hence, the expected value of the waiting time $E\{T_{w_A}\}$ remains constant at this value for all t_R:

$$E_A\{T_w\}(t_R) = p_1 \cdot \frac{V_1}{C} + p_2 \cdot \frac{V_2}{C} \tag{2.61}$$

Following strategy B, the retrieval of the first document D_1 is started immediately. After V_1/C the first document is completely retrieved.

If the true request does not occur until D_1 is completely retrieved, the remaining expected value for the waiting time stays at

$$E_B\left\{T_w\right\}\left(t_R - t_0 > V_1/C\right) = p_2 \cdot \frac{V_2}{C}\,, \tag{2.62}$$

since the user will request document D_2 with probability p_2 and will have to wait for the time necessary to retrieve it, whereas a request for document D_1 does not result in any waiting time. Should the request occur during the speculative retrieval of document D_1, the remainder of document D_1 has to be retrieved with probability p_1, the complete document D_2 has to be retrieved with probability p_2. The necessary time to transport the remaining volume of document D_1 is $V_1/C-(t_R-t_0)$ if $t_R-t_0 \leq V_1/C$ and 0 if $t_R - t_0 > V_1/C$. Using a min()-operation allows us to distinguish between the two cases in one expression. We obtain for strategy B:

$$E_B\left\{T_w\right\}\left(t_R\right) = p_1 \cdot \left(\frac{V_1}{C} - \min\left(t_R - t_0, \frac{V_1}{C}\right)\right) + p_2 \cdot \frac{V_2}{C} \tag{2.63}$$

and analogously for strategy C:

$$E_C\left\{T_w\right\}\left(t_R\right) \;=\; p_1 \cdot \frac{V_1}{C} + p_2 \cdot \left(\frac{V_2}{C} - \min\left(t_R - t_0, \frac{V_2}{C}\right)\right) \tag{2.64}$$

If strategy D is chosen, sequential retrieval of both documents will be attempted. Until the time instant $t_0 + V_1/C$ this strategy is identical with strategy B. After this time instant, the expected value of the waiting time is further reduced until it reaches zero after the second document has been retrieved completely. The retrieval of document D_2 does not start before t_0+V_1/C. Therefore, if the document is requested at time instant t_R, the time duration that was available to retrieve document D_2 was $t_R-(t_0+V_1/C)$, if the user request arrives at $t_R-t_0 \geq V_1/C$ and 0, if $t_R-t_0 < V_1/C$. A max()-operation allows us to distinguish between the two cases in one expression. As in strategy C, the amount of time that can be saved by speculative retrieval of D_2 cannot exceed the necessary duration for its complete retrieval V_2/C. This is represented again by a min()-operation, thus yielding

$$\begin{aligned} E_D\left\{T_w\right\}\left(t_R\right) \;=\; & p_1 \cdot \left(\frac{V_1}{C} - \min\left(t_R - t_0, \frac{V_1}{C}\right)\right) + \\ & + p_2 \cdot \left(\frac{V_2}{C} - \min\left(\max\left(t_R - \left(t_0 + \frac{V_1}{C}\right), 0\right), \frac{V_2}{C}\right)\right) \end{aligned} \tag{2.65}$$

And finally switching documents D_1 and D_2 yields for strategy E

$$E_E\{T_w\}(t_R) = p_1 \cdot \left(\frac{V_1}{C} - \min\left(\max\left(t_R - \left(t_0 + \frac{V_2}{C}\right), 0\right), \frac{V_1}{C}\right)\right) + \\ + p_2 \cdot \left(\frac{V_2}{C} - \min\left(t_R - t_0, \frac{V_2}{C}\right)\right). \tag{2.66}$$

Fig. 2.29 shows a comparison of equations 2.65 and 2.66 for strategies D and E with a Monte-Carlo simulation experiment. The curves yielded by simulation and the theoretical considerations match well.

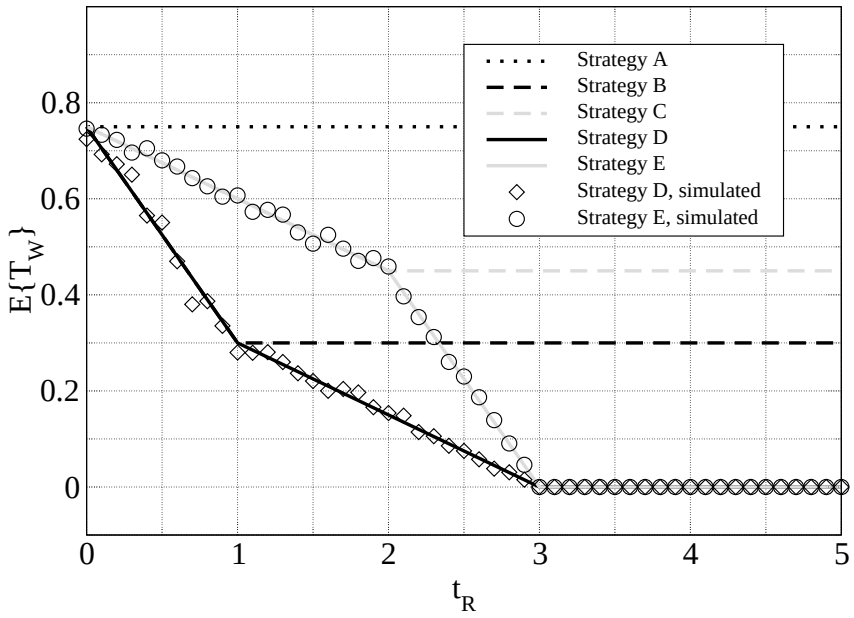


Figure 2.29: *Results of Monte-Carlo simulations showing the influence of distinct prefetching strategies on the average value of the waiting time ($C = 1,\ V_1 = 1,\ V_2 = 2,\ p_1 = 0.45,\ p_2 = 0.15$).*

2.2.2.2 Arbitrary Number of Documents with Unknown Request Time

The previous example hinted an influence of the order in which the documents are transported. For the case of two documents only two orders are possible. We can choose among them by simple swapping. If the number of documents is larger than two, it is convenient to use the concept of permutations to describe all the possible orders. The number of possible orders of an N-element set is $N!$. We write ${}^k element_i$ and ${}^k value$ to denote the i-th element of a set or a value calculated for a

particular permutation k. The index k is used to denote the position of the particular permutation among any defined order (e.g. lexicographic) of all possible permutations. Hence, ${}^{k}p_i$ and ${}^{k}V_i$ are the i-th elements of the ordered set of the documents' probabilities and sizes *after* these sets have been ordered according to the k-th permutation with permutations numbered in the specified order.

If the initial order is represented in a vector ${}^{1}\boldsymbol{x}$, we express a permutation as the multiplication of this vector with a permutation-matrix $\mathbf{P_k}$.

$$ {}^{k}\boldsymbol{x} = \mathbf{P_k} \cdot {}^{1}\boldsymbol{x} \tag{2.67} $$

The matrix that does not change the order is an $N \times N$ identity-matrix.

$$ {}^{k}\boldsymbol{x} = \mathbf{I}_{N,N} \cdot {}^{k}\boldsymbol{x} \tag{2.68} $$

In order to extend the previous results from 2 to N documents, we explicitly state the assumed retrieval policy. At the time the true user request arrives, three distinct cases can be distinguished:

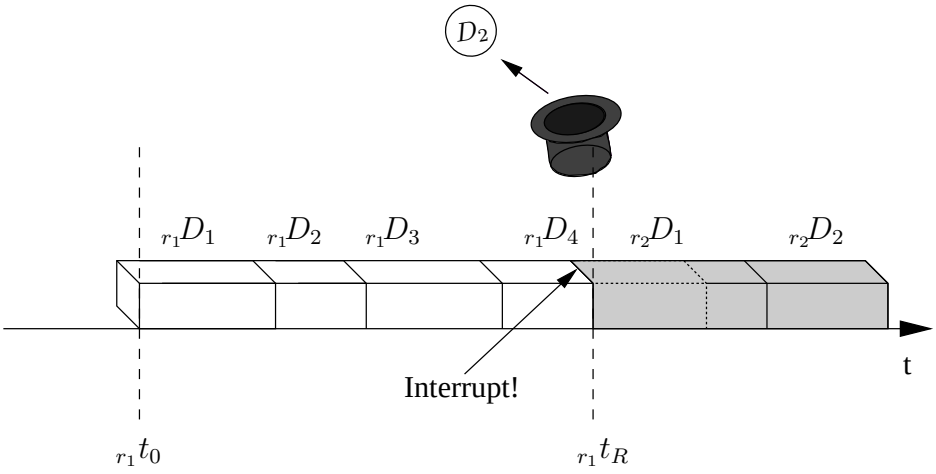


Figure 2.30: *Retrieval policy (complete hit, case 1) – The requested document has already been retrieved completely. It is immediately presentable to the user without any waiting time. Speculative retrieval of the new set of probable documents starts immediately.*

case 1) The requested document has already been retrieved completely. In this case the requested document is immediately presentable to the user. The current retrieval of a document is interrupted and retrieval of the first document of the next set of documents is started immediately (see Fig. 2.30).

case 2) The requested document is the one that is currently being loaded. In this case the retrieval is continued until the document is completely available. Afterwards retrieval of the next set starts (Fig. 2.31).

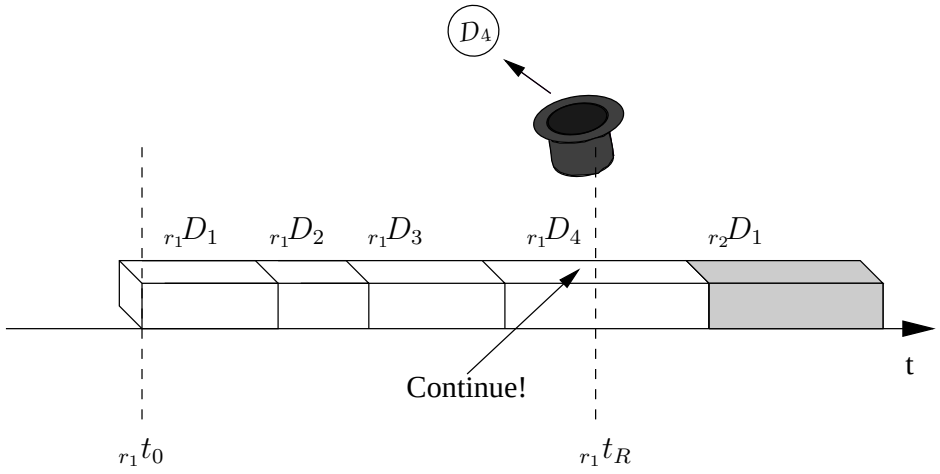


Figure 2.31: *Retrieval policy (partial hit, case 2) – The requested document is currently being retrieved. The waiting time is reduced by the amount of already retrieved data. Retrieval of current document continues until completion.*

case 3) The requested document has not been loaded even partially. In this case the current retrieval of a document is interrupted and retrieval of the requested document is started immediately. Afterwards retrieval of the next set starts (see Fig. 2.32).

In contrast to this retrieval policy, Tuah [TKV97, Tua00] assumes the continuation of the current retrieval in case 3. The document requested by the user is retrieved afterwards. We do not assume this policy, since the document requested by the user is in fact known at this time instant (its probability of request is 1 under the given conditions), retrieval, or continued retrieval of any other document is sub-optimal.

Based on the previously described policy, we wish to calculate $E\left\{{}^{k}T_w \mid t_R\right\}$, the expected value for T_w for a chosen permutation k, under the condition that the request arrives at a certain time t_R

The expected value of the waiting time without any prefetching is the same for all permutations.

$$E\left\{T_w\right\}_{no\ prefetch} = \sum_{i=1}^{N} {}^{k}p_i \cdot \frac{{}^{k}V_i}{{}^{k}C_i} = \sum_{i=1}^{N} {}^{1}p_i \cdot \frac{{}^{1}V_i}{{}^{1}C_i} \quad \forall\ k. \tag{2.69}$$

At $t_R = t_0$ no prefetching has happened, hence

$$E\left\{{}^{k}T_w \mid t_R = t_0\right\} = E\left\{T_w\right\}_{no\ prefetch} \quad \forall\ k. \tag{2.70}$$

Independently of the chosen permutation the expected value for the waiting time reaches zero when all documents have been prefetched after T_{w_0}:

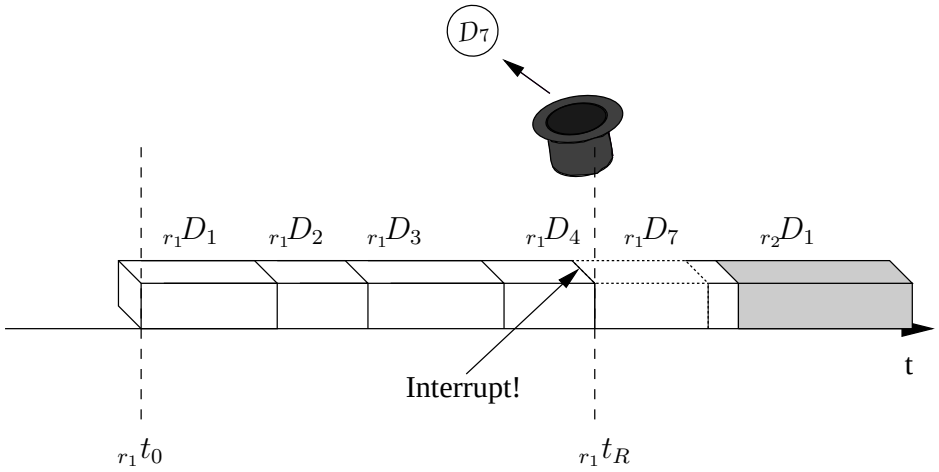


Figure 2.32: *Retrieval policy (complete miss, case 3) – The requested document has not been retrieved even partially. Retrieval of the current document is stopped immediately in favor of the truly requested document.*

$$E\left\{{}^{k}T_w \mid t_R - t_0 \geq T_{w_0}\right\} = 0 \quad \forall\ k, \tag{2.71}$$

with

$$T_{w_0} = \sum_{i=1}^{N} \frac{{}^{k}V_i}{{}^{k}C_i} \quad = \quad \sum_{i=1}^{N} \frac{{}^{1}V_i}{{}^{1}C_i} \quad \forall\ k. \tag{2.72}$$

Hence, we can state the following expression, which includes the cases $t_R = 0$, $t_R \geq T_{w_0}$ and $0 < t_R < T_{w_0}$:

$$E\left\{{}^{k}T_w \mid t_R\right\} = E\left\{T_w\right\}_{no\ prefetch} - E\left\{{}^{k}T_{gain} \mid t_R\right\}, \tag{2.73}$$

with

$$\begin{aligned} E\left\{{}^{k}T_{gain} \mid t_R\right\} \quad = \quad & \sum_{i=1}^{N} \Bigg({}^{k}p_i \cdot \min \Bigg\{ \max \Bigg\{ 0, \\ & t_R - t_0 - \Bigg(- \frac{{}^{k}V_i}{{}^{k}C_i} + \sum_{j=1}^{i} \frac{{}^{k}V_j}{{}^{k}C_j} \Bigg) \Bigg\}, \\ & \frac{{}^{k}V_i}{{}^{k}C_i} \Bigg\} \Bigg), \end{aligned} \tag{2.74}$$

We try to give an intuitive explanation for eq. 2.74: The total time available for prefetching is $t_R - t_0$. It is necessary to distinguish between three distinct cases:

case 1) the document has already been completely retrieved. Then the complete time $\frac{{}^{k}V_i}{{}^{k}C_i}$ necessary to retrieve it is gained, but not more.

case 2) the document is currently being retrieved. Subtracting the time invested on the other documents from the total time available for prefetching yields the time gained.

case 3) no attempt has been made to retrieve the document before t_R. In this case no time is gained, but the retrieval starts immediately at t_R. Hence, the minimum time gain is 0.

The three cases are represented by the min / max operation.

Finally the expected value is calculated by weighting with the documents' probabilities ${}^k p_i$ and summing up over all $i = 1 \ldots N$.

Eq. 2.74 is illustrated in Fig. 2.33, with $t_0 = 0$, for one arbitrary permutation k. The upper curve shows the expected value for the waiting time. It is composed of N linear segments. Each segment corresponds with the retrieval of one particular document.

With eq. 2.74 we have formulated the expected waiting time as a function of the request-time t_R. For the case of a human user the request-time has to be considered a continuous random variable following a probability density function $f_{t_R}(t_R)$ of the request time t_R (see lower plot in Fig. 2.33). In this case we compute the unconditional expected waiting time $E\{T_w\}$.

$$E\{T_w\} = \int_0^\infty f_{t_R}(t_R) \cdot E\left\{{}^k T_w \mid t_R\right\} dt_R. \tag{2.75}$$

In order to minimize $E\{T_w\}$ it is sufficient to choose the permutation that minimizes the gray-shaded area in Fig. 2.33. This can be proved, using the facts that $E\left\{{}^k T_w \mid t_R\right\}$ is monotonically decreasing for all permutations k, and that $f_{t_R}(t_R) \geq 0 \ \forall \ t_R$. With this in mind, and the observation that the slope of every segment depends only on the probability p_i of its document we can derive a simple two-step algorithm that minimizes the waiting time:

1. **sort** the documents with respect to their probability p_i.
2. **fetch** all documents sequentially .

It is important to notice that the optimum sequence only depends on the documents' probabilities *not* on their size or connection speed. With this result we are now able to answer two of the five questions we asked when beginning our analysis: The probabilities p_i should be applied as the criterion for determining the sequence. In order to minimize waiting time, it is not necessary or beneficial to prefer small documents over large documents.

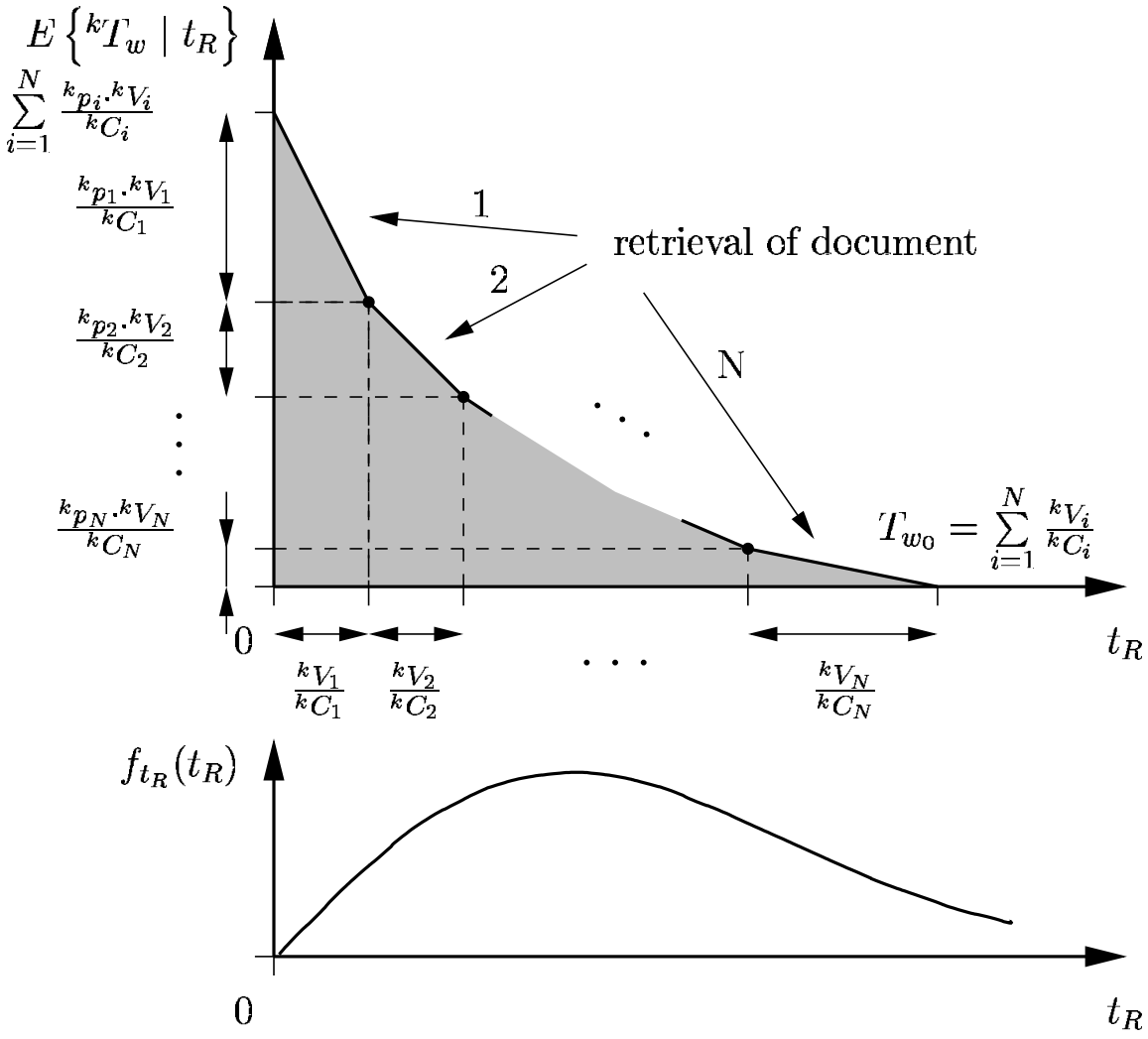


Figure 2.33: *Expected value for waiting time $E\{T_w \mid t_R\}$ and arbitrary probability density function $f_{t_R}(t_R)$ of the request time t_R.*

To answer the third question, i.e. to acquire an understanding for the influence of the probabilities and permutations we analyze an example. The parameters in the example shall be $N = 6$, $C_i, V_i = 1 \ \ \forall\, i = 1 \ldots N$. The probabilities p_i shall be distributed according to Zipf's Law.

Zipf's Law states that the probability of the i-th most likely event is proportional to $1/i$. This is also called the *strict* Zipf's Law. Many interesting experiments show a slight modification of this law. They can be more adequately modelled with a probability proportional to $1/i^{\alpha}$ for the i-th most likely event. The value α then typically takes a value of less than unity. For $\alpha = 1$ this modified law is equivalent to the strict Zipf's Law [BCF+99]. Fig. 2.34 shows the influence of α on the probability mass function.

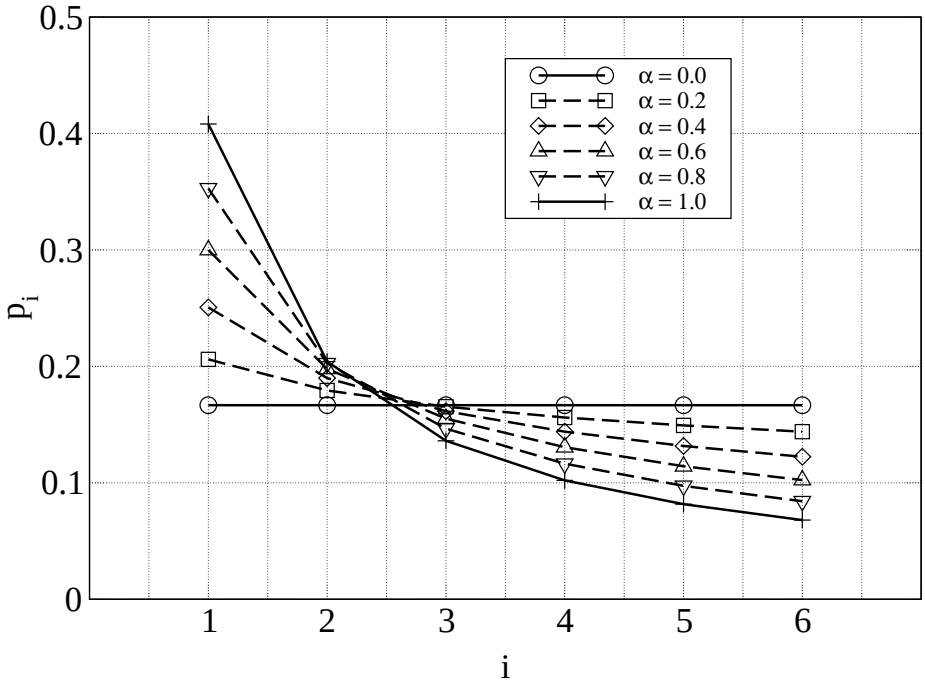


Figure 2.34: *Influence of the parameter α on the element's probability mass function for Zipf Distribution with $N = 6$ elements. Depicted is the probability p_i of the i-th element with $i = 1, \ldots, N$. For $\alpha = 0$ all elements are equally probable.*

We start the example with $\alpha = 1$, hence, $p_1 \approx 0.408$, $p_2 \approx 0.2041$, $p_3 \approx 0.1360$, $p_4 \approx 0.1020$, $p_5 \approx 0.0816$, and $p_6 \approx 0.0680$.

For the $N = 6$ distinct documents $6! = 720$ permutations are possible. Each permutation represents one possible prefetching strategy. Eq. 2.74 allows us to compute $E\left\{{}^{k}T_w \mid t_R\right\}$ for all 720 permutations. Fig. 2.35 shows the resulting curves.

We see that the sequence in which the documents are prefetched, strongly influences the resulting expected waiting times. If the sequence is strictly ordered according to the documents' probabilities, with higher probabilities being prefetched first, the expected waiting time is minimal for all request-times t_R. If the order is

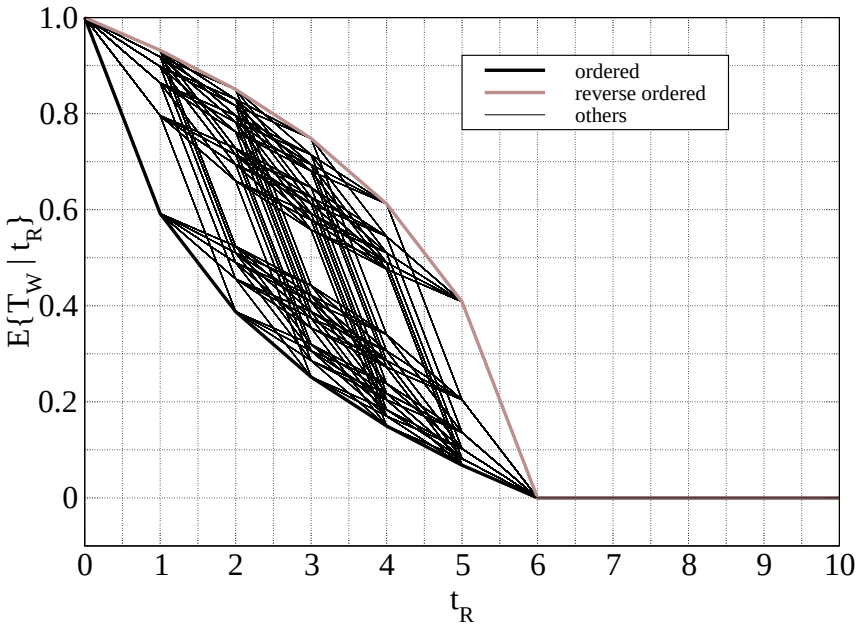


Figure 2.35: *Influence of prefetching strategies on the expected value of waiting time. For 6 candidate documents* $6! = 720$ *possible permutations or strategies exist. The strategy in which documents with higher probabilities are prefetched first (ordered) results in the minimum expected waiting time for all request-times* t_R*. The strategy in which the order is reversed results in the maximum expected waiting time of all prefetching strategies. All other strategies perform between these two extreme cases (*$C = 1$, $V_1..V_6 = 1$*, document probabilities following a Zipf distribution with* $\alpha = 1.0$*).*

reversed the expected waiting time is the maximum among all prefetching strategies. All other strategies result in expected waiting times between these two extreme cases.

Common sense suggests that the more pronounced the differences in the documents' probabilities p_i are, the better the performance of prefetching can get. The distribution according to Zipf's Law facilitates a quantitative analysis of this conjecture. By varying the parameter α we can adjust the distribution from being absolutely flat ($\alpha = 0$) to a stronger preference for the likelier documents when α rises to 1.0 (or higher). Decreasing α reduces the distance between best and worst strategy.

Fig. 2.36 shows the ordered (best) and reverse ordered (worst) strategies. We can see that for the case $\alpha = 0$, when all N documents have the same probability $p_i = 1/N$, no difference between the best and the worst strategy exists.

Monte-Carlo simulation experiment are performed to augment and extend the theoretical results. The parameters (p_i, C_i, V_i) are chosen to match the example.

In a first simulation the prefetching controller has a priori knowledge of the documents' probabilities p_i. The actual waiting times $T_{w_j}, j = 1 \ldots M$ are measured for $M = 1000$ trials. The mean value $\overline{T}_w = \frac{1}{M} \sum_{j=1}^{M} T_{w_j}$ is plotted in Fig. 2.37 for several request times t_R. Simulation results and theoretical analysis show good consensus.

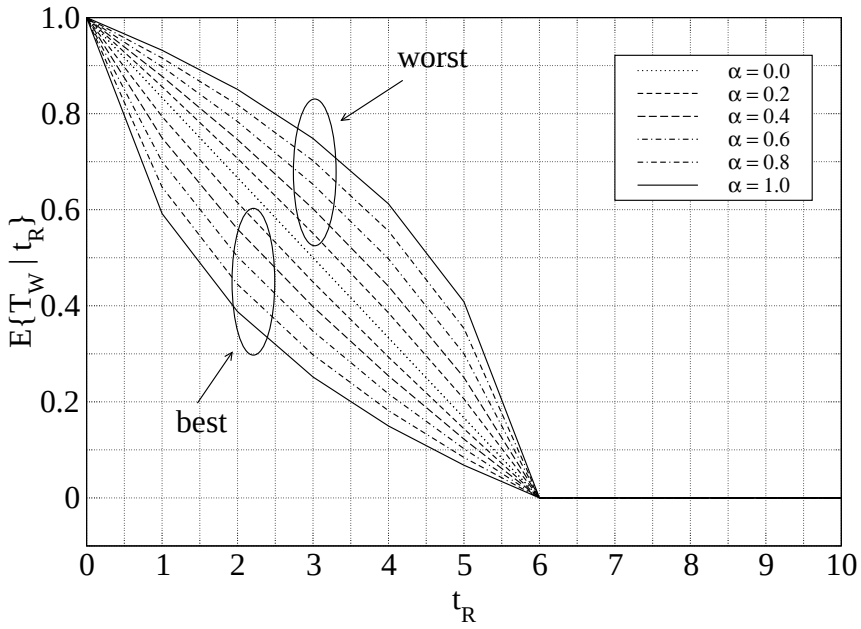


Figure 2.36: *Influence of the parameter α of the Zipf distribution on the expected value of waiting time for ordered and reverse ordered prefetching strategies. The difference between the two strategies increase with increasing α. For $\alpha = 0.0$ the differences diminish, since all probabilities are identical ($C = 1,\ V_1..V_6 = 1$).*

In a second simulation the request-time t_R is uniformly distributed over the interval $[\ 0, T_{w_0}]$ (This distribution simplifies interpretation of results. Other distributions e.g Poisson, Pareto may be used for more realistic scenarios). Furthermore, the prefetching controller has *no* a priori knowledge of p_i and uses the relative frequency of observed documents as estimations of p_i in this simulation. The learning behavior and the influence of α are shown in Fig. 2.38.

According to the theoretical analysis, performance of the prefetching controller should increase with the value of α. The prefetching controller exploits the asymmetries in the documents' probabilities. We can see for the particular setup of parameters the reduction of $\overline{T}_w$. It takes approximately 10 (20, 30) visits for $\alpha = 0.0$ ($\alpha = 0.5$, $\alpha = 1.0$) for the prefetching controller to gather enough observations for a good estimation of the probabilities p_i that is necessary to reach its asymptotic performance.

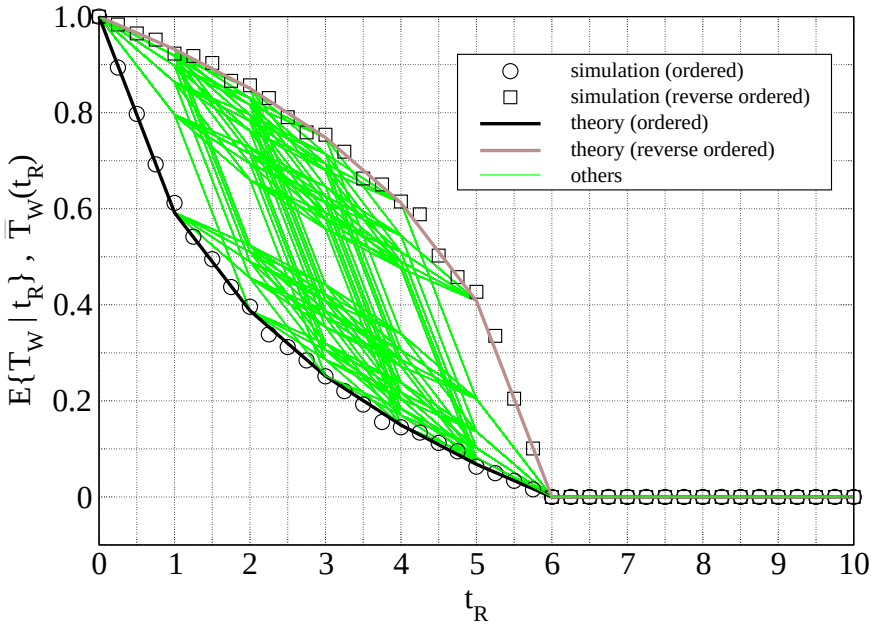


Figure 2.37: *Comparison of Monte-Carlo simulation results and theory. The average waiting time $\overline{T}_w$ of 1000 trials is plotted for multiple request-times t_R, showing good consensus between average waiting times and expected waiting times ($C_1 \ldots C_6 = 1, V_1 \ldots V_6 = 1$, document probabilities following a Zipf distribution with $\alpha = 1.0$).*

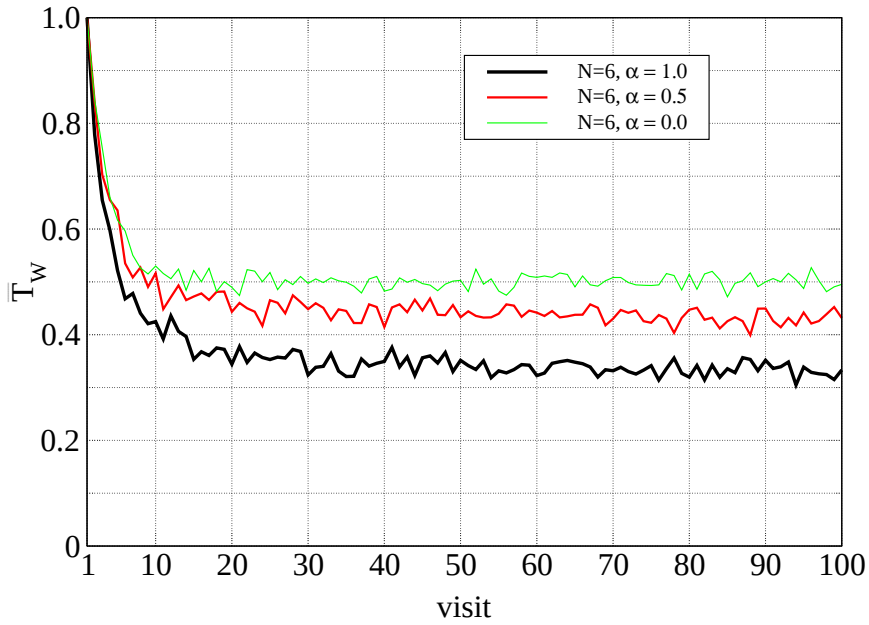


Figure 2.38: *Learning behavior of prefetching controller with no a priori knowledge of document probabilities with equally distributed t_R and varying parameter α ($C = 1,\ V_1..V_6 = 1$, document probabilities following a Zipf distribution with $\alpha = 1.0$).*

2.2.3 Influence on Transported Volume

An important property of speculative prefetching is the increase of traffic caused by the documents that have been transported over the network, but are never requested by the user. This increase in traffic is the price for the decrease in waiting time. We will analyze the nature of this increase, quantify it and will later make use of the obtained results to derive a threshold probability that corresponds to an adjustable user policy.
If no prefetching would be applied, the expected volume that would have to be transported over the network is

$$E\{V\}_{no\ prefetch} = \sum_{i=1}^{N} p_i \cdot V_i. \tag{2.76}$$

If prefetching is applied, the expected volume is increased, but cannot exceed the combined volume of all documents. This maximum is reached when all documents have been transported at T_{w_0} (see eq. 2.72).

$$E\left\{{}^{k}V \mid t_R = T_{w_0}\right\} = \sum_{i=1}^{N} V_i. \tag{2.77}$$

Similarly to eq. 2.73 we calculate $E\left\{{}^{k}V \mid t_R\right\}$, the expected value of the transported volume ${}^{k}V$ for a chosen permutation k, under the condition that prefetching is applied and the request arrives at a certain time t_R

$$E\left\{{}^{k}V \mid t_R\right\} = E\{V\}_{no\ prefetch} + E\left\{{}^{k}V_{additional} \mid t_R\right\}, \tag{2.78}$$

with

$$\begin{aligned} E\left\{{}^{k}V_{additional} \mid t_R\right\} \;=\; & \sum_{i=1}^{N} \Bigg({}^{k}(1-p_i) \cdot \min \Bigg\{ \max \Bigg\{ 0, \\ & {}^{k}C_i \cdot \left(t_R - t_0 - \left(-\frac{{}^{k}V_i}{{}^{k}C_i} + \sum_{j=1}^{i} \frac{{}^{k}V_j}{{}^{k}C_j} \right) \right) \Bigg\}, \\ & {}^{k}V_i \Bigg\} \Bigg), \end{aligned} \tag{2.79}$$

We try to give an intuitive explanation for eq. 2.79: equivalently to the explanation for eq. 2.74, the total time available for prefetching is $t_R - t_0$. Furthermore, it is again necessary to distinguish between the following three cases:

case 1) the document has already been completely retrieved. Then the complete volume ${}^{k}V_i$ has been transported, but not more.

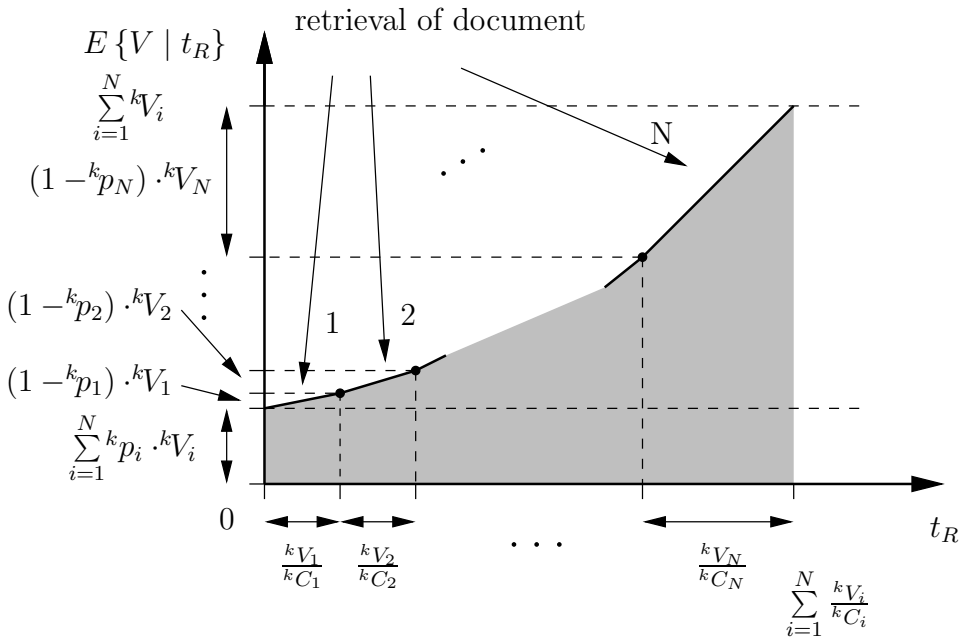


Figure 2.39: *Expected value for transported volume $E\{V \mid t_R\}$. At the beginning of the viewing time ($t_R = 0$) the expected value of the transported volume $E\{V \mid t_R = 0\}$ is $\sum_{i=1}^{N} p_i \cdot V_i$. All documents have been prefetched after $T_{w_0} = \sum_{i=1}^{N} \frac{{}^kV_i}{{}^kC_i}$. Hence the expected value of transported volume $E\{V \mid t_R = T_{w_0}\}$ is $\sum_{i=1}^{N} V_i$. The retrieval phase of the i-th document corresponds to the i-th linear segment of the curve.*

case 2) the document is currently being retrieved. Subtracting the time invested on the other documents from the total time available for prefetching and multiplying the result with the data rate ${}^kC_i$ for the document, yields the transported volume of the document.

case 3) no attempt has been made to retrieve the document before t_R. In this case no volume has been transported. Hence the minimum amount of transported volume is zero.

The three cases are distinguished by the min/max operation. Unlike in eq. 2.74, the expected value of the increase in transported volume is calculated by weighting with the probabilities $1 - {}^kp_i$ that the document will *not* be selected, since the document's volume ${}^kV_i$ will be transported but without being used with probability $1 - {}^kp_i$. Fig. 2.39 illustrates the increase in transported volume for an arbitrary permutation.

Analogously to eq. 2.75, the unconditional expected transported volume $E\{V\}$ is computed by

$$E\{V\} = \int_0^\infty f_{t_R}(t_R) \cdot E\left\{{}^kV \mid t_R\right\} dt_R. \tag{2.80}$$

Since we have now answered our fourth question and have gained an understanding of the increase in transported volume, we proceed with the discussion of the remaining question, whether we should refrain from prefetching very unlikely documents.

2.2.4 User Policy and Optimum Probability Threshold

The previously performed analysis has shown us that the decision to prefetch a particular document D_i results in a decrease of expected waiting time $E\{T_w\}$ and an increase in the expected transported volume $E\{V\}$.

$$\Delta E\left\{{}^{k}T_{w_i}\right\} < 0 \tag{2.81}$$

$$\Delta E\left\{{}^{k}V_i\right\} > 0 \tag{2.82}$$

We believe that the assumption of an individual user policy and a general network operator policy is necessary to weight and balance these beneficial and adverse effects against each other. Most of today's operator policies and a reasonable user policy are well approximated by linear functions of the transported volume, the duration of usage and waiting time [AK02].

In a very simple form, a user policy assigns a cost factor $\kappa_{U,T}$, with unit 1/s, to weight the waiting time of the user.

A simple operator policy may assign a cost factor $\kappa_{N,T}$ to weight the temporal use of the network resource, with unit 1/s, and a cost factor $\kappa_{N,V}$, with unit 1/byte, to weight the amount of volume transported over the network.

If a user policy and an operator policy, such as described is assumed, we can assign incremental costs Δc_i to a decision to prefetch document D_i. For each possible decision the costs can be negative, i.e. $\Delta c_i < 0$ (then we should perform the prefetching) or positive, i.e. $\Delta c_i > 0$ (then we should decide against prefetching).

Since for the decision process we only have to consider the incremental costs, we need to compute the increment in duration of usage which is equivalent with the increment in waiting time. Therefore we state the incremental cost under the assumption of a linear user and operator policy:

$$\Delta c_i = \kappa_{N,V} \cdot \Delta E\left\{{}^{k}V_i\right\} + (\kappa_{N,T} + \kappa_{U,T}) \cdot \Delta E\left\{{}^{k}T_{w_i}\right\}. \tag{2.83}$$

Hence, a negative Δc_i requires the following condition:

$$\kappa_{N,V} \cdot (1 - p_i) \cdot V_i + (\kappa_{N,T} + \kappa_{U,T}) \cdot (-1) \cdot p_i \cdot \frac{V_i}{C_i} < 0, \tag{2.84}$$

from which follows

$$p_i > \frac{1}{1 + \dfrac{\kappa_{N,T} + \kappa_{U,T}}{\kappa_{N,V}} \cdot \dfrac{1}{C_i}}. \tag{2.85}$$

Therefore, we are now able to define a threshold probability p_{th}

$$p_{th} = \frac{1}{1 + \frac{\kappa_{N,T} + \kappa_{U,T}}{\kappa_{N,V}} \cdot \frac{1}{C_i}}. \tag{2.86}$$

This prefetching threshold can now be used to determine whether a particular document should be prefetched or not. Only if the i-th document's probability p_i exceeds the probability threshold p_{th}, the i-th document is prefetched in order to maximize the perceived performance given a user policy and an operator policy. Hence, the prefetching threshold directly controls the amount of prefetching activity. It can be adjusted within a continuous interval:

$$p_{th} \in [0, 1] \tag{2.87}$$

The two extreme values $p_{th} = 0$ and $p_{th} = 1$, lead to either complete activation or deactivation of prefetching activity. If the prefetching threshold p_{th} is adjusted to zero, unlimited prefetching will be performed. In contrast, setting the prefetching threshold p_{th} to one, results in a complete suppression of prefetching activity.

We have now gained a considerable understanding of the fundamental properties of situation aware prefetching and the parameters that influence its performance. The performed analysis has answered our initial questions.

We have found that the documents' probability is the sole criterion to be applied for determining the sequence of the speculative retrievals. The size of the documents is no sensible criterion for this decision. An equation has been derived that allows us to determine the quantitative influence of the documents' probabilities on the achievable performance. Furthermore, an equation to compute the always present increase of transported volume whenever prefetching is applied has been presented. Finally, the probability threshold, derived above, answers the fifth question regarding a criterion to decide when to refrain from prefetching. Again, we emphasize the necessity to assume a user policy and a network operator policy to define an optimum threshold for the prefetching decision.

The performed analysis has been carried out under the assumption of a constantly available network resource. Common sense leads us to conjecture that situation aware prefetching has the potential to be particularly beneficial in scenarios with temporal varying network conditions. Such conditions typically occur in heterogeneous wireless access network with different radio interfaces and only partial coverage.
Hence, our next step is to continue our investigation by performing a system level simulation of situation aware prefetching that is capable of accurately modelling the relevant conditions of these scenarios in order to provide insights to the performance of prefetching under these conditions. The theoretical foundations we have built in our previous analysis will prove their value in the interpretation and understanding of the achieved simulation results.

Chapter 3

System Simulation

In the previous chapter we have performed an analytical investigation of topics that arise when mobile information access is improved by situation awareness. In this chapter we present the methods and results of our simulations, which were performed for the purpose of extending the previously obtained analytical results towards a system level perspective and more complex scenarios.

Since we are especially interested in the effects of situation aware prefetching for mobile information access in heterogeneous wireless networks, a **network model** (3.1) is required in which **user mobility** results in **network topology changes** (3.1.1). Furthermore, **resource sharing** among multiple users (3.1.2) has to be considered. For the purpose of generating realistic user mobility a novel **mobility model** is presented (3.2), which comprises **path generation** (3.2.1), **speed generation** (3.2.2) and the definition of **coverage areas** (3.2.3). A **model for hypertext documents and traffic** based on Mah's empirical distributions is briefly described (3.3) as a further building block of a realistic simulation environment. Finally the obtained **simulation results** are presented and discussed (3.4) for a **single user scenario** (3.4.1) and various **multi-user scenarios** (3.4.2).

3.1 Network Model

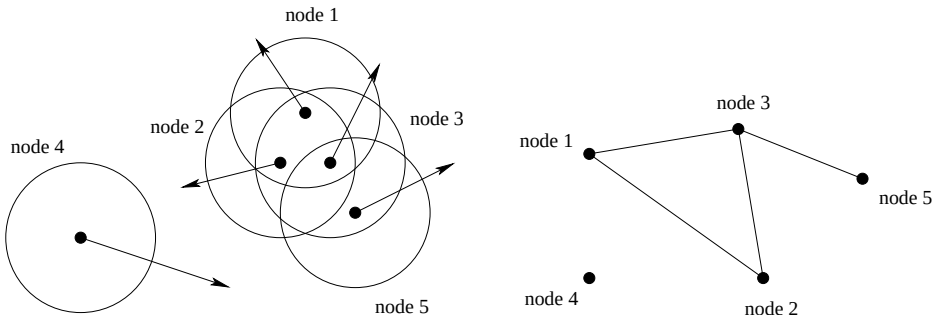
Since we conjecture that situation aware prefetching will be particularly beneficial in mobile scenarios with heterogeneous wireless access technologies we will start by defining an appropriate network model for our simulations. The mobility of the user in combination with finite coverage regions constantly causes changes in the network topology. Typically the network topologies changes too frequently to burden the user with the necessary reconfiguration. Therefore we require the system to perform automatic topology establishment and dynamic topology maintenance.

3.1.1 Topology and Mobility

A definition of the network topology helps us to describe the scenarios under investigation within this work. We use the abstraction of *adjacency* between nodes. Two nodes are adjacent if they can establish connections and communicate in both direction with each other[1]. With this definition we can visualize the topology at a given time instant t by an un-directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where all nodes form the set of *vertices* $\mathcal{V}$ and the available links form the set of *edges* $\mathcal{E}$. The state of the topology can be condensed into the symmetric *adjacency matrix* $\mathbf{A}(t) = [a_{ij}(t)] = [a_{ji}(t)]$ of the graph $\mathcal{G}$, where $a_{ij}(t) = a_{ji}(t) = 1$ if node i and node j can communicate with each other at time instant t and $a_{ij}(t) = a_{ij}(t) = 0$ if not.

An example is given in Fig. 3.1. The corresponding adjacency matrix for the depicted time instant is

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$



(a) Nodes and coverage regions (b) Topology graph $\mathcal{G}$

Figure 3.1: *Dynamic network topology of ad hoc nodes. Node mobility (speed, direction) is indicated by arrows. At each time instant the nodes' current location in conjunction with their communication range leads to a particular topology*

Typical assumptions in ad hoc networking are the lack of any fixed infrastructure, the equality of all participating nodes in terms of mobility and communication technologies and the ability of mobile nodes to directly connect with each other. While in this work we consider frequently changing ad hoc connections between nodes we do

[1]Depending on the wireless technology communication may still be possible in one direction but not in the other, due to asymmetries e.g. transmission power on mobile and base station. However, in this work, we are considering only the case where connection-oriented protocols with ARQ are employed. With these protocols a link has to be considered broken, whenever communication in one or both directions is impossible

not treat ad hoc network systems that follow the previously stated typical assumptions. Instead a combination of a globally available mobile network with only locally available short range access points and a hierarchical network structure is investigated. The set of vertices in the graph $\mathcal{G}$ can then be partitioned into four subsets $\mathcal{V}_m, \mathcal{V}_r, \mathcal{V}_c$ and $\mathcal{V}_s$ representing mobile nodes (mobile devices), resident nodes (access points), central nodes and server nodes. The number of elements in the four sets is denoted by $N_m = |\mathcal{V}_m|,\ N_r = |\mathcal{V}_r|,\ N_c = |\mathcal{V}_c|$ and $N_s = |\mathcal{V}_s|$.

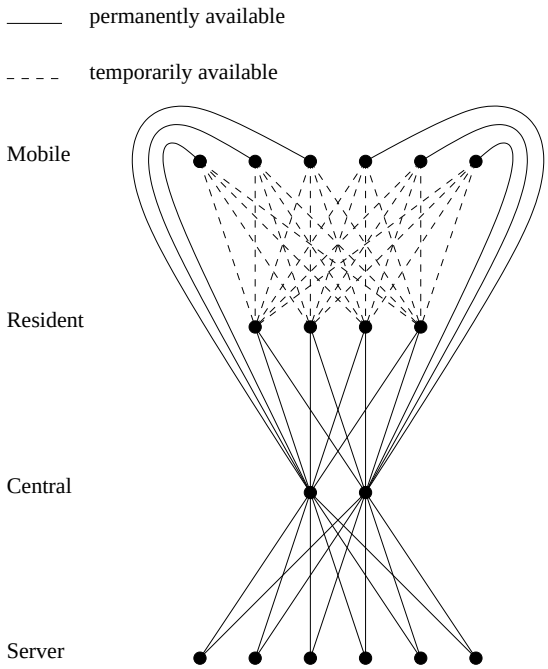


Figure 3.2: *Hybrid hierarchical network topology*

Given this partition and the structure depicted in Fig. 3.2 the adjacency matrix $\mathbf{A}(t)$ can be viewed as being composed of several sub-matrices with each sub-matrix describing the connection between the subsets of nodes or vertices:

$$\mathbf{A}(t) = \begin{pmatrix} \mathbf{A}_{M,M} & \mathbf{A}_{M,R}(t) & \mathbf{A}_{M,C} & \mathbf{A}_{M,S} \\ \mathbf{A}_{R,M}(t) & \mathbf{A}_{R,R} & \mathbf{A}_{R,C} & \mathbf{A}_{R,S} \\ \mathbf{A}_{C,M} & \mathbf{A}_{C,R} & \mathbf{A}_{C,C} & \mathbf{A}_{C,S} \\ \mathbf{A}_{S,M} & \mathbf{A}_{S,R} & \mathbf{A}_{S,C} & \mathbf{A}_{S,S} \end{pmatrix} \tag{3.1}$$

Since all other connections are either permanently available or permanently unavailable, only the connections between mobile nodes and resident nodes result in time dependent sub-matrices $\mathbf{A}_{M,R}(t)$ and $\mathbf{A}_{R,M} = \mathbf{A}_{M,R}^T(t)$. If we denote the all-one and all-zero matrix of dimension $n \times m$ by $\mathbf{1}_{m,n}$ and $\mathbf{0}_{m,n}$ and the identity matrix of dimension $n \times n$ by $\mathbf{I}_{n,n}$ and use the symmetry of the links we can see that for the hybrid hierarchical network topology

$$\mathbf{A}(t) = \begin{pmatrix} \mathbf{I}_{N_m,N_m} & \mathbf{A}_{M,R}(t) & \mathbf{1}_{N_m,N_c} & \mathbf{0}_{N_m,N_s} \\ \mathbf{A}_{M,R}^T(t) & \mathbf{I}_{N_r,N_r} & \mathbf{1}_{N_r,N_c} & \mathbf{0}_{N_r,N_s} \\ \mathbf{1}_{N_c,N_m} & \mathbf{1}_{N_c,N_r} & \mathbf{I}_{N_c,N_c} & \mathbf{1}_{N_c,N_s} \\ \mathbf{0}_{N_s,N_m} & \mathbf{0}_{N_s,N_r} & \mathbf{1}_{N_s,N_c} & \mathbf{I}_{N_s,N_s} \end{pmatrix}. \tag{3.2}$$

For all scenarios that do only employ a mobile network with global coverage, hidden handover and roaming the adjacency matrix differs from the one in eq. 3.2 since the coefficients of the sub-matrix $\mathbf{A}_{M,R}(t)$ become zero for all time instants.

$$\mathbf{A}_{M,R} = \mathbf{A}_{R,M}^T = \mathbf{0}_{N_m,N_r} \tag{3.3}$$

This scenario will be analyzed and simulated for reference purposes and will be termed *classic mobile networking scenario*. Concerning the aspects investigated in this work, the mobility of the user has no influence for this scenario.

The extreme opposite is termed *access with coverage gaps scenario* and will be analyzed and simulated for reference purposes as well. For this scenario only access via short-range communication is allowed and no communication via the mobile network is possible. Therefore $\mathbf{A}(t)$ differs from eq. 3.2 in the form that all coefficients of the sub-matrix $\mathbf{A}_{M,C}(t)$ become zero for all time instants.

$$\mathbf{A}_{M,C} = \mathbf{A}_{C,M}^T = \mathbf{0}_{N_m,N_c} \tag{3.4}$$

In the following we will show that our proposed application of situation information to prefetching will be most effective in the *hybrid networking scenario* where local access via short-range communication is combined with global access via a mobile network. We will use simulations to compare this scenario with the non-hybrid scenarios. For this scenario eq. 3.2 describes the adjacency matrix $\mathbf{A}(t)$.

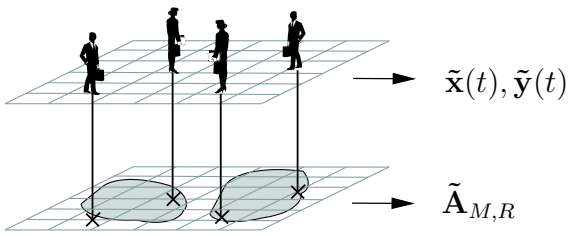


Figure 3.3: *Measured location and connectivity of real world process.*

In the real world the coefficients of the sub-matrix $\mathbf{A}_{M,R}(t)$ are determined by the current location of each mobile node and the coverage areas of the resident nodes. The location of a mobile node is directly coupled with its associated user's location. The user's location in turn depends on the free will of the human being, determining his

desired place or path, which is assumed not to be known by the system. It is therefore reasonable to treat the user's movements as a random process and the mobile node's position as a random variable. As a consequence the coefficients of the sub-matrix $\mathbf{A}_{M,R}(t)$ also become random variables. For further analysis we would like to know the relevant statistical properties of these random variables. Ideally we would be able to obtain and store measured position vectors $\tilde{\mathbf{x}}(t), \tilde{\mathbf{y}}(t)$ and measured sub-matrices $\tilde{\mathbf{A}}_{M,R}(t)$. We could then use the stored sub-matrices $\tilde{\mathbf{A}}_{M,R}(t)$ as direct input to our simulations. For investigating the influence of the coverage regions we would then also be able to use maps representing hypothetical coverage regions in combination with the stored position vectors $\tilde{\mathbf{x}}(t), \tilde{\mathbf{y}}(t)$ in order to generate synthetic sub-matrices $\mathbf{A}_{M,R}(t)$. While the results of ongoing research will yield this kind of data at some time, they are not available at the time being.
Various kinds of mobility models are available as substitutes that are capable of generating plausible position vectors. These models can be parameterized to investigate the influence of mobility under many conditions. A novel mobility model that fulfills our requirements is presented in Section 3.2.

3.1.2 Resource Sharing and Effective Data Rate

In any multi-user communications system the population of users competes for the use of the communication channel [Kle76]. Various medium access control (MAC) schemes are known with different levels of efficiency and complexity, ranging from simple ALOHA to more advanced CSMA (Carrier Sense Multiple Access) protocols. While most wireless LAN standards use CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) in which collisions are possible, today's mobile networks with standards such as GSM-(HS)CSD, GPRS or UMTS use managed TDMA and CDMA schemes without collisions. A typical representative of a wireless PAN standard (Personal Area Network) is Bluetooth. In this standard a master device manages and assigns the radio resource to up to seven other devices, so-called slaves. Master and slaves form a so-called piconet. In conjunction with the statistics of the transported traffic and, of course, the current radio channel conditions, these different MAC schemes strongly influence the effective data rate. Depending on the standard the effective data rate the application experiences is further influenced by many parameters such as the employed radio link protocol (RLP) or the particular implementation of the transport control protocol (TCP).

The vast amount of free parameters of the lower protocol layers of these systems would make an exhaustive exploration of their influence on prefetching performance very cumbersome. Additionally, the computational complexity of simulating hundreds of protocol stacks in concurrently would result in unduly long simulation times. Hence, we have decided to choose a higher degree of abstraction to assign effective data rates within our simulations. Since the implementation of the software components for the fixed (resident proxy, central proxy) and mobile network nodes (mobile proxy) has enabled us to perform measurements on an existing infrastructure we are able

to use the measured effective data rates at the application layer to parameterize our simulations and obtain realistic results.

Performed measurements in a Bluetooth piconet have prompted us to model the short range link in a straightforward way. The maximum effective data rate that a single user would enjoy is shared among all the users in the coverage area of one short range access point. The sharing is fair in a sense that all non-empty user queues are serviced at the same rate. We do not give preferential treatment to short requests or responses. Separate queues exist for uplink and downlink.

The network model used in our simulation assumes the mobile networks to be over-provisioned with the end-devices acting as bottlenecks. Hence, effective data rates are independent of the number of users in the mobile network.

Both mobile networks and short range networks are introducing fixed delays on the uplink and downlink.

3.2 Mobility Model

Latest algorithms for call admission control, handover, prefetching etc. show a clear trend towards exploiting higher order statistics of the underlying statistical processes. In order to simulate the performance of these algorithms, the models used for simulation have to keep up with this trend and become statistically "richer", i.e. they should capture higher-order statistics of user behavior than the models which are available today.

A thorough overview of previous mobility models ranging from the simple Brownian motion model via Markovian models that govern user acceleration, speed and direction to location-trace based models can be found in [Bet01].

We have developed a novel model in order to reduce the shortcomings of previous models in capturing higher-order statistical dependencies in user behavior [AKL03]. The proposed model strikes a balance between the abstraction level of some purely stochastic path generation models that lack certain important properties of the real world and completely deterministic models that fail to produce answers to inherently stochastic problems. The new model uses freely definable layout information and generates sensible paths between randomly selected way-points. Paths of user movement are generated by a combined diffusion/steepest gradient algorithm. In order to completely define the dynamics of the user's movement, speed values are generated at sampling instances[2].

[2]It is important to keep in mind that our model is not intended to be incorporated into the algorithms for call admission control, handover or prefetching, since the necessary knowledge (building layout, radio coverage, typical way-points) is usually not assumed to be available for these algorithms. Instead these algorithms use and adapt or learn much simpler models. Our model is typically used for performance evaluation in simulations where it takes the role of the real world model that creates the true behavior which generates the input for these algorithms.

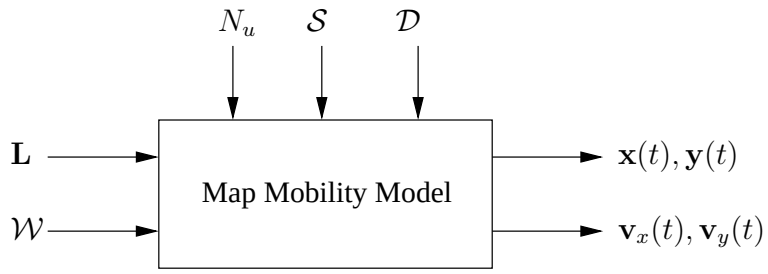


Figure 3.4: *Inputs (layout map matrix* $\mathbf{L}$*, set of waypoints* $\mathcal{W}$*, number of users* N_u*, parameter tuple of speed model* $\mathcal{S}$ *and parameter tuple of dwell time model* $\mathcal{D}$*) and outputs (location* $(\mathbf{x}(t), \mathbf{y}(t))$ *and velocity* $(\mathbf{v}_x(t), \mathbf{v}_y(t))$*) of the mobility model.*

3.2.1 Path Generation

The motion of a mobile node (user) is confined to a rectangular area which is spatially sampled into a *layout map matrix* $\mathbf{L}$ of dimension $N_x \times N_y$ that defines the accessible and inaccessible areas for users with

$$l_{i,j} = \begin{cases} 1 & \text{if} \quad x = i, y = j \quad \text{is} \quad \text{accessible} \\ 0 & \text{if} \quad x = i, y = j \quad \text{is} \quad \text{not accessible} \end{cases} \tag{3.5}$$

A finite set $\mathcal{W}$ of $N_{\mathcal{W}}$ *waypoints* $\{(x_1, y_1) \ldots (x_{N_{\mathcal{W}}}, y_{N_{\mathcal{W}}})\}$ is specified. The user motion will take place on "sensible" paths between these waypoints. A path finding algorithm is employed to find possible paths between any two of these waypoints, avoiding obstacles such as walls or building corners.

3.2.1.1 Diffusion Algorithm

For the computation of sensible paths between two waypoints, we employ an algorithm that has initially been applied for the task of path finding for autonomously moving robots [SA93]. The algorithm is inspired from the analysis of gas diffusion, studied in thermodynamics. The basic idea is to model the target-waypoint as a source of continuously effusing gas that disperses in free space. The gas shall be absorbed by walls and other obstacles. In the vicinity of obstacles this simulated absorption leads to gradient pointing away from the obstacles. After a sufficiently long duration of time the gas concentration at each point within the observed area will come close to the concentration it would assume in an equilibrium state. Now, for any start-point, from which a path exists towards the target-waypoint, a path can be found by computing and following the gradient of the concentration. This path is usually not a shortest possible path, but a remarkable compromise between shortness and avoiding obstacles or narrow passages[3]. The computation of the concentration is straightforward:

[3]In contrast, other path finding algorithms, such as Lee's algorithm, find a shortest path [Lee61]. Plotting these shortest paths in, e.g. a building map, we see that they do not resemble the movements of real human beings well, since they tend to "cut the corners" in a very unnatural way.

For all $N_{\mathcal{W}}$ waypoints $W_i = (x_i, y_i), i = 1 \ldots N$, we have to compute a diffusion matrix $\mathbf{D}_i$ of dimension $N_x \times N_y$. For this purpose we define a filter matrix $\mathbf{F}$ of dimension $N_{\mathbf{F}} \times N_{\mathbf{F}}$, $N_{\mathbf{F}} \in \{3, 5, \ldots\}$[4], with elements

$$f_{p,q} = 1/N_{\mathbf{F}}^2 \quad \forall\, p, q: \quad p, q = 1, \ldots, N_{\mathbf{F}}. \tag{3.6}$$

We start with initializing the diffusion matrix $\mathbf{D}_i$:

$$d_{i,u,v}(0) := \begin{cases} 1 & \text{where} \quad u = x_i, v = y_i \\ 0 & \text{elsewhere} \end{cases} \tag{3.7}$$

The diffusion is now computed by repetitively convoluting the diffusion matrix $\mathbf{D}_i$ with the filter matrix $\mathbf{F}$ and element-wise multiplication by the layout map matrix $\mathbf{L}$. For each element $d_{i,u,v}(k+1)$ we obtain after the $(k+1)$-th repetition

$$d_{i,u,v}(k+1) = l_{u,v} \cdot \sum_{p=1}^{N_{\mathbf{F}}} \sum_{q=1}^{N_{\mathbf{F}}} d_{i,u+p-(N_{\mathbf{F}}+1)/2,v+q-(N_{\mathbf{F}}+1)/2}(k) \cdot f_{p,q}, \tag{3.8}$$

and constantly refreshing the source after each step, by setting

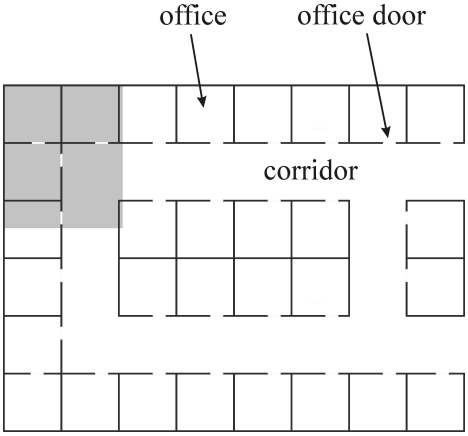
$$d_{i,x_i,y_i}(k+1) := 1 \tag{3.9}$$

until at least

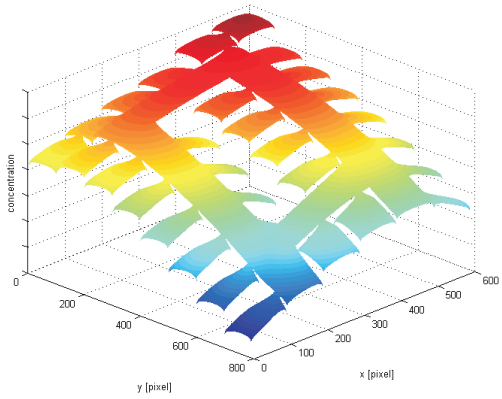
$$d_{i,u,v} + l_{u,v} > 0 \quad \forall\, u, v: \quad u = 1, \ldots, N_x, v = 1, \ldots, N_y. \tag{3.10}$$

Condition 3.10 ensures that the gas has reached all accessible locations and a path can be generated by following the steepest gradient. Figures 3.5 and 3.6 show examples of the obtained results for an office floorplan and an urban layout.
It is beneficial to perform the computation of the diffusion map for each given target-waypoint only once and to store the diffusion matrices.

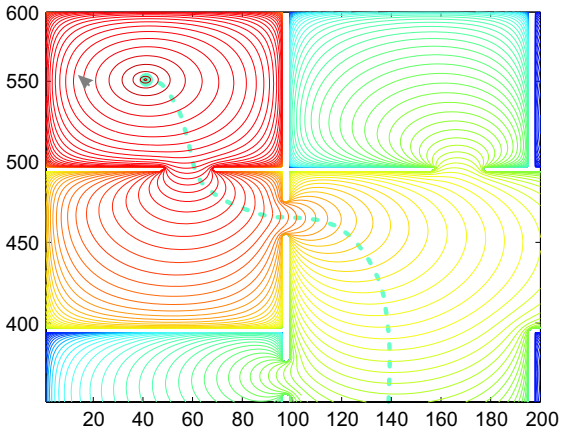
[4]Typically, we use $N_{\mathbf{F}} = 3$ in order to avoid unwanted "leaking" of gas through thin walls (thickness of 1 pixel). Depending on the minimum extension of obstacles, larger values for $N_{\mathbf{F}}$ may be chosen. If for example the minimal thickness of walls in a building layout is 2 pixels, $N_{\mathbf{F}} = 5$ is permissable.



(a) Floorplan for office scenario



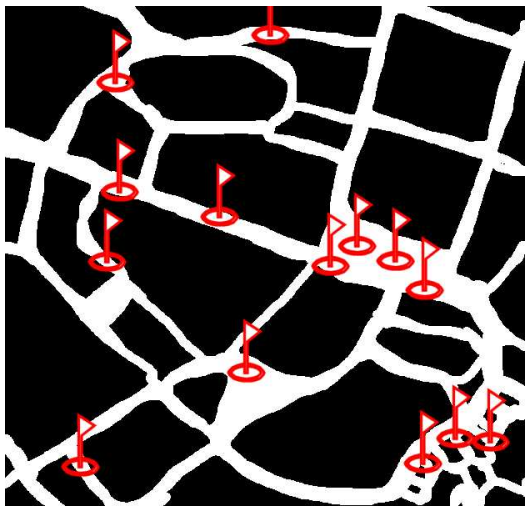
(b) Concentration after diffusion



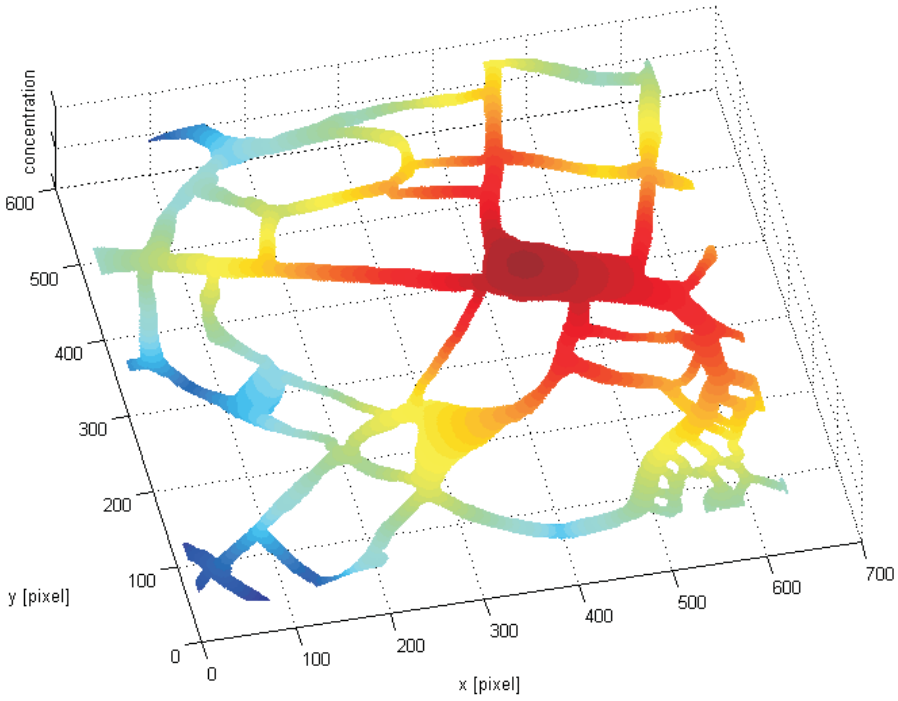
(c) Contour plot of concentration after diffusion for waypoint in top-left room.

Figure 3.5: *A path towards the target-waypoint (40, 200) can be computed from any location by following the steepest gradient, i.e. crossing the contour lines perpendicularly. A sample path is added for illustration purposes (the section depicted in subfigure 3.5(c) corresponds to the gray-shaded area in subfigure 3.5(a) and to the range 350–600 (x-pixel) and 0–200 (y-pixel) in subfigure 3.5(b)).*

(a) High-resolution satellite image of urban area



(b) Accessability matrix defining accessible/ inaccessible locations for simulation. (waypoints are marked with flags.)



(c) Result of simulated diffusion process.

Figure 3.6: *Example of intermediate process results of mobility model for an urban scenario. Starting with a real-world reference (floor plan or, like here a satellite image) an accessability matrix is derived. Waypoints are added manually to arbitrary accessible locations. For each waypoint a diffusion process is computed, where the particular waypoint acts as source of the substance, resulting in a simulated concentration of the substance.*

3.2.2 Speed Generation

We have seen how realistic paths can be generated. For the definition of movement along these paths, it is now necessary to assign a speed to every user for every sampling instant. A number of well-known speed models are available for this purpose [Bet01]. They range from the simplest – constant speed – model to more advanced models with deterministic or stochastic acceleration[5]. A tuple $\mathcal{S}$ summarizes the parameters of the chosen speed model. The speed model has to be chosen carefully to properly represent the dominant real world conditions and statistical properties of the particular scenario under investigation, since the model has significant effects on the statistical properties of the simulated motion.

We have to be particularly aware of the fact that the finite set $\mathcal{W}$ of $N_{\mathcal{W}}$ waypoints in conjunction with the deterministic path generation algorithm results in $N_{\mathcal{W}} \cdot (N_{\mathcal{W}} - 1)$ distinct paths (if both directions are counted), which may result in unwanted artifacts in certain statistics of interest. Fig. 3.7 shows an example that illustrates how this is reflected in a probability density function $f_{\mathbf{T}_c}(\mathbf{T}_c = T_c)$ for the duration of contact T_c of a given coverage region with mobile nodes.

Such discrete peaks might be exploited e.g. by an adaptive handover/hand-off algorithm, resulting in too optimistic results. This means the results obtained in a simulation would be better than what is achievable under real world conditions.

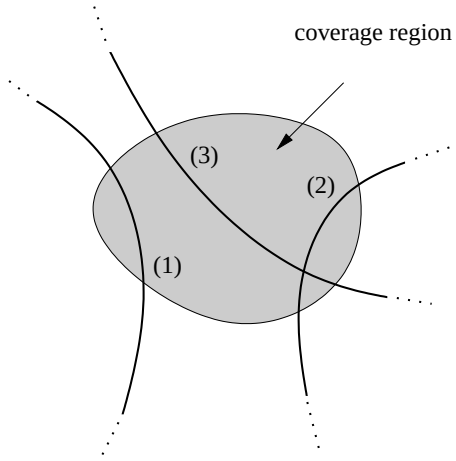
In order to avoid the described effect, we combine our path generating algorithm with the speed model proposed by Bettstetter in [Bet01]. This speed model employs random processes each for

- generating the duration between speed change events,
- choosing the next target speed,
- choosing the acceleration when speed changes occur.

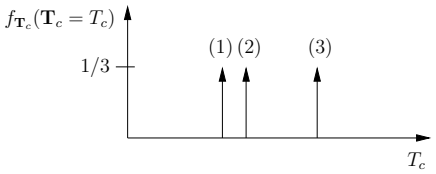
The duration between speed changes is drawn from an exponential distribution with a mean duration μ_{v^*} (e.g. 25 seconds). The target speed v is drawn from a probability density function $f_{\mathbf{v}}(\mathbf{v} = v)$, representing N_v preferred speeds $v_{\text{pref},0} \ldots v_{\text{pref},N_v-1}$ and a floor of speeds, which are uniformly distributed over the interval $[v_{\min}, v_{\max}]$. Bettstetter gives example parameters for a "downtown car" scenario. For our purposes, we have defined a new set of parameters in order to model an urban pedestrian scenario[6]:

[5] Related to the speed models are the models for the *dwell time*, which is defined as the duration a mobile node spends after reaching a waypoint before starting towards the next waypoint. Currently we lack substantial experimental evidence on how dwell times are distributed in reality. We consider the derivation of good models for the dwell times to be still an open issue. Some speed models generate time intervals with zero velocity. Currently we use one of these models as a substitute for a separate dwell time model. Nevertheless we consider it worthwhile to formulate separate models for dwell times as soon as experimental data becomes available.

[6] This parameter set is derived from empirical data for pedestrian movement which has been studied with the aim of optimizing e.g. signal periods of traffic lights or management of pedestrian flows [KPN96, BFA01].



(a) Coverage region with three possible traversing paths



(b) Resulting probability density function $f_{\mathbf{T}_c}(\mathbf{T}_c = T_c)$ for the duration of contact T_c of a given coverage region with mobile nodes

Figure 3.7: *Repercussions of deterministic path generation in combination with a constant speed model, leading to (unrealistic) discrete peaks, which might result in too optimistic results.*

The interval of possible speeds is limited by $v_{\min} = 0.0$ m/s and $v_{\max} = 2.5$ m/s. We use two preferred speeds $v_{\text{pref},0} = v_{\min} = 0.0$ m/s and $v_{\text{pref},1} = 1.5$ m/s, with probabilities $\Pr\{\mathbf{v} = v_{\text{pref},0}\} = 0.2$ and $\Pr\{\mathbf{v} = v_{\text{pref},1}\} = 0.7$. With probability 0.1 none of the two preferred speeds is chosen (see Fig. 3.8). We use the target speed of 0.0 m/s as substitutes for dwell times. The resulting movement appears natural to visual inspection and generates sensible statistics.

A software tool that uses the described method of path generation and various speed models for generating traces of user mobility has been implemented in JAVA. The software reads layout information from standard graphics files which can be generated by any image processing software. A graphical user interface allows to adjust parameters and to visualize the movements of simulated users. The presented mobility model has proved to be straightforward to implement. Despite its simplicity, the diffusion algorithm generates paths that imitate real world paths of mobile users well. The model can be used to investigate a multitude of mobile wireless scenarios by using appropriate layout maps and adjusting the parameters of the speed models with very little effort.

3.2.3 Coverage Model Definition

The link between the position vectors $x(t), y(t)$ of the simulated user mobility and the adjacency matrix $\mathbf{A}(t)$, representing the actual network topology, is established by the definition of coverage regions. As we have pointed out earlier (see eq. 3.1 in

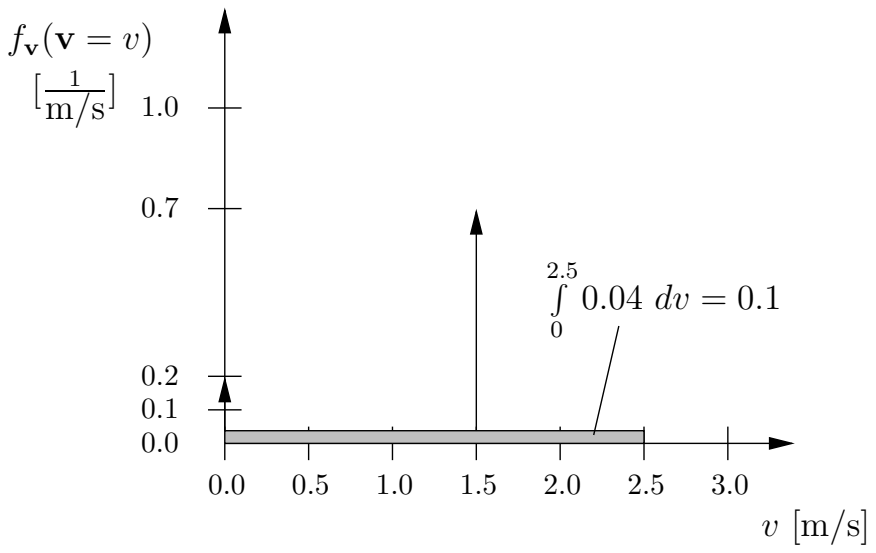


Figure 3.8: *Probability density function for generating target speeds. The two preferred speeds of 0.0 m/s and 1.5 m/s are chosen with probabilities 0.2 and 0.7 respectively. With a probability of 0.1 any other speed of the interval [0.0 m/s ... 2.5 m/s] is chosen.*

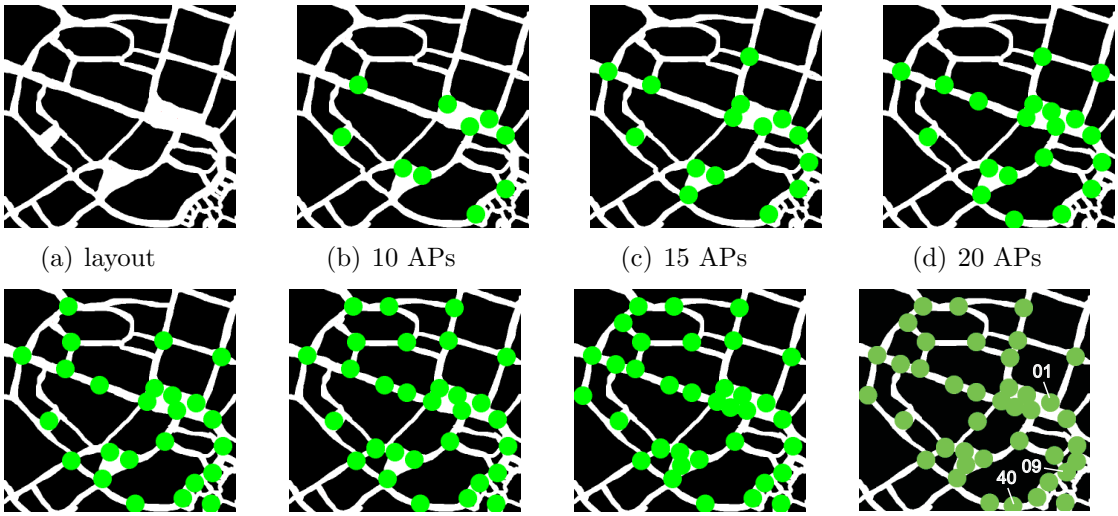
Section 3.1.1), we consider only the link availability between the resident nodes (access points) and the mobile nodes to be time-variant. For each access point k, coverage is defined by the element $c_{i,j,k}$ of the k-th *coverage region matrix* $\mathbf{C}_k$ of dimension $N_x \times N_y$.

$$c_{i,j,k} = \begin{cases} 1 & \text{if} \quad x = i, y = j \quad \text{is} \quad \text{covered by access point } k \\ 0 & \text{if} \quad x = i, y = j \quad \text{is} \quad \text{not covered by access point } k \end{cases} \tag{3.11}$$

For later quantitative comparison between the influence of speculative prefetching and various levels of access point deployment seven levels of access point deployment are specified for an urban area (see Fig. 3.9). The access points are arbitrarily positioned without any particular optimization algorithm. An apparent dichotomy exists for the goals a) to minimize the maximum duration of stretches the users have to spend without coverage; and b) to maximize the amount of time spent within coverage. The manual placement of the access points in our example is an attempt to equalize between the various optimization criteria[7].

Since the movements of the various mobile nodes are generated by independent random processes, the number of users located within the coverage region of a given access point at a sampling instant is time-variant. This effect is demonstrated in Fig. 3.10 for a simulated population of 1000 pedestrian users in the urban scenario.

[7]We wish to emphasize our opinion that optimum positioning of access points is a complex, interesting and potentially fruitful research problem. However, it is beyond the scope of this thesis and will be a topic of future investigation.



(a) layout (b) 10 APs (c) 15 APs (d) 20 APs

(e) 25 APs (f) 30 APs (g) 35 APs (h) 40 APs

Figure 3.9: *Coverage areas for increasing levels of access point deployment are superimposed on an urban layout. Seven levels of access-point (AP) deployment are illustrated in subfigures (a) to (h).*

We also see that the popularity of access points depends on their location in the layout[8]. In this example, the number of users in range of an access point located near the city center (AP09) fluctuates between 30 to 40, whereas an access point located in a more remote position (AP40) is sometimes unoccupied and usually not occupied by more than 10 users.

The number of users in range of an access point is itself a random variable. Interestingly we can see in Fig. 3.11 how this distribution is closely approximated by a Poisson distribution.

[8]If our mobility model is applied, the popularity of an access point is proportional to the sum of length of the path segments traversing through its coverage region.

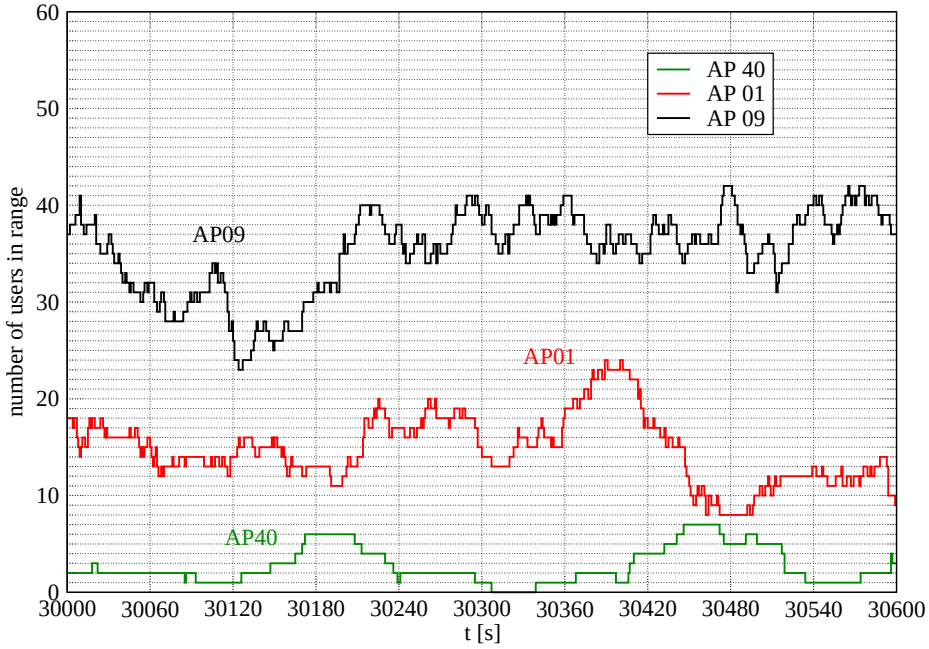


Figure 3.10: *The number of users within the coverage region of three distinct access points show strong temporal fluctuation. For the chosen parameters (1000 mobile nodes (pedestrian), 40 access points, urban layout) the number of users within the coverage of a popular access point (AP09) varies between 24 and 42 for the depicted time segment. A less popular access point (AP40) has a minimum of zero and a maximum of 7 users within range.*

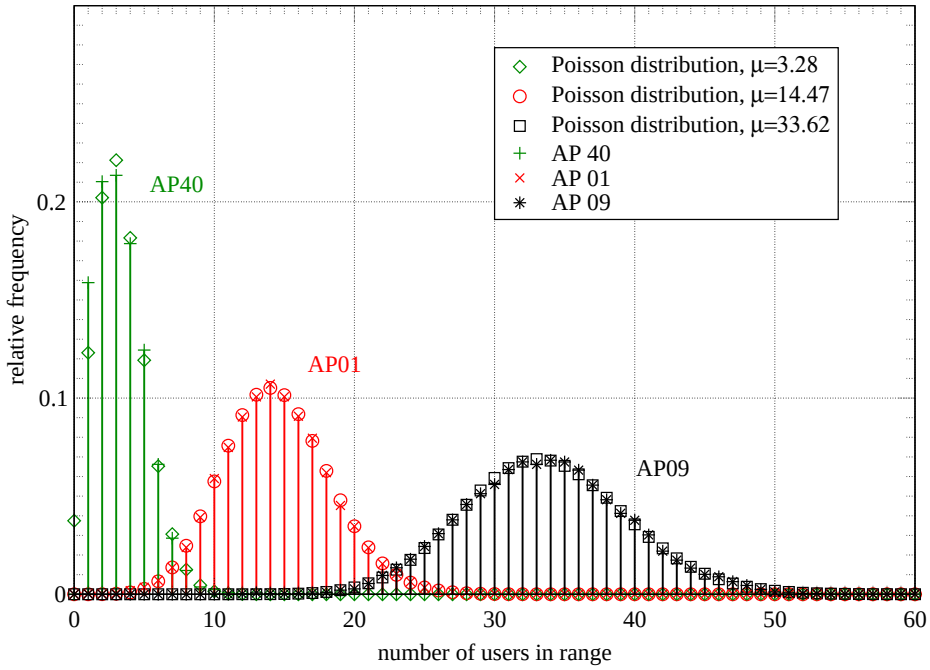


Figure 3.11: *Relative frequency of users within the coverage region of three distinct access points. Interestingly, the relative frequency is closely approximated by a Poisson distribution (1000 mobile nodes (pedestrian), 40 access points, urban layout).*

3.3 Document and Traffic Model

For our theoretical analysis presented in Section 2.2 we used the abstraction level of *documents* as the objects of the retrieval process in a hypermedia system. We chose this abstraction for reducing model complexity and achieving good analytical tractability and clarity of interpretation.

However, some effects that occur in reality might remain unobserved, since they are caused by the internal document structure of a hypermedia system. Fig. 3.12 provides a more detailed illustration of a document's internal structure (the more abstract document model used in Section 2.2 is repeated for comparison).

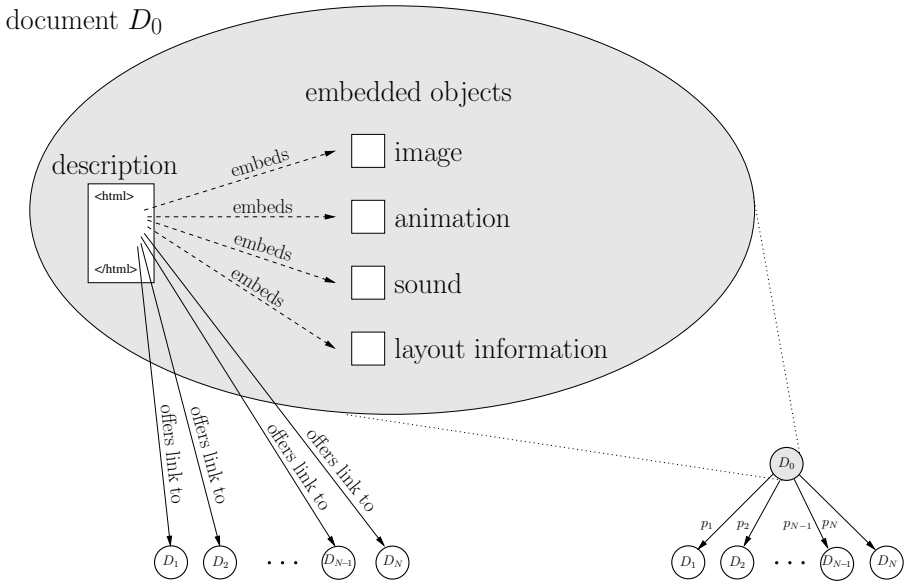


Figure 3.12: *The internal structure of a hypermedia document is put in comparison with the more abstract model (in the lower right corner). The description (e.g. HTML) of the document contains references to embedded objects, e.g. images, animations, sounds or layout information. Further documents are offered to the user via "links", which are also contained in the description.*

We see that a document is composed of a description, typically using some markup language e.g. HTML, XHTML or WML, and a number of embedded objects, e.g. images, animations, sounds or layout information which are referenced from said description. Furthermore, the description contains references ("links") to other documents.

Whenever a user selects a link pointing to another document he starts to wait for the presentation of the complete document. Therefore, the client application sends a request for the particular document's description to the server. The server reacts by sending a response containing the description to the client application. The request and response concerning the retrieval of the description are termed *primary request* and *primary response* respectively.

Upon delivery and parsing of the primary response for references to embedded objects the client applications starts to issue requests for the embedded objects. Again the server reacts by sending the appropriate responses. The requests and responses concerning the retrieval of the embedded objects are termed *secondary requests* and *secondary responses* respectively. After all secondary responses have arrived at the client, the client presents the complete document or "page impression" to the user. At this moment the *waiting time* of the user is considered to be finished and the *viewing time* starts. When the user selects and requests the next document the viewing time stops and waiting time starts again.

The distinction between primary and secondary requests and responses results is necessary to appropriately study the influence of delays occurring on the communication links. Due to the fact that the first secondary request cannot be issued before the primary response has arrived, the minimum contribution of these delays on the overall perceived latency is *twice* the round-trip delay.

In contrast to our analytical investigation of prefetching in which we have assumed a system without any delay, we use this more detailed model for system level simulations in order to achieve also insights on the effect of these delays.

We use random processes for synthetically generating the properties of the documents that will be the transferred data for our traffic model. Mah has derived empirical distributions for the following properties that characterize hypertext documents [Mah97]:

- size of primary requests
- size of primary responses
- number of embedded objects
- size of secondary requests
- size od secondary responses
- duration of viewing time

The waiting time the user has to endure is a function of the documents' structure and volume that is created by this model, the chosen strategy for prefetching and the parameters of the communication networks (delay, datarate, coverage etc.) over which the documents are to be transported. Fig. 3.13 shows an example of a resulting sequence of requests, responses and viewing times.

Now, that both models of user activity, i.e. the mobility model and the traffic model are defined, we are ready to present the results obtained by our simulations.

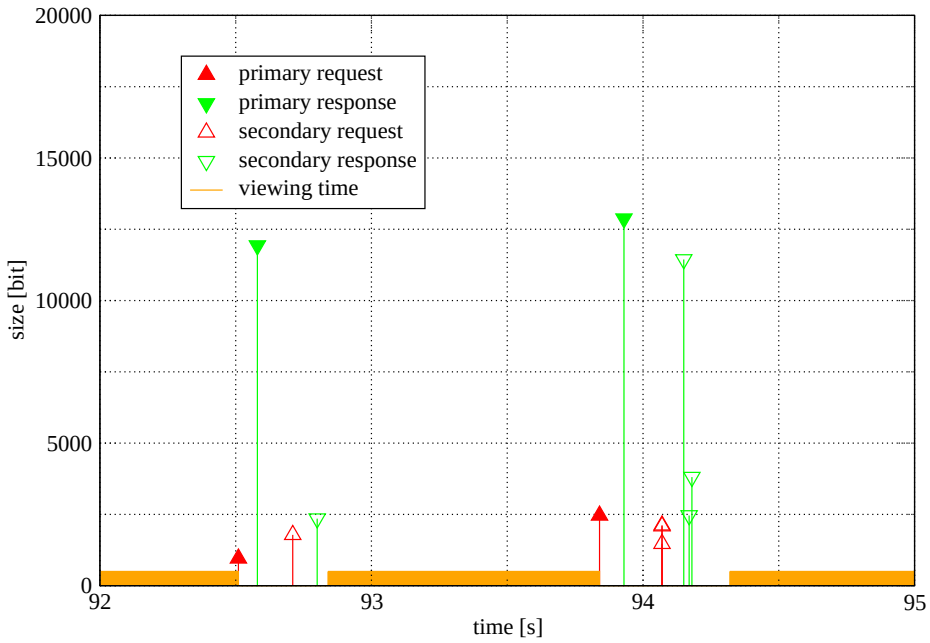


Figure 3.13: *Snapshot of sample HTTP requests and responses and viewing times. The primary request is issued at approx. 92.5 s and is followed by its corresponding primary response. The document contains only one embedded object. Therefore, only one secondary request is issued. After delivery of its corresponding secondary response the viewing time starts (orange) and is finished when the primary request for the next document is issued (approx. 93.8 s).*

3.4 Simulation Results and Discussion

With suitable models for the network, user mobility and traffic we are now equipped to perform the necessary simulations to study the influence of situation aware prefetching in scenarios that are too complex for purely analytical description and investigation. We will start with a scenario of a single user in the classical networking scenario. We will then continue with full-scale system simulations comprising 1000 users and up to 40 access points.

3.4.1 Single User, Classical Mobile Networking Scenario

3.4.1.1 Single Trial Experiment

The theoretical analysis performed in the previous chapter has resulted in quantitative laws that describe how much the average waiting time for the user is reduced and how much volume has to be transported when prefetching is applied. A first simulative verification of the theoretical results resulted in a good consensus for the theoretical and initial simulative results (see Section 2.2). The document model that has been used for theoretical analysis and the initial verification does not represent the internal structure of a hypertext document. In reality, this internal structure results in multiple requests and responses, which, in conjunction with delays caused by the network and the server, might influence the perceived performance. Therefore, our *system level simulation* uses the more detailed document and traffic model described in Section 3.3 and assumes delays for the network and the server. Two separate channels for requests and responses (uplink and downlink) are modelled with identical parameters. The data rate of each channel is chosen to be constant at $1 \cdot 10^5$ bit/s. Each channel introduces 0.1 seconds of delay. The server delay is fixed at 50 ms.

The channels are exclusively dedicated to one user. The user plan consists of 100 documents that are identically chosen for the prefetching and no-prefetching scenarios. The properties of each document are generated following Mah's model. The heavy-tailed distribution for the viewing times is modified by rejecting viewing times over 300 seconds. For the single user case this reduces the overall time for executing the user plan. The influence on the cumulated waiting times is only marginal, since these long viewing times would occur only rarely and usually all documents have been prefetched within 300 seconds. For the multi-user simulations we will later use the unmodified model, since for the multi-user case the influence of shortened viewing times and the resulting increase in average network load would falsify the simulation results.

Each document has $N = 7$ possible successors, from which one is drawn from a distribution following Zipf's law with parameters $N = 7$ and $\alpha = 1.0$.

Only two values for the prefetching threshold p_{th} are used: The no-prefetching case is simulated by setting $p_{th} = 1.0$, whereas for unrestrained prefetching p_{th} is set to 0.0.

In Fig. 3.14 the influence of prefetching is depicted. For the chosen parameters it takes the user 3343 seconds to complete his plan consisting of 100 documents if no prefetching is applied. The same plan is executed in 3263 seconds when prefetching is applied. It is necessary to note that these times would be longer if the viewing time were not limited to 300 seconds. After these durations no more documents are requested, resulting in the flat segment on the right of the curves. We see the cumulated waiting time the user has to endure with and without prefetching. While executing his plan without prefetching the user has to wait for 330.7 seconds. This is reduced to 250.4 seconds when prefetching is applied. In this trial prefetching reduced the waiting time by 80.3 seconds or 24.3%. The temporal development of the ratio $T_{w,prefetching}/T_{w,noprefetching}$ between the cumulated waiting time with and without prefetching is depicted in Fig. 3.15.

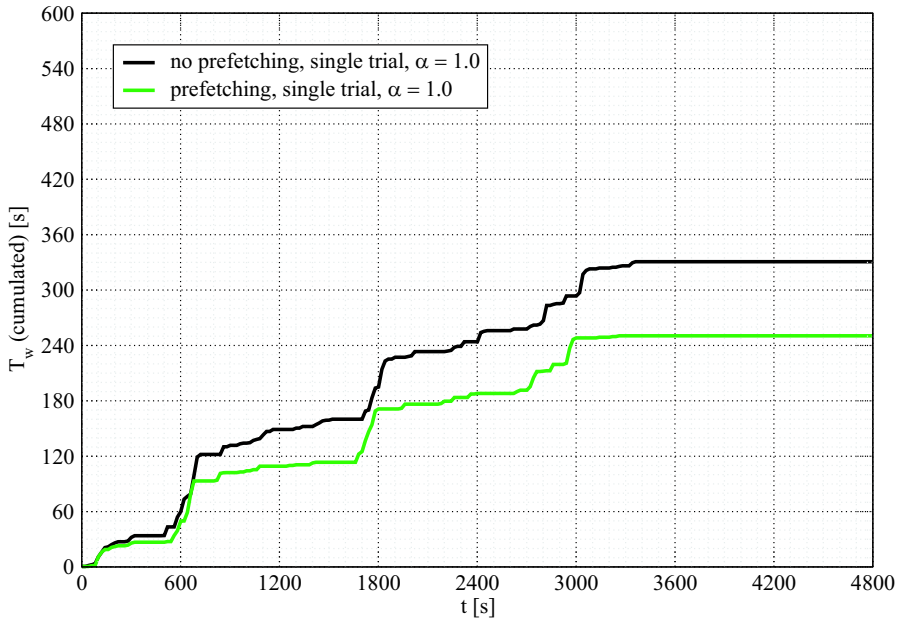


Figure 3.14: *Comparison of cumulated waiting time of single user in the classical mobile networking scenario with and without prefetching. The user's plan consists of 100 documents following a modified version of Mah's traffic model. When prefetching is applied, the user accomplishes the plan in 3263 seconds compared to 3343 seconds if no prefetching is applied. By applying prefetching the cumulated waiting time is reduced from 330.7 seconds to 250.4 seconds, which is a reduction by 80.3 seconds or 24.3%.*

The sporadically occurring increases of this ratio are caused by the fact that in the prefetching case the user arrives earlier at some larger documents that add up to the cumulated waiting time. This effect is only temporary, since in the case without prefetching the user will also request these documents.

Our theoretical analysis has shown that the achieved reduction of user waiting time is always accompanied by an increase in transferred data volume V. Fig. 3.16 shows

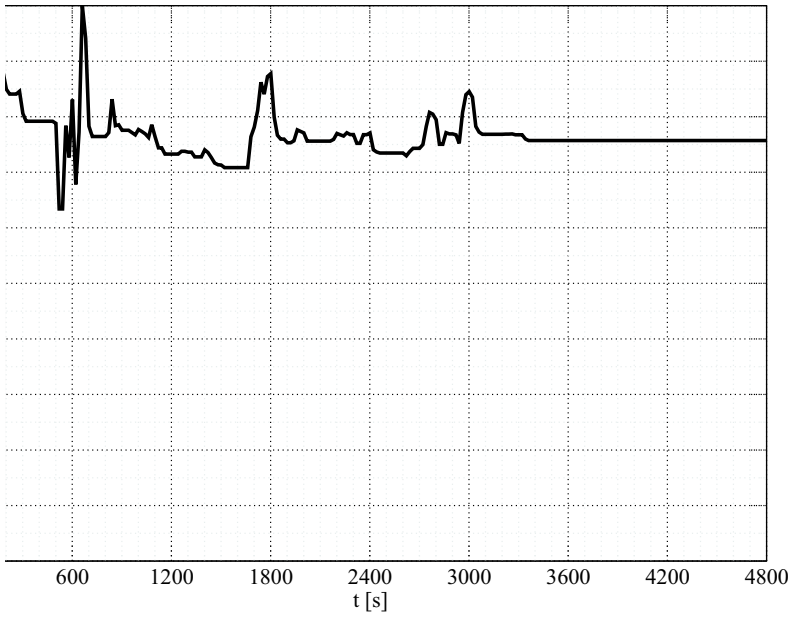


Figure 3.15: *Temporal development of the relative cumulated waiting time achieved by prefetching. After 100 documents the ratio between the cumulated waiting times is 75.7%.*

the absolute values of the cumulated transferred volume for the same user plan consisting of 100 documents. The cumulated volume for speculative and non-speculative requests and responses necessary for transferring the 100 documents amounts to $6.05{\cdot}10^7$ bits if prefetching is applied. Without prefetching only non-speculative requests and responses have to be transported and amount to $2.88 \cdot 10^7$ bits. The temporal development of the ratio is plotted in Fig. 3.17. We can clearly see that more than twice the amount of data has to be transported when prefetching is applied.

For networks in which the user is charged for the amount of transported volume (volume charge) this directly corresponds to an increase in cost of the same factor. If the user is charged for the overall time span he uses the network (time charge), he profits from the decrease of the necessary time to execute the complete plan. In the example, the total time necessary to execute the plan (sequence of documents) has been slightly reduced by 2.4%, which would result in the same amount of saved network costs if time charge is assumed. However, this result concerning the saved costs if time charging is assumed has to be taken with a grain of salt for two reasons. Firstly, because the overall plan execution time has been influenced, i.e. shortened by limiting viewing times to 300 seconds. Secondly, because for longer viewing times the user might decide to shutdown the network connection until he requires it again. It is an interesting fact that the charging policy chosen by (mobile) network operators is often not coupled to the connection type of the operated network. Often circuit-switched services are not time charged but volume charged and packet-switched services are not necessarily volume charged but time charged.

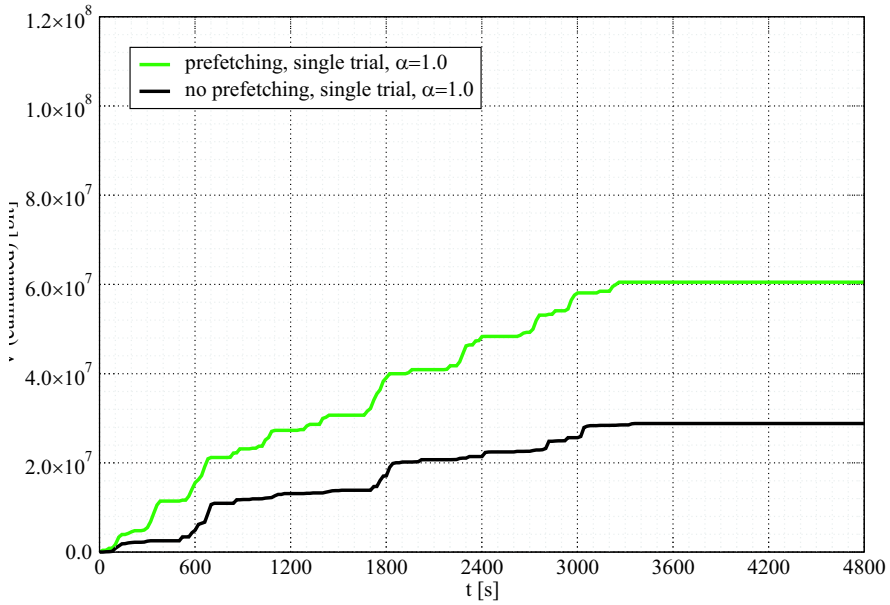


Figure 3.16: *Cumulated transferred volume for user plan consisting of 100 documents. The cumulated volume for speculative and non-speculative requests and responses necessary for transferring the 100 documents amounts to* $6.05 \cdot 10^7$ *bits if prefetching is applied. Without prefetching only non-speculative requests and responses have to be transported and amount to* $2.88 \cdot 10^7$ *bits.*

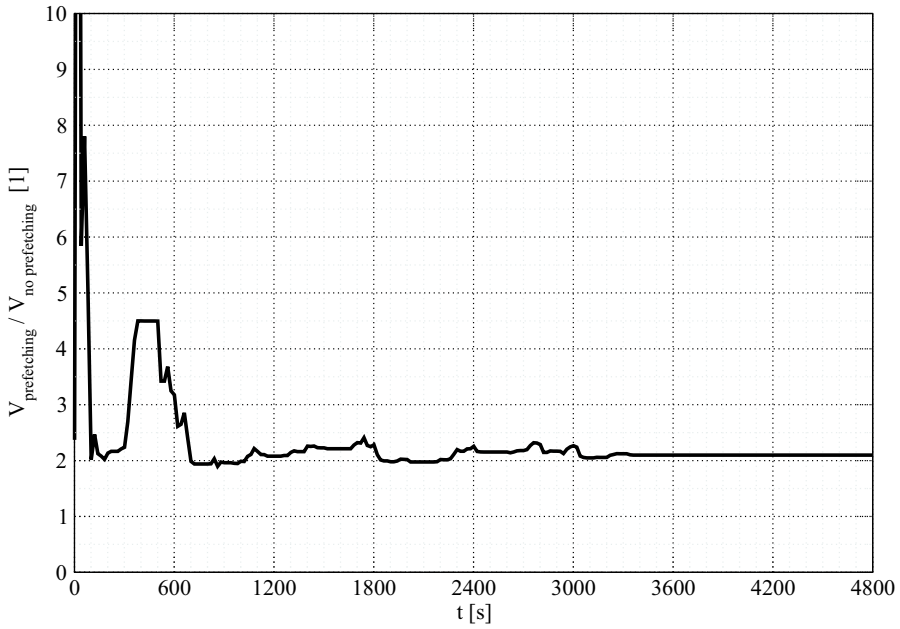


Figure 3.17: *Temporal development of the relative cumulated transported volume. After 100 documents the ratio between the cumulated transported volume is 210%.*

3.4.1.2 Multi Trial Experiment

The figures for the results of the single-shot trials show strong temporal fluctuations. The results are very sensitive to the outcomes of the individual random trials that generate the documents' requests and the viewing times. This is a well known effect that frequently occurs in simulations of application-layer traffic. For the purpose of obtaining better insight into how a system performs *on the average* the method of Monte Carlo simulations can be applied. The following results are obtained by averaging over 10 trials. The parameters are chosen identically to the single trial experiment. All trials have been initialized with different and random seeds of the random generators. To keep the high variation of the individual trials in mind their outcomes are also included in the respective diagrams. All system parameters are chosen identical to the single shot simulations. In Fig. 3.18 the average waiting time of 10 trials with and without prefetching is shown in addition with the outcomes of the individual trials. We can see the strong inter-trial variations. The respective ratios of the individual trials and the mean ratio of the waiting time are shown in Fig. 3.19.

In the single trial experiment described previously we seem to have picked a random sequence of documents and waiting times that resulted in a fairly representative perceived performance of the system. For the average over 10 trials we see a reduction in mean cumulated waiting time from 383.5 seconds without prefetching to 290.7 seconds when prefetching is applied. This reduction of 24.1% comes with an increase of almost exactly 100% in transferred data volume ($3.43 \cdot 10^7$ bit to $6.88 \cdot 10^7$ bit). (see Fig. 3.20 and Fig. 3.21).

The multi trial experiment confirmed the fact already hinted by the single trial experiment: While prefetching is able to reduce the waiting time to approximately 75%, the increase in transported volume is almost twofold for the chosen parameters. This is acceptable if the network resource is exclusively dedicated to a singe user, since otherwise the capacity would be wasted without any benefits for the user. The choice to apply prefetching effectively constitutes an otherwise not available degree of freedom that can be used to achieve an improved perceived performance without necessary changes in the networks infrastructure.

In the results presented so far, the decision whether to apply prefetching or not has been made by setting the prefetching threshold p_{th} either to 0.0 (prefetching all documents) or 1.0 (no prefetching). From our theoretical analysis we know that a better balance between the reduction in waiting time and the increase in transported traffic can be achieved by adjusting the prefetching threshold to intermediate values between 0.0 and 1.0.

Furthermore, according to the theory, the reduction in waiting time and the increase in transferred volume depends on the degree of randomness of the documents. If we assume a distribution of the probabilities according to Zipf's Law, this randomness may be adjusted with α.

To investigate the influence of α and p_{th}, simulations for various values of α and threshold probabilities p_{th} are performed. The results have been normalized with

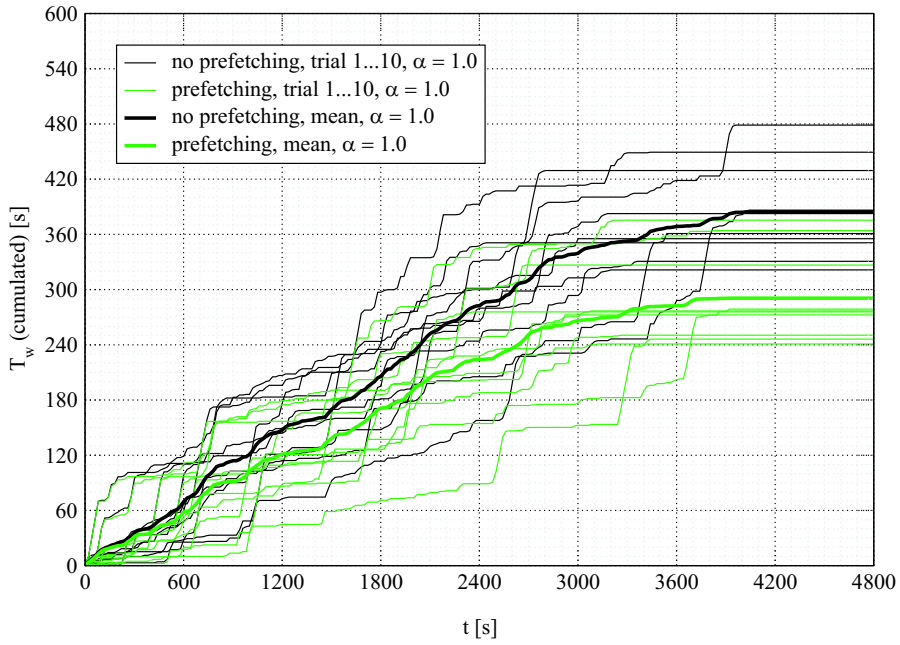


Figure 3.18: *Cumulated waiting times and averages of 10 trials with and without prefetching. After 100 documents the average waiting times is 383.5 seconds if no prefetching is applied. If prefetching is applied users have to wait for an average cumulated time of only 290.7 seconds.*

respect to the no prefetching case and are depecicted in Fig. 3.22.

The influence of both parameters p_{th} and α is showing in Fig. 3.22. When the documents probabilities strongly vary ($\alpha = 3.0$) a high prefetching threshold probability, i.e. low prefetching activity, already results in a significant reduction of waiting time ($\approx 30\%$), whereas the amount of transported volume increases only slightly ($\approx 7\%$). However, any further reduction of the prefetching threshold does not yield substantial additional benefit, but mainly increases the amount of transported volume by up to 80% if unrestricted prefetching is performed.

If the differences in the documents' probabilities are less pronounced, low prefetching activity does result in smaller reductions in waiting time of $\approx 18\%$ for $\alpha = 1.0$ and $\approx 15\%$ at the cost of an $\approx 27\%$ and $\approx 37\%$ increase in transported volume. These reductions can be increased up to $\approx 25\%$ at the cost of an additional increase in transported volume ($\approx 90\%$).
The obtained simulation results are corresponding nicely with the theoretic results from eqs. 2.74 and 2.79 respectively.

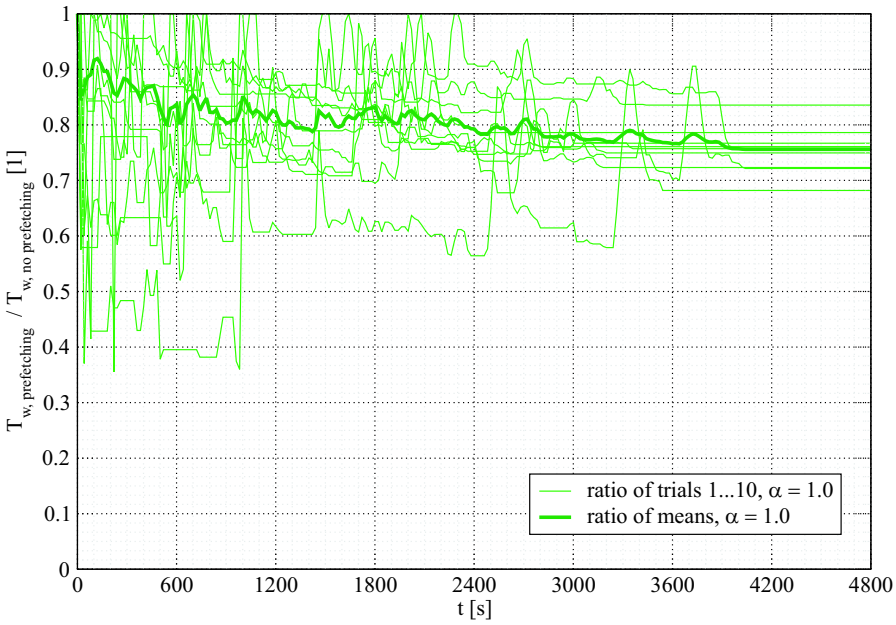


Figure 3.19: *Ratios of cumulated waiting times and ratio of averages of 10 trials with and without prefetching.*

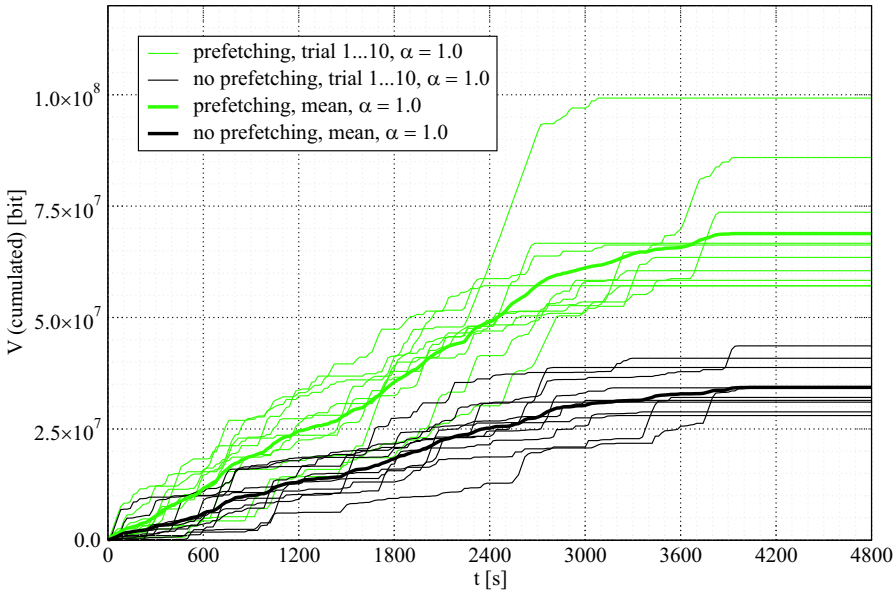


Figure 3.20: *Cumulated transported volumes and averages of 10 trials with and without prefetching.*

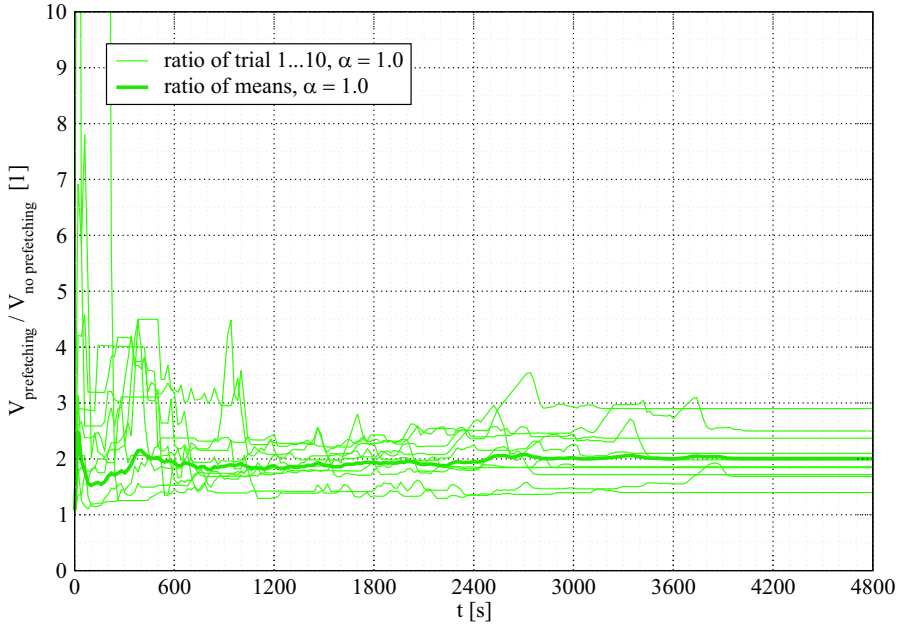


Figure 3.21: *Ratios of cumulated transported volume and ratio of averages of 10 trials with and without prefetching.*

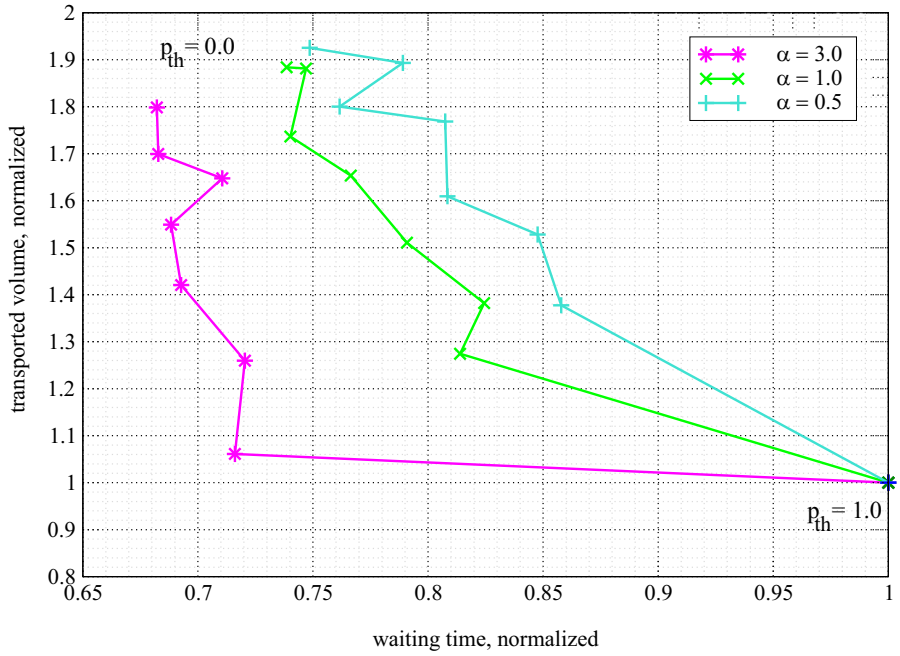


Figure 3.22: *Influence of both parameters p_{th} and α. For $\alpha = 3.0$ low prefetching activity already leads to a significant reduction of the waiting time without causing strong increases in transported volume.*

3.4.2 Multi User Scenarios

The previously performed simulations have been carried out on the basis of a scenario in which a single user enjoyed an exclusively dedicated network resource with constant data rate. This scenario is a reasonable model for most of today's mobile networks with standards such as GSM-CSD, GSM-HSCSD or GPRS with managed quality of service. For these cases it has been shown that prefetching offers an additional degree of freedom, which, depending on the individual user policy and the documents' probabilities, can be effectively used to increase the perceived performance of the network.

We wish to proceed our investigation of situation aware prefetching by verifying or revising our earlier conjecture that prefetching should be particularly beneficial in scenarios with heterogeneous networks and networks with coverage gaps. For this purpose a number of large-scale simulations are carried out. The simulations comprise a population of 1000 mobile users within an urban area as depicted in Fig. 3.9. The users are serviced by a mobile network in conjunction with up to 40 access points.

To obtain a first overview we compare the perceived performance with and without prefetching in the heterogeneous scenario (hybrid networking scenario) and the two homogenous scenarios (classic mobile networking scenario, access with coverage gaps scenario). Again, we choose the number of candidate successor documents to be $N = 7$ and the plan length to be 100 documents.

The following parameters are chosen on the basis of the measured values for GSM-GPRS and Bluetooth. Effective data rates for requests are $10 \cdot 10^3$ bit/s and $40 \cdot 10^3$ bit/s for responses via the mobile network. The bottleneck for the short range link shall be the device with $50 \cdot 10^3$ bit/s for the requests and $300 \cdot 10^3$ bit/s for the responses. Four piconets per access points are assumed, resulting in a combined effective data rate for requests of $200 \cdot 10^3$ bit/s and $1200 \cdot 10^3$ bit/s for responses. The delay for requests and for responses shall be 2 seconds for the mobile network and 0.5 seconds for the short range link. This configuration is an acceptable model of the implemented prototype system. For convenient comparison the results are all depicted in Fig. 3.23.

The **reference case** is the classical networking mobile networking scenario without any prefetching ("GSM only, no prefetching"). Since in this case no prefetching is applied the average amount of transported volume per document is equivalent with the average volume of the truly requested documents, which is $270.5 \cdot 10^3$ bit. The users had to wait for an average time of 15.3 seconds before they were presented with the complete page impression.

For conciseness we will use the term "average waiting time" for the average waiting time per document, and the term "average volume" for the average amount of transported volume per document.

We start with the two **homogenous networking scenarios**. When only the mobile network is used in conjunction with full prefetching **("GSM only, full prefetch-**

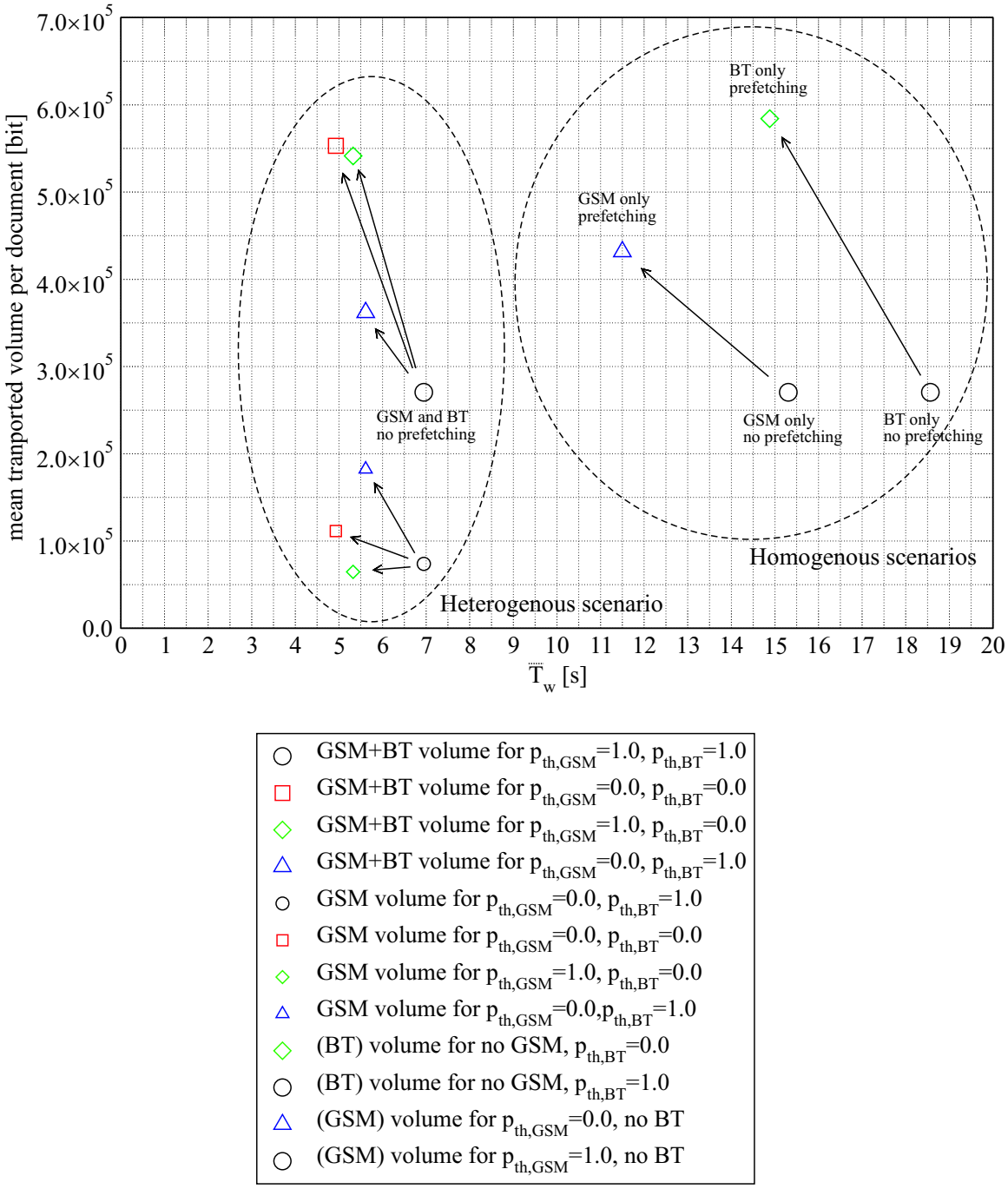


Figure 3.23: *Overview of initial simulation results on prefetching influence in the heterogeneous networking scenario (BT and GSM) and the two homogenous scenarios (GSM only and BT only). Each configuration results in particular combination of average waiting time per document $\overline{T}_w$ and average transported volume per document. The GSM share of the traffic in the heterogeneous scenarios is indicated with the four data points having smaller symbols (denoted as "GSM volume for..." in the legend).*

ing") the average waiting time is reduced to 11.5 seconds (−25%), whereas the average volume is increased to $431.8 \cdot 10^3$ bit (+60%).
Very interestingly the scenario with coverage gaps without prefetching **("BT only, no prefetching")** results in average waiting times of only 18.5 seconds, which is only 21% more than in the "GSM only, no prefetching" case. This result is unexpected, since the 40 access points in the scenario cover only a small fraction of the accessible area. When full prefetching is applied **("BT only, full prefetching")** the average waiting time is reduced to 14.8 seconds (−20%), which is even less than in the "GSM only, no prefetching" case. The average volume is increased to $582 \cdot 10^3$ bit (+116%). However, this does not result in any adverse effects.

The scenario with probably the most practical relevance is the **hybrid networking scenario**, for which the network is comprised of the mobile network with global coverage and number of short range access points with coverage gaps. The combination of both networks results in a significant improvement of the perceived performance. Without any prefetching the average waiting time is reduced to 6.9 seconds, which is a reduction by 54% compared to the "GSM only, no prefetching" case.

It is very interesting to notice the fact the combination of a mobile network with a short range network results not only in a considerable reduction of waiting time but also in a reduction of traffic transported via the mobile network. For the simulated configuration the GPRS share of the average volume drops from $270.5 \cdot 10^3$ bit to $73.8 \cdot 10^3$ bit (−73%). In other words, the short range network "steals" 73% of traffic from the mobile network.

For studying the influence of prefetching in the hybrid scenario we will initially distinguish between three cases:

a) full prefetching via the mobile network

b) full prefetching via the short range network

c) full prefetching via both networks

For **case a)**, i.e. full prefetching via the mobile network, the waiting time is further reduced to 5.61 seconds (−19%), whereas the average combined volume, transported over both networks is $362.0 \cdot 10^3$ bit (+34%). While the average combined volume only rises by 34% the share transported via the mobile network increases by 147%, from $73.8 \cdot 10^3$ bit to $182.3 \cdot 10^3$ bit. This drastic increase in traffic via the mobile network is likely to result in an unacceptable increase of cost incurred from network charges.

Potentially more attractive is **case b)**, in which prefetching is only performed via the short range network. In this case the average waiting time is reduced to 5.3 seconds, i.e. by 23% compared to the non-prefetching case. While the combined average volume increases to $541.4 \cdot 10^3$ bit (+100%), the share of the mobile network

actually drops to $64.7 \cdot 10^3$ bit (-12%) due to the prefetching activity via the short range network.

The maximum prefetching activity occurs in **case c)**, where unrestrained prefetching is performed via both networks, resulting in an average waiting time of 4.9 seconds, which is the minimum of all strategies discussed so far. This is a reduction of 29% compared to the non-prefetching case. The average combined volume increases to $555.0 \cdot 10^3$ bit ($+104\%$), which is in turn the maximum of all strategies for the hybrid network scenario, so far. Since prefetching activity is also performed via the mobile network, its share of the average volume rises by 51% to $111.5 \cdot 10^3$ bit.

The simulation results of the three cases lead us to strongly suggest case b) for a good balance between perceived performance and network cost. The results also illustrate the benefits of the combination of augmenting the mobile network with a short range network and prefetching, since these combined efforts result in a reduction of average waiting time by almost two-thirds (-65%) from 15.3 seconds to 5.3 seconds, while at the same time the average volume transported via the potentially expensive mobile network is reduced by more than three-quarters (-76%).

3.4.2.1 Influence of Document Probabilities

According to the results of the theoretical analysis we should expect a considerable influence of the document probabilities on both the average waiting time and the average volume. Again, the assumption of a Zipf distribution allows us to investigate this influence by varying the parameter α (see also Section 2.2.2.2). The parameter α is varied from $\alpha = 0.0$, which constitutes the worst case for prefetching, since in this case all documents are equally probable, to $\alpha = 3.0$, which models fairly good conditions for prefetching. Fig 3.24 shows the results for the heterogeneous networking scenario. For $\alpha = 1.0$ the results are identical to the results depicted in Fig. 3.23. As expected any variation towards higher α's reduces the average waiting time and the average volume. Interestingly the variation of α results in a fairly linear relation between the average waiting times and the average volume, at least for the range of $\alpha = 0.0$ to $\alpha = 3.0$. Nevertheless, the average combined volume cannot drop below $270.5 \cdot 10^3$ bit, which is the average volume without any prefetching , whereas the average waiting time could (almost) reach zero for very high α's, i.e. when all documents that are requested by the users, can be very well foreseen by the system.

3.4.2.2 Influence of Probability Threshold

Of course the Zipf parameter α is not adjustable in any real system but a result of the user's behavior. In contrast, the prefetching threshold probabilities $p_{th,BT}$ and $p_{th,GSM}$ are a degree of freedom that can be used to optimally adjust the system. For the initial simulations we have used only the extreme values of 0.0 and 1.0 for the threshold probabilities, which resulted in the average waiting times and average

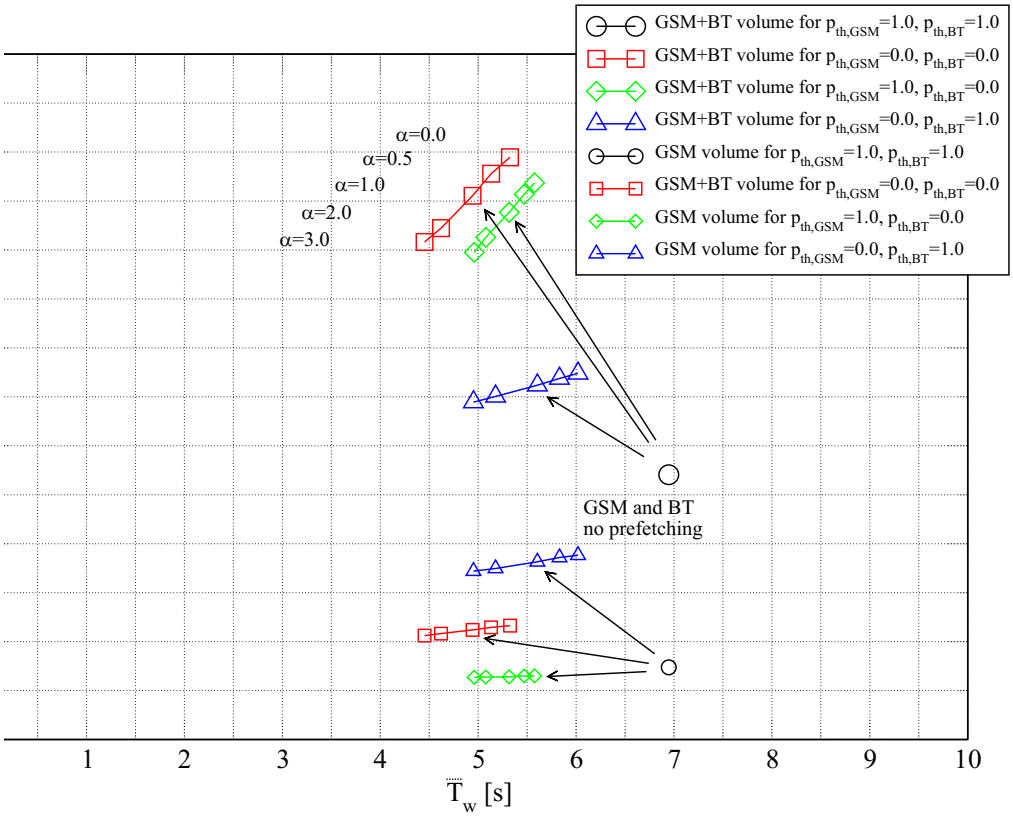


Figure 3.24: *Influence of document probabilities in heterogeneous scenario. The parameter α is varied from $\alpha = 0.0$ (worst case for prefetching, since in this case all documents are equally probable), to $\alpha = 3.0$ (fairly good conditions for prefetching). The GSM share of the traffic is indicated with the four data points having smaller symbols (denoted as "GSM volume for..." in the legend).*

volumes depicted in Fig. 3.23. The two prefetching thresholds actually constitute a two-dimensional degree of freedom, whose complete range can be used to fine-tune the system. The average waiting time and the average volume transported via the mobile network have been recorded for 64 simulations, corresponding to 8×8 combinations of prefetching threshold probabilities $p_{th,GSM}$ and $p_{th,BT}$. The results are depicted in Fig. 3.25. Certain combinations of prefetching thresholds are generally suboptimal, i.e. they can be replaced by another better combination, independently of a user policy (see also Section 2.2.4). They gray-shaded rectangular area is used to illustrate this effect: The combination $p_{th,GSM} = 1.0$, $p_{th,BT} = 0.0$ is better than all other combinations that lie in the gray shaded region, since is causes both shorter average waiting time *and* less average volume transported via the mobile network.

Following this argumentation and applying it to all data points in Fig. 3.25, we can derive the rule that $p_{th,BT}$ should always be set to 0.0, i.e. prefetching via the short range network should be unrestrained for the given scenario parameters. Observing

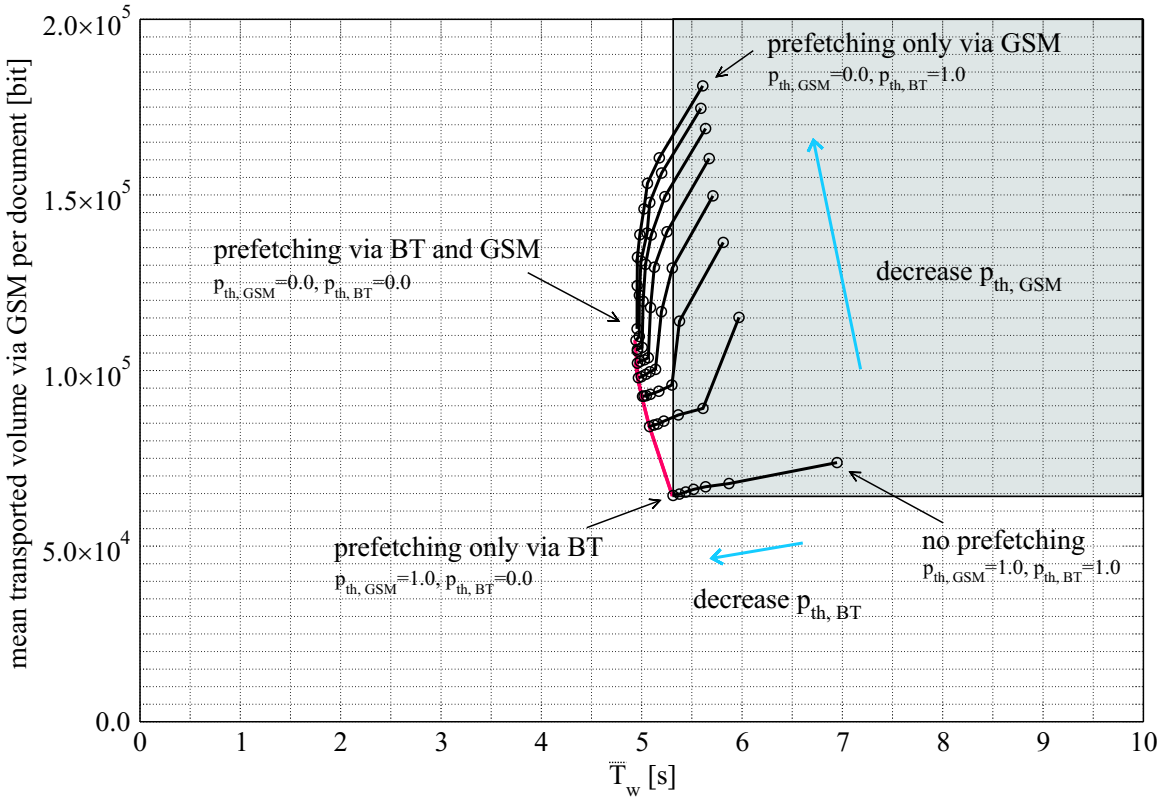


Figure 3.25: *Influence of probability threshold in heterogeneous scenario. The prefetching threshold probabilities $p_{th,BT}$ and $p_{th,GSM}$ are both varied from 0.0 to 1.0, with each combination resulting in a particular average waiting time and average volume transported via GSM. All combination not lying on the red curve are sub-optimal, independently of the user-policy.*

this rule, the threshold for prefetching via the mobile network $p_{th,GSM}$ can still be adjusted from 0.0 to 1.0, depending on the user policy[9]. This combinations that are generated by this variation are indicated with the red curve. However, only low or no prefetching activity via the mobile network is advisable, since the reduction in average waiting time is small compared to the increase in average volume transported via the expensive mobile network.

3.4.2.3 Influence of Number of Access Points

So far, the simulations involving the short range network have been performed with a fixed number of deployed access points within the urban layout. For the actual build-

[9]The obtained results lead to the interesting constellation, that a subset of combinations can be categorized to be *sub-optimal*, independently of the user-policy, whereas the *optimality* of certain combinations is only defined, given a particular user policy.

up of a short range network the number of access points is an important parameter, since it directly influences the systems costs. It is therefore important to obtain an understanding of the relation between the perceived performance and the number of access points.

Since situation aware prefetching has resulted in an improved perceived performance for the scenarios with 40 access points, it is especially interesting to see how it influences the perceived performance for smaller numbers of access points.

The partial coverage of the short range network in combination with the mobility of the users results in intermittent phases of availability and unavailability of the short range network. The durations of the phases shall be denoted $T_{d,1}$ (duration of availability) and $T_{d,0}$ (duration of un-availability). Their relative frequencies are displayed in Fig. 3.26.

We observe a difference in the effect of stepwise increasing the number of access points from 10 to 40. Whereas the durations of the availability phases almost remain unaffected, the durations of phases without coverage are successively reduced by increasing number of access points. Only after 35 access points have been deployed, a slight increase towards longer durations of availability is visible. Inspecting the layout of the coverage regions in Fig. 3.9, we see that this is to be expected, since the coverage regions do only start to overlap after 35 access points are deployed.

We have to be aware that both histograms use logarithmic scales on their axis of ordinates, "visual averaging" may be deceiving. Hence, the averages are computed and depicted in Fig. 3.27.

We see that while the average duration $\overline{T}_{d,1}$ of the availability phase only increases moderately from 86.8 seconds for 10 access points to 11.4 seconds for 40 access points, the average duration $\overline{T}_{d,0}$ of the unavailability phase drops considerably from 343.8 seconds to 76.2 seconds.

We revisit the coverage with gaps scenario again to investigate how the perceived performance is influenced by the number of access points. In this scenario, traffic is transported only via the short range network. Hence, we expect only poor performance for small numbers of access points. Fig. 3.28 shows the obtained results.

We see that configurations below 30 access points are hardly capable to provide average waiting times which would enable satisfying user interaction with a hypertext system. Without prefetching, the average waiting time only drops to 18.5 seconds when 40 access points are deployed. However, it is interesting to see that the same performance is already achieved with 34 access points when prefetching is applied. Hence, prefetching could compensate for 6 access points, which would save 15% of the costs for hardware and deployment.

More potential to achieve good performance is expected for the heterogeneous scenarios, where the coverage gaps of the short range network are filled by the mobile network.

Fig. 3.29 shows the dependency of the average waiting time on the number of access points.

We start by inspecting the case without any access points. This case is equivalent with the "GSM only, no prefetching" case in Fig. 3.23 and results in an average waiting time of 15.3 seconds. We see how increasing numbers of access points lead to a considerable decrease of the waiting time to 6.9 seconds for 40 access points which is in turn equivalent "GSM and BT, no prefetching" case in Fig. 3.23.

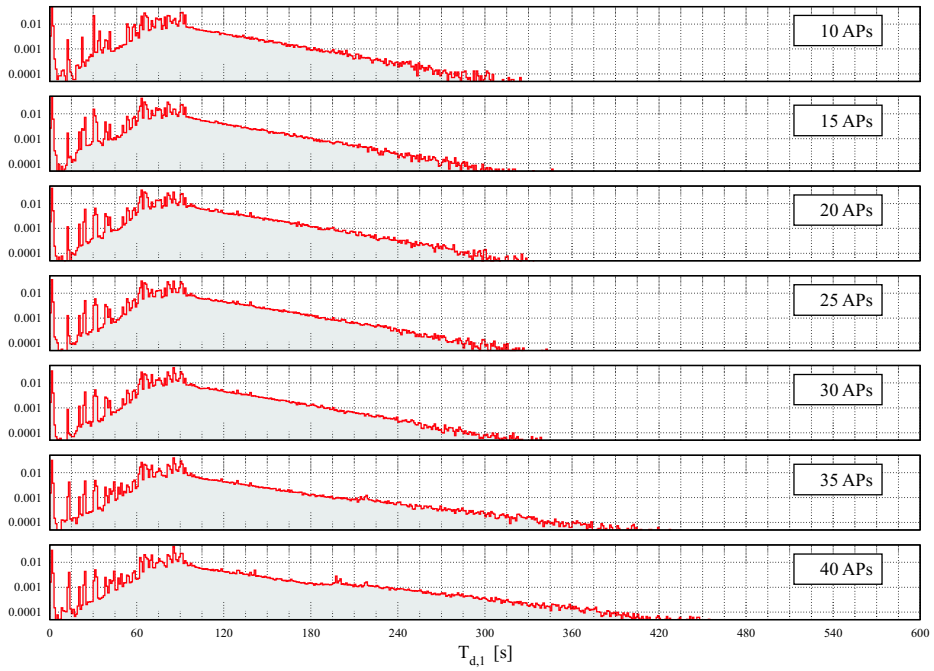
We can see clearly that situation aware prefetching over the short range network is particularly advantageous. For this case prefetching yields the same performance with 27 access points than what is achieved with 40 access points if no prefetching is performed, thus saving 13 access points or one third of the deployment cost.

If prefetching is additionally performed via the mobile network the average waiting time is further reduced. However, as can be seen in Fig. 3.30, this increase is only small and, as our earlier simulations already hinted, accompanied with a huge increase in average volume transported via the costly mobile network. In contrast, if full prefetching is performed only via the short range network, it leads to a slight reduction of the average volume transported via the mobile network.

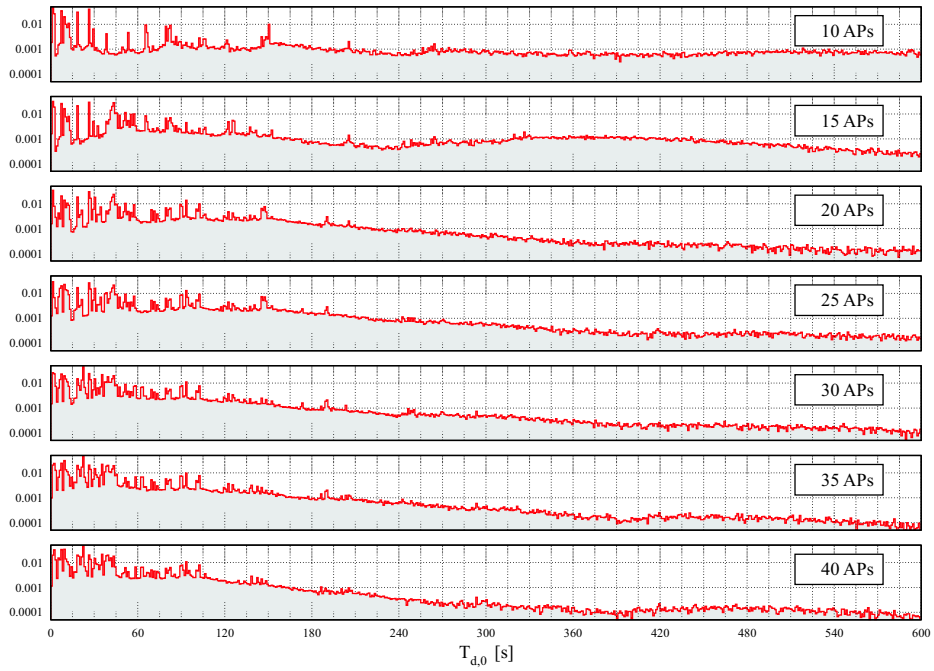
With the help of the performed simulations and the obtained results we have now gained considerable understanding of the effects of situation aware prefetching in both heterogeneous and hybrid networking scenarios.

We have seen how the average waiting time and transported volume depend on the documents' probabilities, the prefetching threshold probabilities and the number of deployed access points within the short range network. Especially the combination of a globally available low-rate and partially available high-rate communication network and situation prefetching has proved to have particularly beneficial effects on perceived performance and network costs.

We continue our presentation of situation aware prefetching by discussing certain implementation aspects, such as a suitable system architecture and protocols, performance measurements and experiences obtained from initial deployment and operations.



(a) duration of availability



(b) duration of un-availability

Figure 3.26: *Influence of number of access points on relative frequencies of durations of availability and unavailability of short range network.*

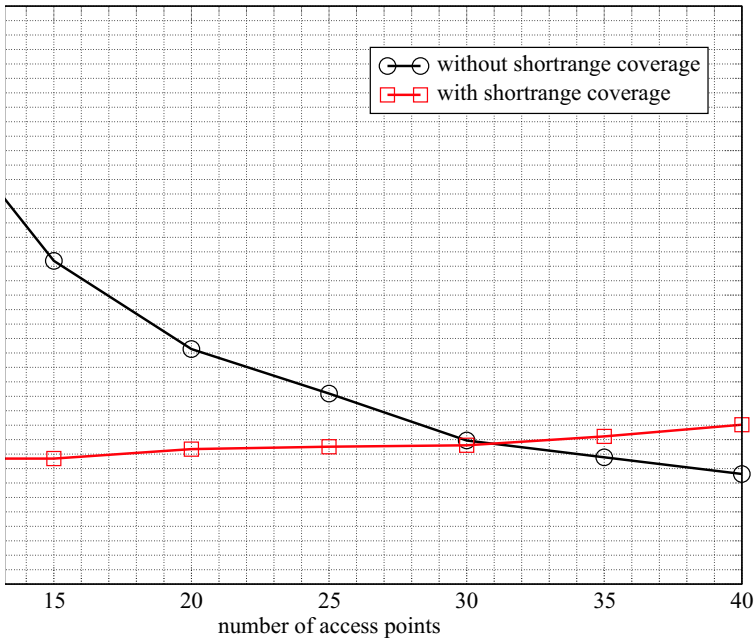


Figure 3.27: *Influence of number of access points on average durations of availability and unavailability of short range network.*

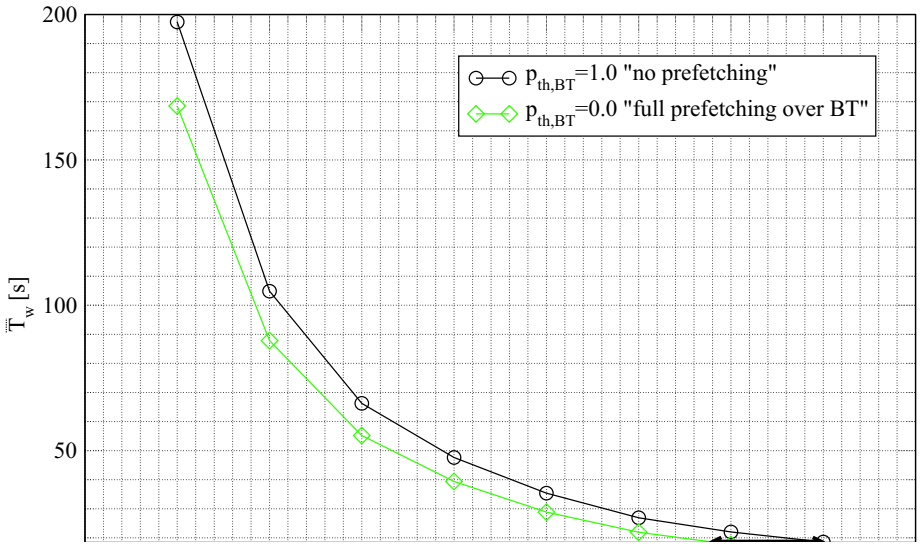


Figure 3.28: *Influence of prefetching and number of access points on average waiting times in coverage with gaps scenario (no mobile network is available.).*

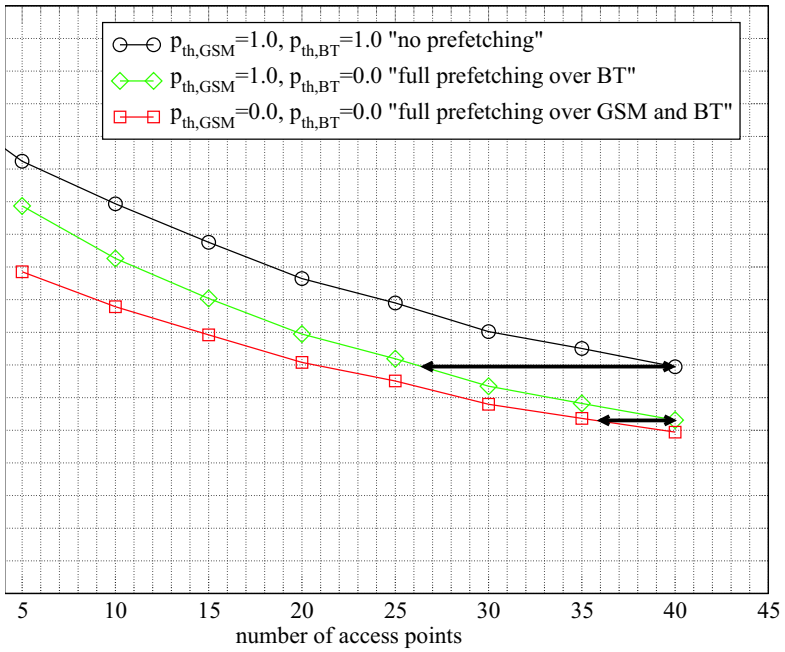


Figure 3.29: *Influence of prefetching and number of access points in heterogeneous scenario on average waiting time.*

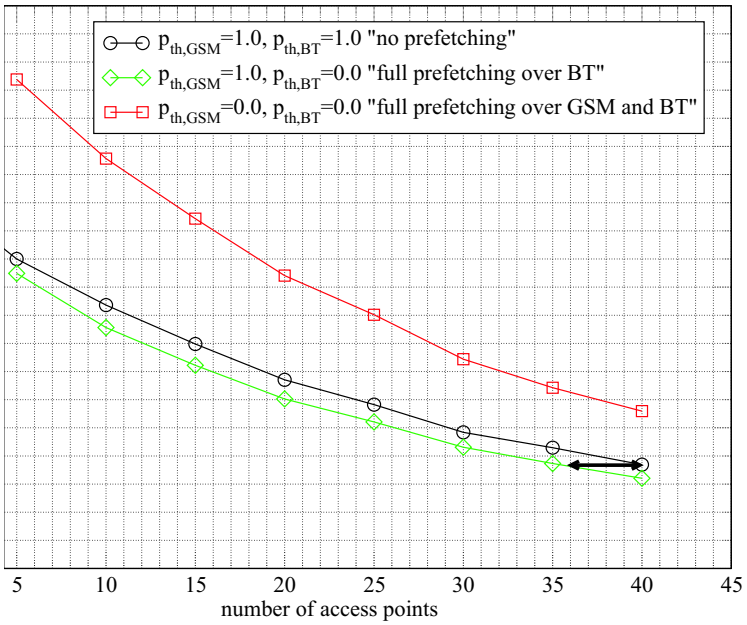


Figure 3.30: *Influence of prefetching and number of access points in heterogeneous scenario on average transported volume via mobile network.*

Chapter 4

Implementation Aspects

Both the theoretical analysis as well as the simulation results show that situation-aware prefetching does significantly improve the perceived performance of a hypertext system in theory. Furthermore, the investigation has shown that especially the combination with a heterogeneous wireless access network has large potential to improve the user perceived performance. Hence, a realization of the concepts promises to be worthwhile. In order to prepare a future realization several implementation aspects are investigated.

We start a brief discussion of **relevant conditions and constraints in mobile networks and devices** (4.1) by describing **networking conditions** (4.1.1) and **software conditions** (4.1.2). Under consideration of these conditions a **system architecture for situation-aware prefetching under heterogeneous networking conditions** is presented (4.2) with a focus on its elements for **situation awareness** (4.2.1), **application layer proxies** (4.2.2) and **cache consistency** (4.2.3). **Software development, integration and test** is briefly described (4.3) and results for **performance measurements** are given (4.3.1). Experiences obtained during the phases of **deployment and initial operations** in the city of Landsberg am Lech, Germany (4.4) conclude the discussion of implementation aspects.

4.1 Relevant Conditions and Constraints in Mobile Networks and Devices

Modern mobile communication networks with packet-switched services, such as GPRS and UMTS in conjunction with sophisticated mobile devices that provide still limited, yet considerable resources in terms of CPU performance, memory and graphical user interface, form a basis for mobile access to hypertext systems such as the world wide web. Despite huge differences in network cost and data rates, no fundamental differences exist between the software components used for accessing the world wide web from a desktop computer or a smartphone.

4.1.1 Networking Conditions

Neither the client application (CA), i.e. the browser, nor the server application (SA), i.e. the webserver have to be aware of the fact that the transport of the documents takes place via a mobile network. Fig. 4.1 illustrates how the the CA on the wireless information device (WID) and the SA are isolated from the actual network structure by the TCP layer.

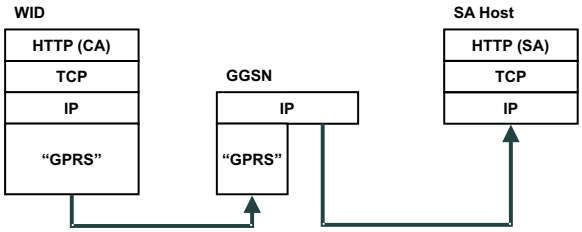


Figure 4.1: *Simplified protocol stack for a hypertext client application (CA), residing on a wireless information device (WID) using a mobile network's data service. The actual client and server application (SA) are isolated from the mobile network by a TCP layer. The GGSN (GPRS gateway support node) acts as gateway between the mobile network and the public internet. For a detailed view of the GPRS protocol stack see [BVE99].*

For the purpose of designing and implementing an architecture to facilitate prefetching and the use of additional short range networks such as Bluetooth or wireless LAN, some constraints have to be considered.

Since European mobile networks currently use IPv4, most operators employ network address translation (NAT) in order to conserve the use of address space [SE01]. Hence, most GPRS terminal will be "behind" NAT-gateways. This typically prohibits the use of static IP addresses for the terminals, since the addresses will be dynamically assigned from a pool of addresses by the NAT-gateway. Therefore, a protocol should not require incoming connections on the mobile device. Instead, connections have to be established from the mobile terminal to a server. Similarly, the use of incoming datagrams via UDP (user datagram protocol) should be avoided, since the mobile terminal's may have changed and the prior address may have been already reused.

While typically the mobile network is used to access the internet, other wireless communication means are also present on mobile phones.

Bluetooth has effectively replaced the previous de-facto standard, IrDA[1], for communicating between mobile terminals and other devices such as laptops. In contrast to IrDA, which requires free line-of-sight for its infrared link, Bluetooth does not require line-of-sight and allows distances up to 100 meters. This property of Bluetooth in combination with its extensive proliferation makes it an ideal candidate for a short

[1]Infrared Data Association, a set of physical specifications and communications protocol standards for the short range exchange of data using infrared light

range network to augment the existing mobile networks. Furthermore the operation of Bluetooth is license-free, which, in combination with inexpensive common-of-the-shelf components enables the provisioning of mobile network access with considerable performance at fairly low cost.

Today, this potential competition is not desired by today's mobile network operators. As a consequence, the software versions of mobile devices released to the public provide only the functionality to connect an external device, such as a laptop using Bluetooth to the phone and via the mobile network to the internet. In this case the mobile phone operates as a modem. In contrast, the phones are usually not capable of using Bluetooth to connect themselves via an access point to the internet. In this case the phone would have to act as a PPP[2] client. The PPP client functionality is in fact present on the phones, since it is necessary to connect to the internet of circuit-switched connection, such as GSM-CSD or GSM-HSCSD is used. Obviously, the lack of this feature is not caused by technical reasons.

So far, wireless LAN (WLAN) has not been integrated into mobile phones, mainly due to concerns about its power consumption. However, recent announcement regarding suitable chip-sets in combination with an increasing commercial interest in the market potential of WLAN "hotspots", prompts us to consider it when designing our architecture.

4.1.2 Software Conditions

In order to equip a mobile device with the capability to perform situation-aware prefetching it is necessary to observe and partially modify its communication with the server application.

A majority of mobile devices such as smartphones or PDAs is already equipped with client applications ("mobile browsers") that are capable of parsing and rendering most of today's hypertext markup languages such as HTML, WML or X-HTML and image formats such as JPEG[3], GIF[4] or PNG[5]. Typically these client applications are strongly embedded in the device's operating system (e.g. Symbian OS, Microsoft Pocket PC). Hence, any modification of the client application by third parties is difficult to achieve. A completely new development of a client application is hardly recommendable, since the development effort of a client application with reasonable functionality is estimated in the range of several dozen man-years.

Fortunately, two facts make it possible to overcome this problem. Firstly, the existing mobile client applications are following standard web protocols and are capable of properly communicating via an HTTP-proxy. The network address of the proxy is freely configurable. Secondly, the operating systems have multitasking capabilities, which allow the quasi-concurrent execution of processes. Third-party software

[2]Point to Point Protocol
[3]Joint Photographic Experts Group, lossy image compression standard
[4]Graphics Interchange Format, lossless image compression standard
[5]Portable Network Graphics, lossless image compression standard

for these operating systems can be developed in high-level programming languages (C++, Java) with reasonable effort, which enables the design and implementation of a software architecture that makes use of a proxy. The proxy is put into the communication path between the client application and the server application in order to enhance the perceived performance of the hypertext system by situation-aware prefetching.

4.2 System Architecture

During our theoretical investigation of situation awareness and its application to prefetching in heterogeneous wireless networks we have assumed the existence of an entity that collects relevant statistical data and derives and executes suitable actions regarding prefetching and network selection.

In the following we will discuss the design of a distributed architecture that performs the tasks of this previously abstract entity.

The minimum benchmark for our architecture shall be its capability to fulfill three essential tasks:

a) collect and aggregate the relevant situation information,

b) facilitate communication between the mobile device and other nodes via both the mobile network (PLMN) and the short range network for both context and content,

c) proactively retrieve suitable content onto the mobile device.

However, we strive to keep the architecture as open as possible for further situation aware application besides prefetching. Hence, we introduce several generic components into the architecture and then show their realization for fulfilling the three tasks stated above.

4.2.1 Elements for Situation Awareness

In our model of a situation space we separated between symptoms, which give (partial) information about the user's situation, and consequences that represent actions the user or another entity may take or require. Within our architecture we shall therefore distinguish between sensors for symptoms (*symptom sensors*) and sensors for consequences (*consequence sensors*). Since the purpose of achieving situation awareness is the ability to proactively perform actions, that may be helpful to the user, instances that actually execute the actions are required. We shall term these instances *consequence effectors*. Both symptom sensors and consequence effectors may be located on the device (mobile ...), in the environment (resident ...) or in the central network (central ...).

The observations made by the various symptom sensors have to be aggregated and used to derive sensible consequences. For this purpose a *situation inference engine* (SIE) is included into the architecture. Instances of the SIE may be on the device (M-SIE, mobile SIE), within the central network (C-SIE, central SIE) or in the environment (R-SIE, resident SIE), i.e. on network nodes directly accessible via short range communication. Several advantages come with each location of the SIE. While a M-SIE has the clear advantage to propose suitable consequences even without any network connection and without compromising personal data to any other instances, it suffers from the devices' limits in computational performance and memory. A C-SIE has the advantages of sufficient computational resources and the ability to propose consequences to consequence effectors, residing in the central network, even without any connection to the mobile device.

Since a user has to establish only a trust relationship to one or few C-SIEs, it has also clear advantages over multiple R-SIEs, with whom the user would have to establish multiple trust relationships.

While the previously introduced sensors fulfill the task of collecting situation information, it is still necessary to transport this information towards the SIEs, and the proposed consequences from the SIEs to the consequence effectors whenever sensors, SIE and effectors are not located on the same node. Additionally, for the case of a hypertext application, it is necessary to transport the requests and responses for documents over the selected network(s).

4.2.2 Application Layer Proxies

It has turned out that a combination of multiple application layer proxies is ideally suited for fulfilling the three essential tasks mentioned above. In the following we will point out the arguments that have led to this architectural decision and discuss the function of the involved communication elements.

For efficient communication via a short range network as well as enabling intra- and inter-network handover, Kammann proposed a split-proxy architecture which allows to use a lightweight, wireless adapted serial protocol between the mobile device and the short range access point [KB02]. It is well known that the Transport Control Protocol (TCP) of the standard Internet TCP/IP protocol suite is not well adapted to the properties of the wireless communication channel. This leads to significantly degraded performance in terms of achievable throughput over wireless links. The most prominent reason for this performance degradation resides in the flow control mechanism of TCP. This flow control mechanism has been designed to work with fixed line network communication channels. With this type of channel, errors usually are caused by congestion of network components e.g. queues in routers or network adapters. The TCP flow control algorithm therefore tries to resolve the congestion by throttling the rate of transmission. When applied to a wireless communication channel the errors introduced by the wireless channel, caused by ambient noise and

multipath fading, are mistaken for congestion by the flow control algorithm, leading to a reduction of the packet rate without any positive result [XPMS01]. In addition to this and minor other problems with TCP over wireless links the commonly used PPP protocol for enabling TCP/IP over the serial wireless connection causes significant setup times. Especially in scenarios with small coverage areas (e.g. 802.11 or Bluetooth) and high mobility these setup times are in the same order of magnitude as the actual durations of radio contact. On the other hand no real necessity exists to make a mobile device a full-featured Internet node only for the purpose of enabling hypertext applications.

For our intentions to apply situation awareness to prefetching, the three proxies proposed by Kammann are well suited entities to realize the functionality of the mobile, resident and central sensors for symptoms and consequences. The proxies shall be termed mobile proxy (MP), resident proxy (RP) and central proxy, respectively. The mobile proxy is also ideally suited to implement the functionality of the consequence effector for prefetching, since, by definition, it can behave like the client and request documents without the need for changes in the HTTP-protocol. The central proxy implements or is the contact point for the situation inference engine (SIE).

In Fig. 4.2 all software entities (processes) and hardware entities (devices) and the respective communication links among them are depicted. Hardware entities are shown as rectangular boxes with software instances allocated to them as oval shapes. The arrows between software entities show request/response relations between them (with arrows indicating the direction of the request).

Each wireless information device (WID) is typically equipped with a subset of possible communication technologies (Bluetooth, IEEE 802.11, GSM/GPRS, UMTS and others). Two software entities reside on these devices: Firstly, the client application (CA), which is in most cases a multi-format (HTML/WML/XHTML) browser, but can also be a domain-specific (e.g. touristical sightseeing, technical maintenance, tele-medicine) browsing application for improved navigation or any form of agent application that performs tasks for the user that necessitate communication with the outside world. Secondly, the mobile proxy which functions in cooperation with its peers (resident proxy, central proxy) in the fixed part of the network, to provide communication services to the CA. The MP itself makes use of the devices communication capabilities which are provided in an operating-system specific way by kernel procedures (kernel entity not depicted).

The short range communication technologies are used by the MP to connect to an available resident proxy (RP) which resides on a stationary device termed local service point (LSP). Usually the RPs only forward the requests/responses to/from the third proxy type termed central proxy (CP) residing on a central proxy host which is part of the internet. LSPs are using local area networks (ethernet), wireless local area networks (IEEE 802.11), public switched telephony networks (PSTN) e.g. dial up, DSL) or, in rare cases, public land mobile networks (PLMN) to connect to the internet. A special form of LSP, which is not connected to the internet hosts a server application in order to provide local content is depicted in Fig. 4.3.

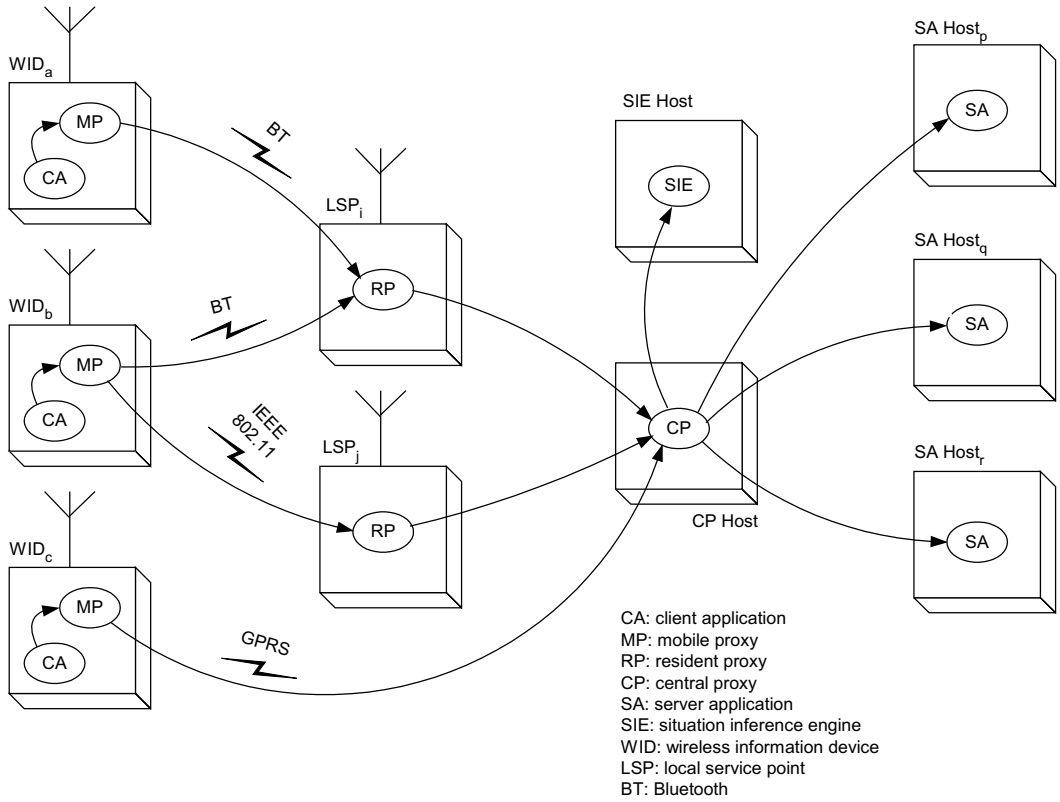


Figure 4.2: *Multi-proxy network architecture.*

This configuration is suitable for providing relatively static content at remote places where a permanent network connection may not be available or be too costly.

If the LSP is connected to the central network, the central proxy (CP) forwards the requests to the server applications (SA), typically web- or application servers residing on their SA hardware. For WIDs that have generally or temporarily (due to coverage gaps) no short range communication capability, the CP is directly communicating with the MPs without employing any RP. The involvement of the CP in all communication events facilitates it to act as sensor for the symptoms and consequences regarding communication issues, such as document retrieval or network selection. The CP reports this information to a situation inference engine, a software instance residing on its dedicated host hardware (SIE host). The SIE is responsible for storing statistical information and to infer and advice on future consequences. The advice which is to be used by the MP and certain forms of client applications is included into the HTTP-header of the responses by the central proxy (piggy-backing).

While Figs. 4.2 and 4.3 show only the logical connections on the TCP-plane, Fig. 4.4 shows in more detail how the protocol stacks on the involved nodes relate to each other. However, the portion of the stacks below the IP layer are abstracted

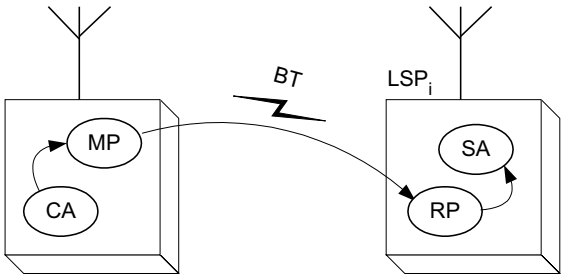


Figure 4.3: *Isolated Local Service Point (LSP). A server application is residing on the LSP and provides local content without requiring connectivity to the central network.*

by "GPRS", "WLAN" and "Bluetooth".

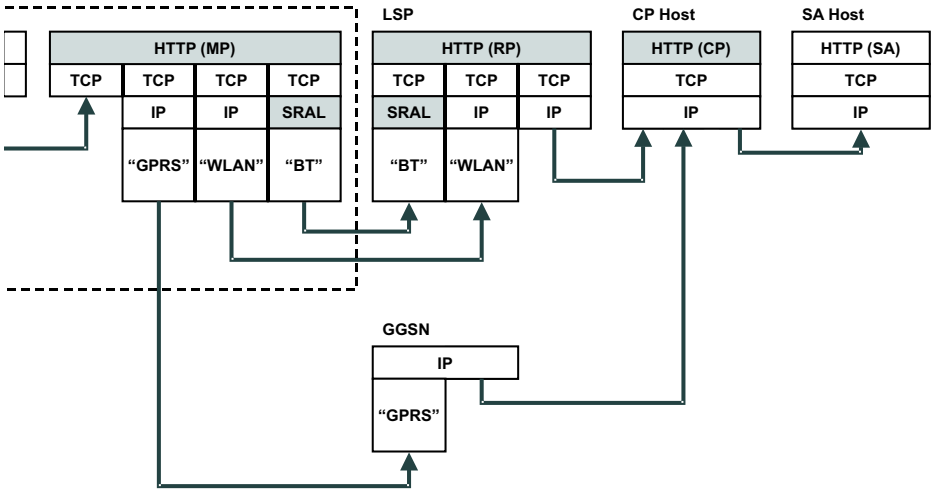


Figure 4.4: *Protocol layer perspective of network architecture. Wherever possible existing components have been employed (extensions are indicated by gray-shading.). Extensions encompass the three proxy components (MP, RP, CP) and a slim short-range adaptation layer (SRAL).*

An important benefit of the proposed architecture is the possibility to counter an existing waste of resources that is hard to tolerate in a wireless world. Despite the fact that optional compression (source coding) is standardized for HTTP it is de facto not existing due to lacking implementation or activation in most server and client applications. Since all documents pass through the CP it can apply standard universal source compression techniques such as forms of the Lempel-Ziv algorithm. The MP performs the necessary decoding without the need for cooperation of server or client application. In addition to this form of lossless form of compression, the CP is in a suitable position to adapt content towards the device constraints (e.g. screen

resolution, color depth, installed codecs) of the WID, which facilitates significant conservation of communication resources as well as storage on the device. Furthermore the CP plays a vital role in a proposed scheme to mitigate the adverse effects of dynamic content generation on achievable caching and prefetching performance. We will use the following section to discuss the nature of these effects, introduce the scheme and further illustrate the communication between the previously introduced software entities.

4.2.3 Cache Consistency and Dynamic Content

A problem which does not only occur if prefetching of content is performed, but whenever copies of data are kept, is the possibility of a change in the original data. Whenever such a change happens the copy becomes *stale* or invalid[6]. Usually it has to be prevented that a consumer of the data receives a stale version. Algorithms and protocol extensions that prevent this use of stale data are usually termed *cache invalidation schemes* and have been a field of intense research termed *cache consistency*. For the cache configuration and application considered here, we can distinguish between four well-known and fundamentally distinct schemes:

1. temporal invalidation,
2. location-dependent invalidation,
3. active validation/invalidation by client,
4. invalidation by callback.

Temporal invalidation uses an expiration date which is assigned to the copy of the data. After this expiration date the copy is considered stale. This scheme is realized in HTTP by transferring the expiration date with an optional header field.

Example: Expires: Wed, 31 Dec 2003 18:00:00 GMT

It should be noticed that this scheme is not a general solution to the cache consistency problem, but constitutes an agreement between the consumer and the provider of data where the provider assures that the original will not be changed until the expiration date, or if it is changed before the expiration date no negative effects are caused by using the stale copy (i.e. the data is stale but not invalid).

Location-dependent cache invalidation has been proposed in [ZL01]. This scheme considers the case of a mobile user with known geographical position for whom data is considered to be valid only if he is within a certain geographical region. As an example the query for "the nearest restaurant" is given. The result for this query is clearly depending on the user's location. Once the user is moving into a region where another restaurant is closer, the old result is no longer valid. To achieve sensible

[6] A useful definition of terms is *stale* – the original data has been changed, *invalid* – the copy is stale and and it has negative effects to use the stale copy.

validation it is proposed to store a region, called *valid scope area* with the copy of the data. Zheng et al observe that queries that specify the search for an object with the predicate "nearest" become patches of a Voronoi diagram. The data is considered stale and invalid when the user leaves the valid scope area.

When **active validation by client** is employed, the client has the responsibility to validate the copy each time the copy is accessed. The client therefore actively contacts the server the original data was retrieved from, and asks whether the original data has been changed since the time instant the copy had been made. For this purpose the cache instance stores this time instant with the copy. Within the HTTP protocol this scheme is employed wherever a client uses a method (usually GET) and makes it conditional by an optional header.

Example: If_Modified_Since : Thu, 27 Mar 2003 11:20:00 GMT

If the document has changed since the specified time, the server returns the newer version. If it has not changed, the server returns an error-code (304) that tells the client that the copy is still valid.

The **invalidation by callback** scheme requires the server to keep track of all copies that have been made. Whenever the original data is changed the server actively notifies all instances that keep copies that a never version exists. This scheme is not realized within the HTTP protocol.

Another important issue that has a strong practical influence on caching and prefetching is dynamically generated content. A large and increasing number of websites make use of the modern webservers' ability to keep track of user sessions by dynamical generation of the links e.g. within HTML-files. When a request corresponding to a dynamically generated link arrives at the server, the server application is capable of separating the information to the necessary data from the additional information that is used for keeping track of sessions. As a result it frequently occurs that identical content is served under multiple distinct URLs. Fig. 4.5 illustrates the problem.

Frequently, the adverse effects of this phenomenon are mentioned as arguments against the employment of caching and prefetching since the percentage of cacheable documents is significantly reduced.

A similar phenomenon occurs also for statical content in cases where the same data e.g. a certain image used for an icon, is used by multiple websites under potentially distinct filenames. Fig. 4.6 illustrates the effect.

This effect is not hindering caching and prefetching, since all instances of the data are cacheable and can be re-used. Nevertheless additional transmission and storage resources could be conserved if the quality of the data could be detected.

We have been elaborating on the issues of cache consistency and dynamic content generation in order to set the scene for our proposed scheme to cope with these problems for typical mobile scenarios. The effects of the scheme are effective under the assumption that the communication capacity is significantly lower and the latency significantly higher for the wireless access link than within the core network. Under

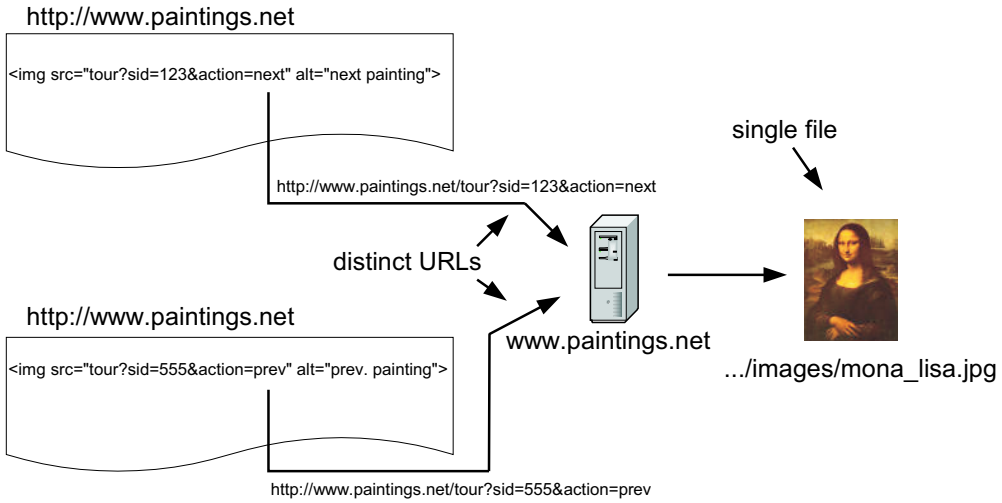


Figure 4.5: *Distinct dynamic URLs/identical content phenomenon. Increasingly, website access is managed by storing session information in dynamically generated URLs. The content is usually tagged as un-cacheable. However, most of the content is static (e.g. images) and not generated dynamically.*

this assumption it is feasible and beneficial to actively attempt validation of all content, even if it has not been explicitly approved for caching. Within this scheme we step away from using URLs or URIs to identify and index data. Instead, well known *message digestion algorithms* such as MD5[7] and SHA-1[8] are used that facilitate rapid comparision of data for equality. This enables us to check for validity of content without any regard of the – potentially dynamically – assigned label of (i.e. link pointing to) the data.

Figures 4.7 and 4.8 illustrate the communication between the five entities of the proposed protocol necessary for mitigating the previously described effect of dynamic content generation: It is important to keep the significant differences in data rate, latency and cost of the various communication links in mind. The CA and the MP reside on the same device and are therefore assumed to achieve very high data rates, whereas between the MP and RP the link will always be slower than all other links in the system due to the character of the wireless medium. These different data rates and latencies are indicated (not to scale) in the figures. For easier reference we mark events or steps of interest with corresponding circled numbers (e.g. ①) in the text and figures.

Beginning at ① the CA issues a request Req_1 towards the mobile proxy (MP) residing on the same device. The MP includes a unique ID_{WID}, identifying the WID in the header and forwards ② the request via wireless short range communication to

[7]MD5, message digest algorithm, 128 bit digest length, RFC 1321
[8]SHA-1, message digest algorithm, 160 bit digest length, RFC 3174

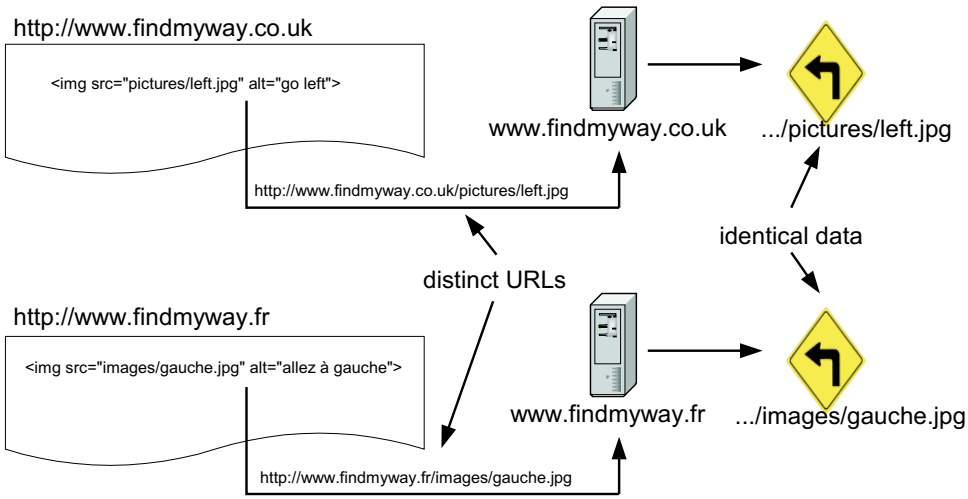


Figure 4.6: *Distinct static URLs/identical content phenomenon. Conventional caching schemes do not recognize that the content, two distinct URLS are referring to, is in fact identical.*

the resident proxy (RP) on a reachable local service point (LSP). The RP forwards the request ③ unchanged towards the central proxy, which in turn forwards ④ the request towards the server application (SA). If no LSP would be in reach the RP is omitted and the request is forwarded via the PLMN and the appropriate gateways towards the central proxy directly.

The server application receives ⑤ and parses the request, generates the response Resp_1 with the requested data and starts to send it towards the CP. Typically there is no need to include ID_{WID} into the response, since all communication between are connection-oriented. Therefore the response is associated with the corresponding request automatically[9].

The central proxy waits until the complete response has arrived ⑥. It then computes the message digest of the data included in the response. For each WID that is served by the CP its keeps a list containing the message digests of all response data that has been sent towards the particular device. If the data has not been sent towards the WID before, its digest is not found in the list. In this case the CP includes the message digest into the list and into response header and sends the complete response to the resident proxy.

The resident proxy does not perform any operation on the data or on the headers. It starts immediately ⑦ to forward the data streaming from the CP towards the

[9]This is the case for the common practice to transport HTTP traffic over TCP. Nevertheless this is note mandatory as HTTP can also be transported over connection-less (e.g. UDP) or message-oriented (e.g. e-mail) channels between proxies. In this case the proposed scheme requires to include information identifying the WID into the responses as well.

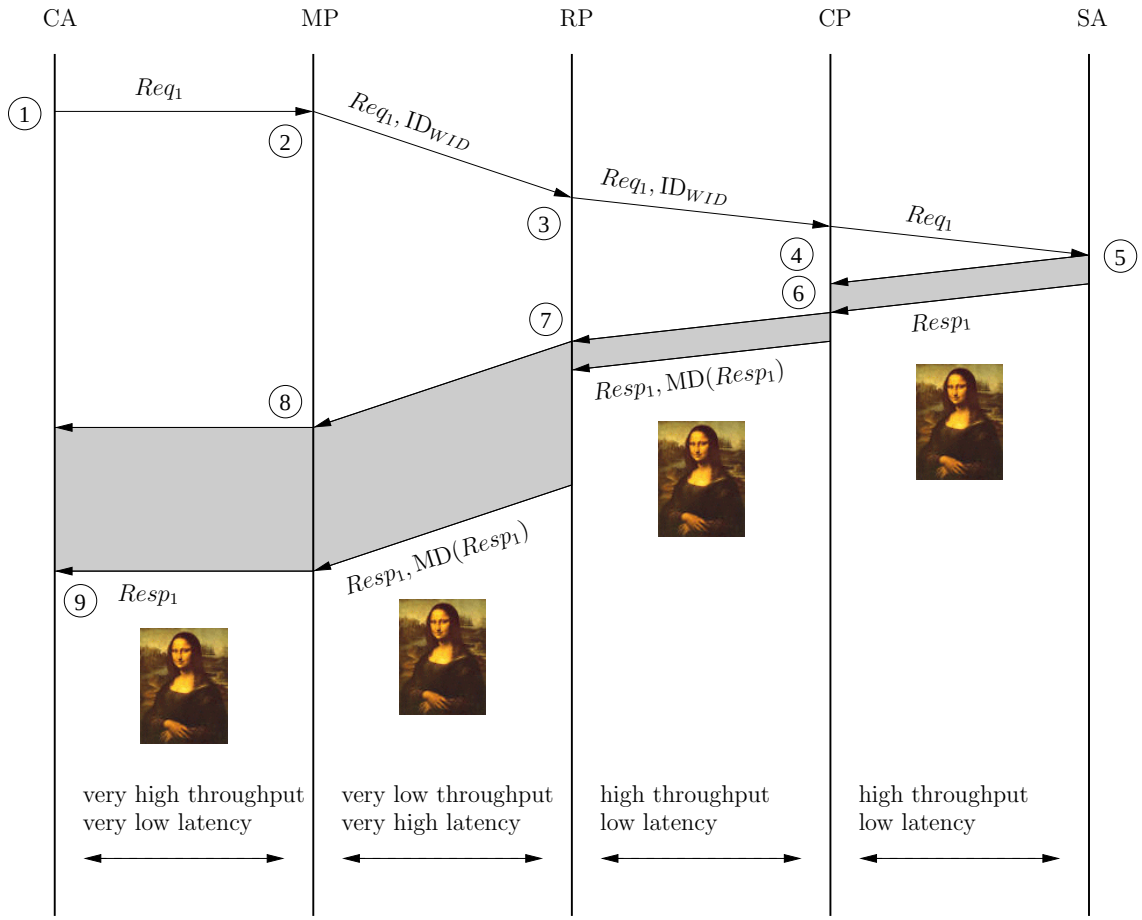


Figure 4.7: *Illustration of proposed protocol for cache consistency and dynamic content (cache miss). A message digest of the response's content is computed by the central proxy (CP), included into the response header and stored in a list at the CP.*

MP. The communication between the RP and the MP is slow as has been mentioned before. Therefore, the data arriving from the CP has to be queued. Upon arrival ⑧ of the initial bytes of the response at the mobile proxy, it starts to forward them towards the client application. This transfer is likely to be the fastest in the chain, due to the fact that it is transported by inter-process communication within a device. Therefore, no queues are likely to be built up. The included message digest is stored in a table together with the response data for potential later reuse. This constitutes the actual caching operation. After that the complete response is forwarded ⑨ to the client application. The CA can present the results to the user or respectively parse the document description for references pointing to embedded objects.

In the following we illustrate how the previously invested efforts, i.e. computing of message digests and caching of responses is employed to reduce the communication load (see Fig. 4.8). At a later instant another request shall be issued by the CA. The steps ① to ④ are identical to the previous case. This time it happens that

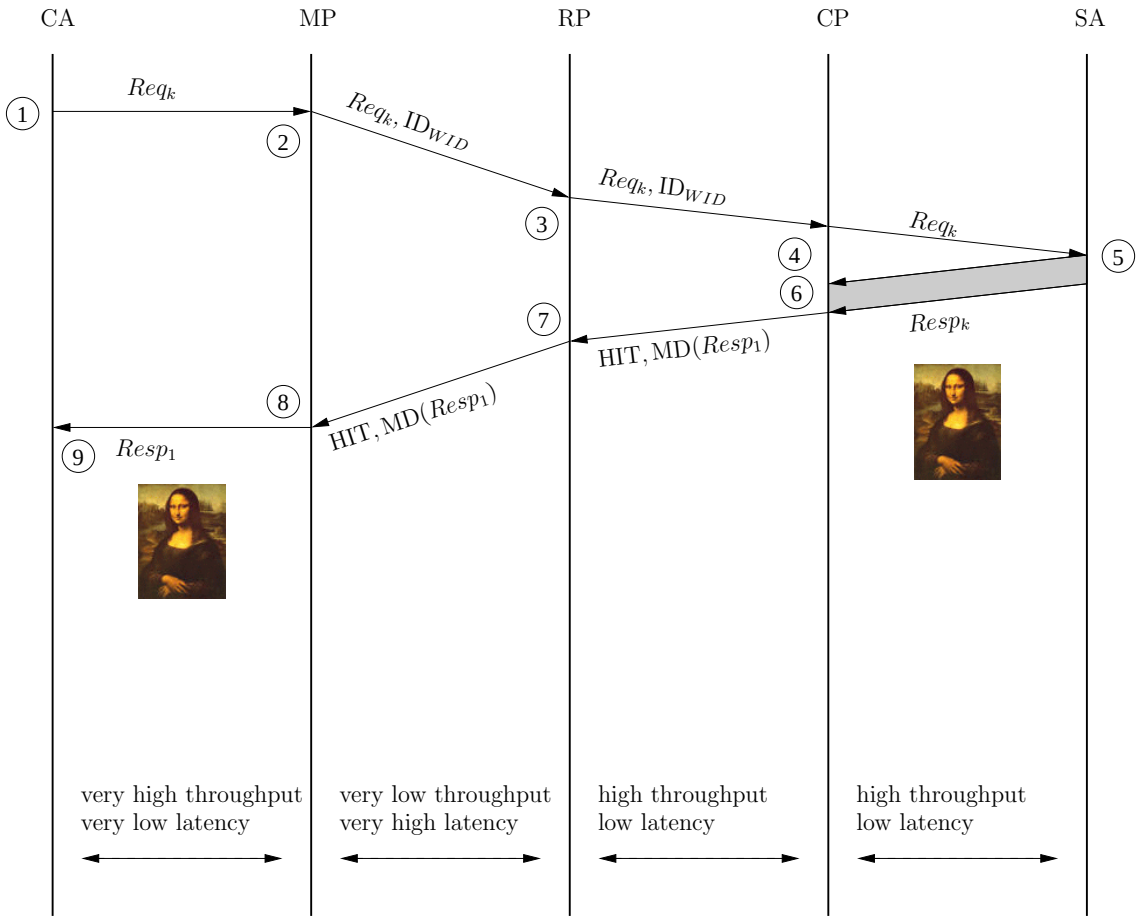


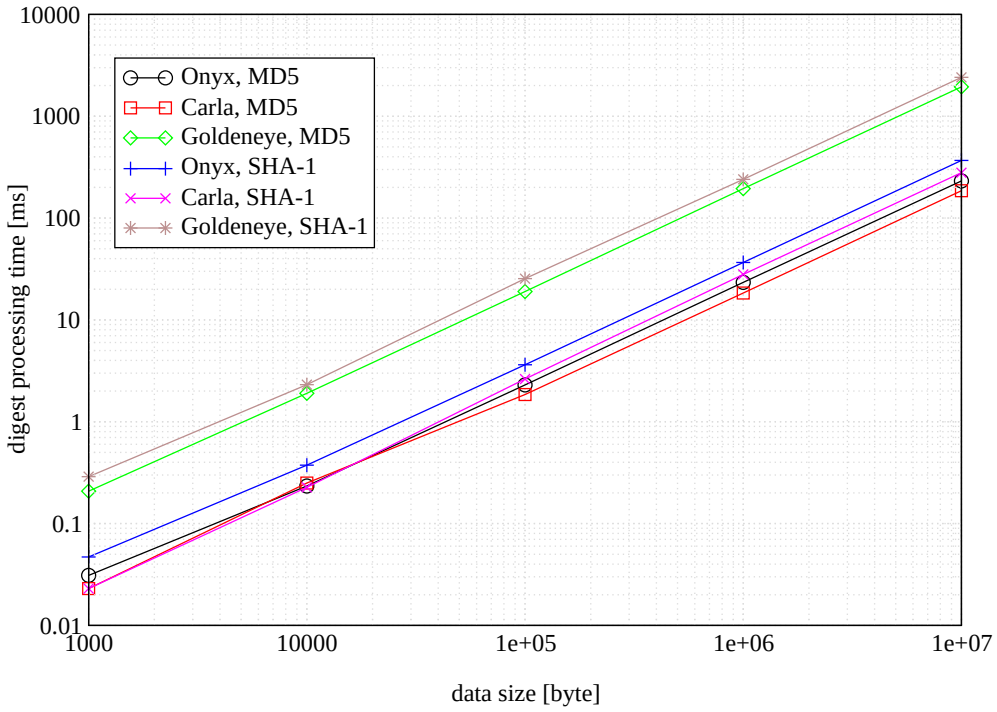
Figure 4.8: *Illustration of proposed protocol for cache consistency and dynamic content (cache hit). Whenever the message digest of the response's content computed by the central proxy (CP) is found in its list of MDs, a cache hit is detected and signalled in the response header. The actual content is not transferred, resulting in a significant reduction of perceived latency and reduced network load.*

the data included in the response is an exact copy, i.e. the same picture, of the data already transported towards the WID. This circumstance is detected, since the message digest is computed and is found in the list held for the particular WID at the central proxy. The CP then creates a response with a header signalling the fact that the data can be found on the WID (HIT) and the message digest. This short response is sent ⑥, forwarded ⑦ and received at the ⑧ at the mobile proxy. The MP uses the message digest as a key to retrieve the data from its cache and delivers it to the client application where the content is presented ⑨.

It should be emphasized that no cooperation from server applications is needed for deploying this scheme.

A finite probability exists for the proposed protocol to cause an error due to the fact that two distinct documents can lead to the same digest (i.e. masking of a desired documents by an old having the same message digest). However, the MD5 algorithm is, due to the fact that its main field of application is cryptography, designed to produce equally distributed random-like digests. In RFC 1321 it is conjectured that "the difficulty of coming up with two messages having the same message digest is in the order of 2^{64} operations, and that the difficulty of coming up with any message having a given message digest in on the order of 2^{128} operations".

Furthermore, we have to consider the fact that computing the message digest will consume resources and time and thus introduce additional latency. In order to assess this unwanted additional latency we have performed trials for two different message digest algorithms (MD5 and SHA-1) on three different platforms. We are especially interested in the type of increase (linear, polynomial, exponential, ...) of computation time over message length. The results are depicted in Fig. 4.9 and show moderate computation times with only linear increase for increasing message sizes. We can see that MD5 performs marginally faster than SHA-1 for all platforms. This is partially due to the shorter message digest size for MD5. The results permit the conclusion that the delays introduced by the computation time for any given message length will be significantly smaller than the delays caused by the limited transmission data rate. We therefore consider it advisable to employ the described scheme in cache validation especially for dynamic content generation and service provisioning to mobile devices. Without any pronounced preference we have chosen to the MD5 as the default message digest algorithm within our proposed protocol due to its sufficient length, slightly faster computation time and availability for a large number of platforms.



host	CPU	Memory	Operating System
onyx	Pentium-IV – 2 GHz	512 MByte	MS Windows 2000
carla	Pentium-IV – 2 GHz	512 MByte	Linux 2.4
goldeneye	UltraSPARC-IIe – 500 MHz	2048 MByte	SunOS 5.8

Figure 4.9: *Duration of message digest computation for MD5 and SHA-1 on distinct hardware platforms and operating systems.*

4.3 Software Development, Integration and Test

In order to demonstrate the practicability and the benefits of the proposed system architecture appropriate software components have been selected or developed, integrated and tested.

Several commercial and open-source implementations of server applications (SA), such as Microsoft Internet Information Server (IIS) or Apache HTTP Server exist. Since our proposed architecture employs the standard HTTP 1.1 protocol to communicate with these SAs, no need for changes exist. Any content provided by these SAs can be accessed.

Mobile devices usually have client applications (CA) already embedded within their operating system, such as Microsoft CE or Symbian OS.

While several implementations of proxies exist in various languages (C/C++, Java, Python etc.), none of them fulfilled our requirements, such as the use of multiple networks, prefetching, cache consistency, transportation of information regarding symptoms or consequences and most importantly, efficient execution on currently available mobile devices. Hence, the proxy components were developed from scratch. Many functionalities are necessary on all three types of proxies (MP, RP, CP). Since all three proxies are typically execute on different host platforms with different operating systems, the Java programming language was chosen for its platform independency. This platform independency is achieved by abstracting the operating systems' functionalities within the concept of a virtual machine which provides an identical execution environment and application programmer interfaces (APIs) for all platforms.

The platform on which all three types of proxies execute are listed in Table 4.1.

	WID (MP)	**LSP (RP)**	**CP Host (CP)**
CPU	ARM9, 156 MHz	VIA EDEN, 667 MHz	UltraSparc II, 440 MHz
memory	12 Mbyte	512 Mbyte	1024 Mbyte
radio interfaces	GPRS, (HS)CSD, Bluetooth	GPRS, (HS)CSD, Bluetooth, WLAN	–
other interfaces	–	Ethernet	Ethernet
OS	Symbian v7.0	Linux 2.4	SunOS v5.8
Java (version)	Personal Java (Java 1.1.6)	J2SE (Java 1.4)	J2SE (Java 1.4)

Table 4.1: *Platform characteristics for mobile proxy (MP), resident proxy (RP) and central proxy (CP). Since the Personal Java version on the mobile device corresponds only to Java version 1.1.6, all methods that are to be used on the mobile device must be conform with this version.*

The tightest restrictions among the virtual machines are imposed by Java 1.1.6, which does not allow the usage of several language constructs present in the more

recent version of Java 1.4. However, these restrictions are modest and constitute no major hindrance.

An additional hurdle is the lack of Bluetooth support in currently available versions of Personal Java[10]. Due to this lack it was necessary to implement thin adapters between the Bluetooth stack and the virtual machine in C++ for Symbian (mobile device) and Linux (LSP).

During the design and implementation of distributed software instances several peculiarities have to be considered that do not occur in conventional stand-alone applications.

Firstly, the distributed instances collaborate with each other by some (unreliable) network resource. In order to provide continuous service without the need for restarting or administrative interference, they have to achieve a level of fault tolerance that allows them to continue their execution after errors have occurred.

Secondly, each application may be accessed by multiple other entities concurrently. In consequence, each process consists of dozens of threads that execute in parallel within the same memory space, in our case the same virtual machine. Since frequently resources are accessed from multiple threads it is necessary to use semaphores to synchronize the threads' access to these resources. Common pitfalls that arise from this fact are deadlocks. They occur when two threads have locks on different resources and wait for each other to release the lock on the other resource. In this case both threads wait for ever or until the execution of the process is stopped.

Such deadlocks are often very hard to reproduce and may occur e.g. only once within a day of continuous execution due to a certain timing constellation, which makes their detection and elimination problematic.

Due to the risk of deadlocks and the intricateness of on-device debugging, the importance of a careful design of the software architecture and protocols increases. With this in mind, the software architecture of all three proxy types (MP, RP, CP) and the situation inference engine was modelled, designed and implemented with a software development tool that allowed simultaneous roundtrip software engineering, i.e. the software is simultaneously represented in the programming language (Java in our case) and the unified modelling language (UML) [RJB99, Oes99]. Changes in the program code are automatically represented in the UML model and vice versa. Parts of the interaction between the client application, the proxies and the server application have been modelled with the graphical representation (GR) of the specification and description language (SDL) [BHS91]. SDL was mainly used for clarifying and discussing protocol issues but refrained from using formal testing methods based on SDL. This approach has proved especially valuable to keep an overview on the large number of case differentiations each component has to consider during its interaction with the other components but relieved us from the tedious task of modelling the overall system. The concise notation of SDL is also indispensable when asking questions

[10]Standardization efforts for Bluetooth support in Java have just recently led to commercially available devices [JSR]. However, so far, these devices do not fulfill other requirements concerning memory or CPU performance.

like “what happens if the connection is interrupted?”, “what happens if the URL is malformed?” or “what happens if the browser does not support compression?” in order to achieve the necessary amount of fault tolerance in a distributed system.

Integration and testing on the real target devices has been performed as early and carefully as possible. For this purpose an infrastructure of several local service points (LSPs) and mobile devices has been set up in our laboratory, facilitating convenient integration and testing in addition to the testbed fielded in the town of Landsberg am Lech, Germany. The mobile device and the LSP hardware are depicted in Figs. 4.10 and 4.11.

Figure 4.10: *The primary type of wireless information device used within the testbed is a common-off-the-shelf Sony Ericsson P800. A mobile proxy (MP) on the device selects between the short range network (Bluetooth) and the mobile network (GSM-GPRS).*

The developed software has reached a state of considerable maturity and is operational in the Landsberg testbed.

It cannot be emphasized enough, that the software suite, as well as the procedures for its efficient deployment, has been achieved through the consolidated efforts of the complete research team!

4.3.1 Performance Measurements

Every additional element in a communication path introduces additional delay and is likely to reduce the throughput. Of course, the ability to perform prefetching and

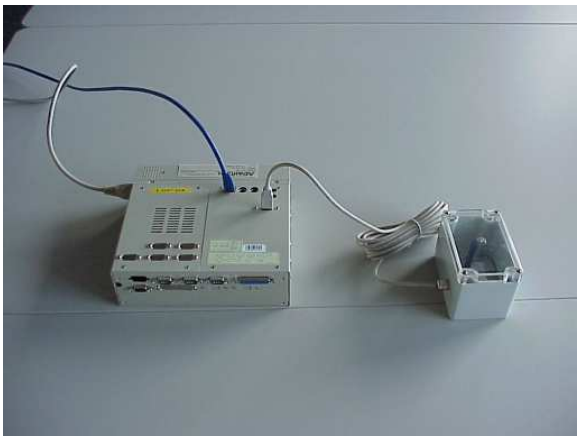


Figure 4.11: *Local Service Point (LSP) with Bluetooth transceiver module. Up to four detached Bluetooth transceiver modules can be connected by USB (universal serial bus). The LSP has both IEEE 802.11 (WLAN) and Ethernet interfaces to connect to the intranet/internet.*

to use the short range network increases the overall perceived performance. Nevertheless, an important focus in the design and implementation of the proxy software is the reduction of adverse effects of individual proxies as far as possible by applying several techniques of software performance maximization, such as optimized memory allocation, reuse of objects or thread pooling.

In order to evaluate the success of these efforts, throughput and delays of HTTP-traffic via Bluetooth and GPRS have been measured for several configurations.
Fig. 4.12 shows the obtained results for throughput.

For all measurements the configuration is noted beneath or above the results. For each case 10 measurements have been performed. Each measurement is represented by a circle in the plot. The average value of the measurement is represented by a triangle.
The first measurement is performed with the purpose to evaluate the throughput of device-internal socket communication (**MP→CA**) between the processes of the mobile proxy (MP) and a client application (CA). No radio interface is involved in the communication path for this measurement. Instead, the measurement is an indication how fast the mobile proxy can "hand out the data" to the client application. The average throughput from the MP to a CA is $2.49 \cdot 10^6$ bit/s.

With the second measurements the throughput from the resident proxy (RP) via the Bluetooth interface and the MP to the CA (**RP** $\xrightarrow{\mathbf{BT}}$ **MP→CA**) is determined. Since in this case the Bluetooth radio interface is the bottleneck, the measured result indicates the maximum throughput we can achieve via Bluetooth. The average throughput from the RP via Bluetooth and the MP to the CA is $3.11 \cdot 10^5$ bit/s. Since in this case the content resides on the LSP, transfer via GPRS is not considered.

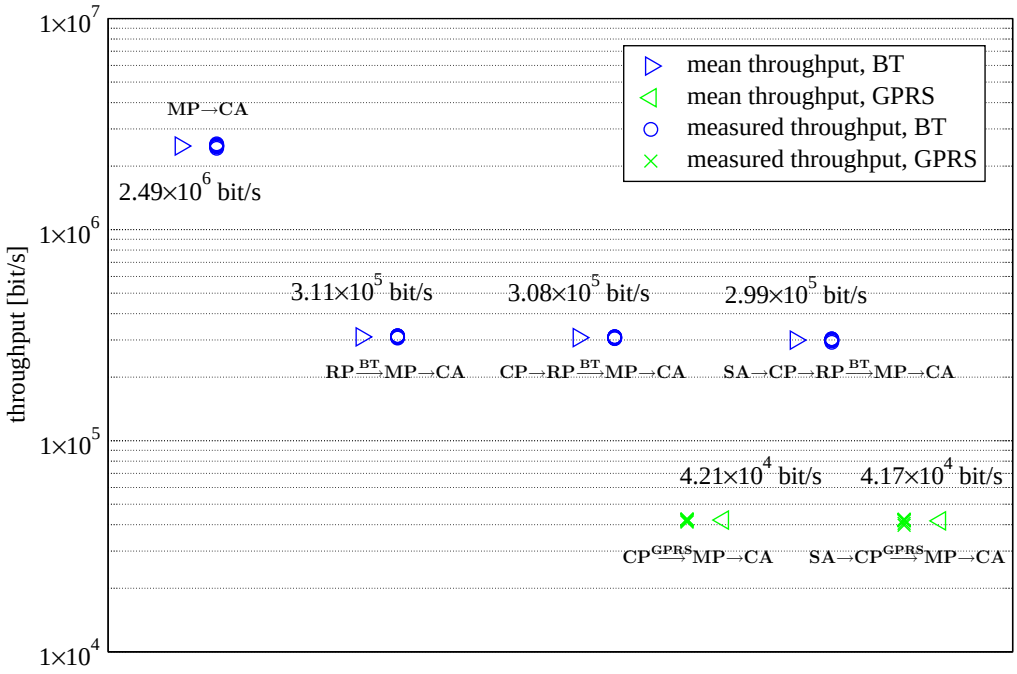


Figure 4.12: *Measured throughput for data transfer via Bluetooth and GPRS for various configurations. The configuration is noted beneath or above the results. In the first case the MP is the content source. Hence, the achieved throughput of* $2.49 \cdot 10^6$ *bit/s is the throughput achieved for device-internal socket-communication from the MP to the CA (client application); in this case no radio interface is in the path. In the other cases the content source is the RP, CP or SA (server application) and Bluetooth or GPRS are in the communication path. For each case 10 measurements have been performed, each measurement corresponds to a circle; triangles indicate the averaged values.*

In the third case the central proxy (CP) is the source of the data. The throughput from the CP via the RP via Bluetooth to the MP and finally to the client application (**CP→RP** $\xrightarrow{\textbf{BT}}$**MP→CA**) is only marginally reduced to $3.08 \cdot 10^5$ bit/s. Hence, this additional hop[11] has almost no influence due to the dominance of the Bluetooth bottleneck.

If the data on the CP is accessed via GPRS the resulting throughput from the CP via GPRS to the MP and the CA (**CP** $\xrightarrow{\textbf{GPRS}}$**MP→CA**) is $4.21 \cdot 10^4$ bit/s. Here the GPRS link is the dominant bottleneck. The Bluetooth link and the RP are not involved.

When the SA is the data source the throughput from the SA via the CP via the RP via Bluetooth to the MP and finally to the CA (**SA→CP→RP** $\xrightarrow{\textbf{BT}}$**MP→CA**) is $2.99 \cdot 10^5$ bit/s, which is again only a marginal reduction.

[11] Here "hop" is also a hop in the IP layer but more importantly on the TCP and on the HTTP layer, since both TCP and HTTP terminate at the proxies. (cf. Fig. 4.4)

If GPRS is used the throughput from the SA via the CP via GPRS to the MP and then to the CA (**SA→CP** $\overset{\text{GPRS}}{\longrightarrow}$ **MP→CA**) is also only marginally reduced to $4.17 \cdot 10^4$ bit/s.

While we have seen that the addition of the proxies into the communication path have only a marginal influence on the throughput, it is conjectured that since each proxy has to parse the HTTP headers of the requests this will cause additional delays. Measurements of the delays have been performed in the same sequence as for the throughput discussed above. The results are depicted in Fig. 4.13 and discussed briefly.

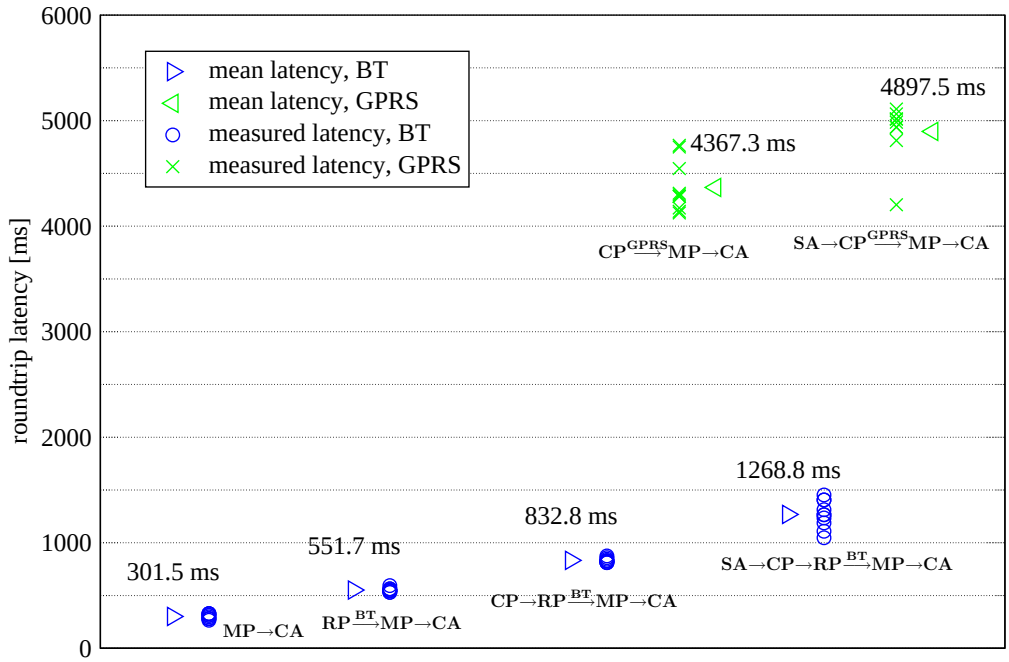


Figure 4.13: *Measured roundtrip latency for data transfer via Bluetooth and GPRS for various configurations. The configuration is noted beneath or above the results. In the first case the MP is the content source. Hence, the resulting delay of* 301 *ms is caused by device-internal socket-communication from the MP to the CA (client application); in this case no radio interface is in the path. In the other cases the data source is the RP, CP or SA (server application) and Bluetooth or GPRS are in the communication path. For each case 10 measurements have been performed, each measurement corresponds to a circle; triangles indicate the averaged values.*

For the first measurement the source of the data is the mobile proxy (MP) on the device. The measured roundtrip latency of 301 ms, i.e. the duration until the first byte of the response arrives at the client application (CA), is therefore mainly caused by device-internal socket-communication from the MP to the CA (**MP→CA**) and by the necessary time the MP needs to parse the HTTP-header of the request.

If the RP is the data source the data is transported via Bluetooth. For this case ($\mathbf{RP} \xrightarrow{\mathbf{BT}} \mathbf{MP} \rightarrow \mathbf{CA}$), the roundtrip latency is 551.7 ms.

When the central proxy (CP) is the data source and the transport is performed using the short range network ($\mathbf{CP} \rightarrow \mathbf{RP} \xrightarrow{\mathbf{BT}} \mathbf{MP} \rightarrow \mathbf{CA}$) the overall roundtrip latency is further increased to 832.8 ms.

If the mobile network is employed the resulting latency is 4367.3 ms, which is significantly longer, despite the fact that one proxy less is in the communication path ($\mathbf{CP} \xrightarrow{\mathbf{GPRS}} \mathbf{MP} \rightarrow \mathbf{CA}$).

If the server application (SA) is the data source the transport via the short range networks passes through all three proxies ($\mathbf{SA} \rightarrow \mathbf{CP} \rightarrow \mathbf{RP} \xrightarrow{\mathbf{BT}} \mathbf{MP} \rightarrow \mathbf{CA}$) which results in a roundtrip latency of 1268.8 ms, which is still nearly four times less than 4897.5 ms, which are incurred if GPRS is involved ($\mathbf{SA} \rightarrow \mathbf{CP} \xrightarrow{\mathbf{GPRS}} \mathbf{MP} \rightarrow \mathbf{CA}$).

Since the short range network resource may be shared by multiple users participating in a piconet, these users compete for the use of the communication channel. In the Bluetooth standard access to the radio resource within a piconet is centrally managed by a master device, which, is the local service point (LSP) for our case. In order to evaluate how effectively this medium access control scheme performs, we have measured the throughput for one to seven slave devices.
Fig. 4.14 shows the setup for the measurements.

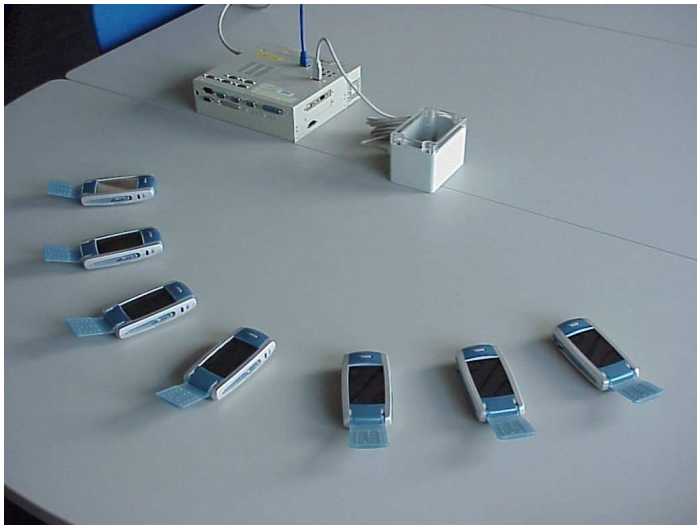


Figure 4.14: *Setup for measuring the throughput of of up to seven concurrent users in Bluetooth piconet.*

Trials for 1 to 7 slave devices concurrently participating in a piconet have been performed. The throughput for each device participating in a trial has been measured. To identify the influence of statistical fluctuations three sets of trials have been performed, resulting in a sum of $7 \times 3 = 21$ trials and $(1+2+3+4+5+6+7) \times 3 = 84$ measurements.

cc

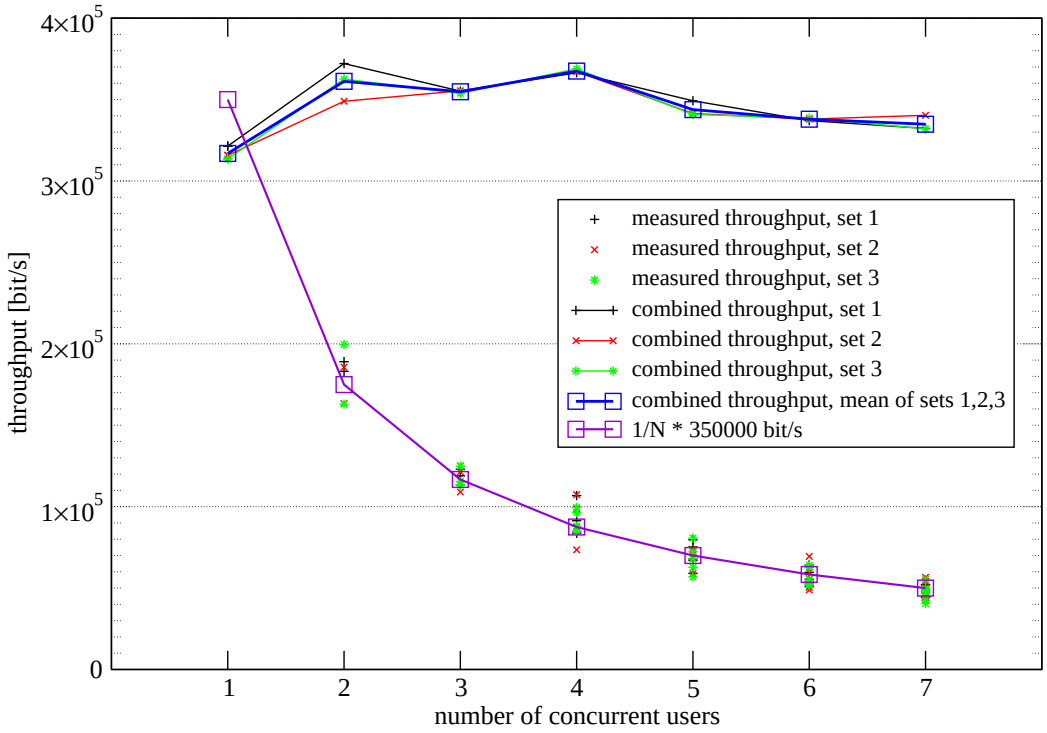


Figure 4.15: *Measurement results for multi-user throughput. Three sets of measurements are depicted, with each set comprised of a sequence of measurements from one to seven concurrent users.*

We see that the combined throughput is fairly independent of the number of devices and approximately $3.5 \cdot 10^5$ bit/s. Comparing the curve for $(1/N) \cdot 3.5 \cdot 10^5$ bit/s with the measured throughput for the individual devices we see a close correspondence. Hence, me may conclude that Bluetooth's approach to manage the access to the medium of a piconet by a master node results in a fairly efficient use of the radio resource.

The measured values of the throughput and delay for Bluetooth and GPRS have been used to parameterize the simulations described in the previous chapter. These simulations have shown that, despite these additional delays, the perceived performance of the system is significantly improved if situation aware prefetching is applied. In consequence, we conclude from the measurements and the simulations that the addition of the proxies is justified, since it enables both situation aware prefetching and the use of the short range network, which results in a significant increased perceived performance of the overall system.

4.4 Deployment and Initial Operational Experiences

An experimental and demonstration testbed centered around a mobile city information system for tourists has been deployed in the town Landsberg am Lech, close to Munich. The system uses local service points (LSPs) for cost efficient content delivery and user localization. The initial setup consisted of 11 LSPs located at landmarks and places of interest distributed across the town center of Landsberg. The locations of LSPs are depicted in Fig. 4.16.

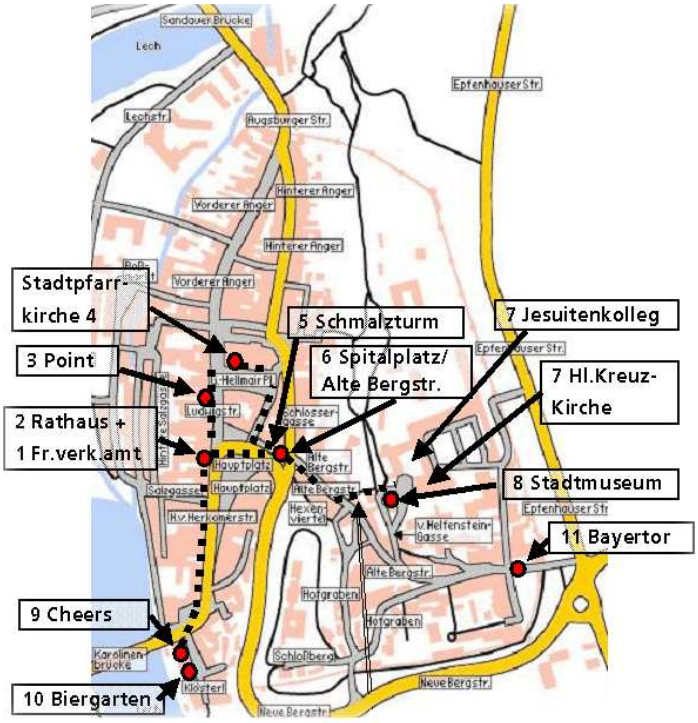


Figure 4.16: *Locations of local service points (LSPs) in the Landsberg testbed. All LSPs incorporate a resident proxy (RP). Several LSPs (e.g. Rathaus, Fremdenverkehrsamt, Schmalzturm, Spitalplatz, Stadtmuseum) are connected via DSL (digital subscriber line) or WLAN and VPN (virtual private network) to the project's intranet in which the central proxy (CP) with access to the internet resides. Hence, short range access to the world wide web is available at these positions. Some LSPs (e.g. Stadtpfarrkirche) are not connected and provide local content by local server applications.*

Bluetooth is used as the short range radio standard between LSP and mobile devices. The limited range of up to 50 meters in combination with the geographic distribution of the LSPs results in a partial coverage of central Landsberg. GSM-GPRS coverage is available within the complete town center, resulting in a situation

well modelled by the heterogeneous network with coverage gaps scenario described in Section 3.1.1.

All mobile terminals participating in the testbed are equipped with Bluetooth and GSM-GPRS. The devices' standard browser is configured to use a mobile proxy (MP) (residing on the device). The MP chooses between mobile network and short range networks. Furthermore, it performs situation aware prefetching, stores and serves the prefetched hypertext elements to the browser application. Whenever the presence of a particular LSP is sensed by detecting its Bluetooth signal, a map with the user's current position is displayed, based on the inherent location information.

Most of the deployed LSPs are connected to the project's intranet either directly via DSL (digital subscriber line) and VPN (virtual private network) or via a wireless local area network (WLAN) and WLAN-access routers, which in turn are connected via DSL and VPN to the intranet. All connected LSPs are equipped with resident proxies (RPs) which connect to a central proxy (CP) residing within the intranet and having access to the global internet. Hence, access to the world wide web is provided. Some LSPs have no connection to the intranet. These LSPs are examples of the configuration depicted in Fig. 4.3. They are equipped with resident proxies and local server applications which facilitate them to provide local content.

Remote maintenance of the testbed infrastructure in Landsberg is usually possible, since most components such as LSPs and routers reside in a VPN and are accessible from our intranet or via the PSTN (public switched telephony network). Typical maintenance tasks are updates of the operating system, the proxy software or local content[12].

To enable quick updates of unconnected LSPs, software and content can be uploaded from the mobile devices via Bluetooth. The upload functionality is hidden from users and protected by password[13].

We have made the interesting observation that whenever Bluetooth coverage is available, usage of the hypertext system significantly increases. We believe that this is partly caused by a "threshold effect" in terms of system response time. When user have to wait too long for system responses, they become distracted from their initially made requests. The low latency experienced in regions of Bluetooth coverage prevents this negative and leads to increased usage.

So far, users of the testbed do not have to pay for GPRS usage themselves. However, it seems that the knowledge of the "free-of-charge" service during Bluetooth coverage leads users to adopt a relaxed attitude during usage, compared to our previous observation of users of e.g. WAP services with both time and volume charged

[12]Only in rare cases it is necessary to physically access the LSPs, e.g. low-level driver problems of the Bluetooth transceivers necessitated the physical removal and reconnection of these interfaces.

[13]Initial experiments to connect via Bluetooth from one connected LSP to several otherwise unconnected LSPs for remote maintenance have been successful. Even if the multihop-protocol is not capable of carrying the user traffic, remote maintenance is very desirable since it enables content updates at very low cost.

tariffs. Users frequently start to "'surf" the hypertext system similar to typical usage patterns found in settings in which the World Wide Web is accessed from desktop computers.

The amount of traffic caused by the activity of test users over the short range network would often be literally unaffordable if transported and charged according to today's tariffs for conventional mobile data services.

One of the most noticeable observations is the considerable extension of the devices' battery stamina. When hypertext activity is mostly performed during short range coverage, the batteries of the mobile devices last several times longer compared to the case in which all traffic is transported via GPRS[14].

So far, the decision to base the reference implementation on Java, Symbian and Bluetooth seems to be a fortunate approach, due to the increasing proliferation of this combination on mobile phones. A recent trial showed that our reference implementation executes and performs properly on a commercial phone with UMTS functionality (Motorola A 920 with Symbian OS), without changing a single line of code.

[14]Initial measurements have shown that the same amount of traffic transported via Bluetooth only results in approximately one tenths of the power consumption incurred if GPRS is used.

Chapter 5

Conclusions and Outlook

5.1 Conclusions

In this thesis new concepts, insights and results have been presented that have been developed in the field of situation awareness and its application in prefetching for improving mobile information access in heterogeneous wireless networks.

A novel situation model has been formulated and discussed from an information theoretic perspective. The task of obtaining and continuously adjusting suitable probabilities for the model has been formally treated as an estimation problem.

In this work the target application has been prefetching of hypermedia documents. Nevertheless, the discussion of the situation model has intentionally been kept as generic as possible in order to enable its application to other domains, such as handover decisions, proactive computing or future user interfaces to search engines.

A thorough analytical investigation of prefetching in hypertext systems has been performed, yielding new qualitative and quantitative insights into the effects of prefetching on the average waiting time and average transported data volume. The analysis led to the conclusion that the documents' probabilities are the sole criterion for selecting prefetching candidates. Furthermore, an optimum threshold probability has been derived and its relation to a user policy has been discussed.

The investigation has been further extended by means of simulations towards various mobile networking scenarios. For this purpose a novel mobility model has been developed and has been used in conjunction with models for network topology and traffic to obtain insight on the influence of situation aware prefetching in both heterogeneous and hybrid network scenarios.

The simulation results have shown the considerable benefits of situation aware prefetching in the various scenarios. It has been shown that prefetching is especially advantageous in a heterogeneous wireless network consisting of a short range network with properties of Bluetooth and a mobile network with properties of GSM-GPRS.

The ability to prefetch content constitutes an additional degree of freedom in the optimization of the usage of network resources. Both the theoretical analysis as well as the simulation have shown that specifically the probability threshold of prefetching is the parameter to adjust this degree of freedom to fulfill a given user policy.

5.2 Outlook

Many new insights have been obtained during our studies. However, new questions have evolved and a large set of new ideas has been spawned.

The interpretation of prefetching and the related threshold probabilities as an additional degree of freedom has particularly interesting implications and possibilities of employing this degree of freedom for dynamic pricing of network resource usage, in order to maximize network utilization and user benefit. Since network traffic becomes "elastic" when prefetching is applied, network usage may be stimulated and discouraged depending on current overall demand. This gives rise to a closed-loop control scheme. An initial outline of this concept can be found in Appendix A.

The model of situation awareness developed in this thesis has been designed for applications in pervasive and ubiquitous computing far beyond the prefetching techniques that have been investigated as an application example in this work. For this reason the extent of the model and the theoretical analysis performed in Chapter 2 far exceeds the requirements for mere prefetching and provides a solid foundation for the development and analysis of future services that achieve pro-activity based on context information.

The continuing operation of the Landsberg testbed provides the opportunity to apply the theoretical models for situation awareness in real world applications and user tests. Due to the increasing availability of context sensors on devices as well as in the environment, it will become possible to interpret and use this sensor data as symptoms and aspects within the situation model and perform experimental work on situation awareness in a multitude of mobile applications and services.

Appendix A

Dynamic Pricing for Demand Control

Any economical system shows fundamental relations between the amount available but limited resources, the demand for these resources and the price to pay for them. If we want to understand the way these relations influence our technical system we have to understand their underlying mechanisms and try to synthesize a simplified model. Mapped to our problem domain we classify the actors in our economical system into two categories: consumers and providers[1]. Our limited resource shall be the transport capacity of the network infrastructure.

We are interested in the short term (seconds to hours) dynamic behavior of the system. Therefore, we may consider the amount of installed network infrastructure to be constant. We assume that for each network element providing transport capacity an optimum amount of demand exists that constitutes an optimum operating point from the provider's perspective that maximizes some target variable (e.g. revenue, consumer satisfaction). The provider can adjust the price for the service by any law or *provider policy* he deems appropriate.

The consumer behavior will be influenced by the price to be paid for the data transport service. Low prices will lead to increased demand, whereas high prices will lead to users refraining from or delaying their demand, depending on their individual *consumer policy*. If the provider policy takes the actual combined demand on the network resource into account for adjusting the price, the overall system constitutes a feedback loop. Measuring the difference between the optimum operating point demand and the actual demand enables the provider to use the feedback character of the system to create a control system with the intention to stabilize the system's state close to the optimum operating point.

In today's operational mobile communication systems this control is realized by monitoring network wide demand and utilization and rarely (e.g. monthly) adjusting tariffs and distributing these tariffs to the to the consumer. A crude form of control

[1]This is a course simplification, since the industries of information technology and telecommunications are famous for their complex value chains (or rather meshes). Nevertheless, for our purpose it is sufficient to subsume operators, providers, integrators, vendors etc. into the category of providers

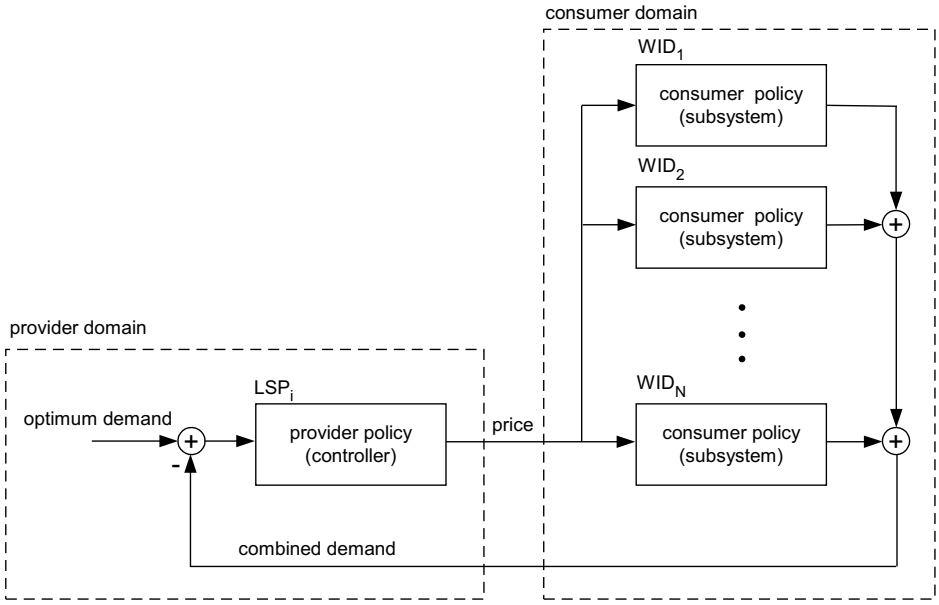


Figure A.1: *Control structure for dynamic pricing scheme*

on a time scale of hours is realized by offering fixed reduced prices at fixed times of low demand (weekends, at night) and increased prices at times of high demand (weekdays, business hours) in order to smooth demand. This system seems to work quite well with today's mobile networks. The fairly large cell sizes lead to considerable smoothing of demand in conjunction with the dominance of voice traffic with far less burst traffic than e.g. web browsing do not necessarily require any more advanced control. In contrast to these conditions the scenarios under investigation in this work have heterogeneous radio access technologies with large cell sizes from the PLMNs and very small coverage areas from the short-range communication technologies. While these small cells are necessary for providing high capacity, they are subject to strong and rapid fluctuations of demand and thus resource utilization. From this results the fact that resources are under-utilized most of the time, while peeks in demand can quickly result in blocking or congestion. In these cases the smoothing influence of an improved control scheme becomes therefore highly desirable. The control should be improved in three aspects: a) it should be faster, i.e. have time-constants in the range of seconds; b) it should be working on a local basis, i.e. controlling the demand of individual network resources (e.g. LSPs); c) it should enable users to manage their costs without interaction for each individual request. Therefore the price should be signalled to the user unobtrusively. The user should be able to set a form of policy that governs the demand his activity will produce.

A control structure as depicted Fig. A.1 is suggested to realize these requirements.

For each LSP the combined demand on its network resources is measured and

compared against the desired optimum demand. The difference is used as input for a controller that essentially implements the provider policy. The resulting price is announced to all mobile devices in range. A process on the device, in our architecture the mobile proxy, uses this information and shapes demand according to the policy set by the consumer[2]. The individual behavior of all WIDs in range results in the combined demand for the network resources of the LSP. The feedback loop is closed.

Neither the number of other WIDs serviced by the same LSP, nor their policy is exactly known to a particular WID. Therefore the future price is considered to be random variable from the perspective of any optimizing entity associated with an individual WID. Estimation techniques are necessary to predict its future values.

[2]Additionally the current price may be signalled (by sound or preferably less obtrusively by color or icon) to the user

Appendix B

Fast Generation of High-Dimensional Uniform Probability Vectors

For simulation purposes an algorithm is required to generate uniformly distributed $N-$dimensional probability vectors $\mathbf{p} = [p_1, \ldots, p_N]^T$. In the simplest algorithm a set of $N-1$ numbers are drawn from a uniform distribution $f_z(z)$. If their sum exceeds 1.0, the set is rejected, else the set is accepted and $p_N = \sum_1^{N-1} p_i$. While this algorithm works well for low dimensions ($N = 2, 3, 4$), it suffers from the rapidly increasing rejection rate for higher dimensionality.

In contrast, the following algorithm facilitates the required fast generation of high-dimensional uniform probability vectors:

1. generate $N-1$ numbers z_i from uniform distribution $f_z(z)$.

$$f_z(z) = \begin{cases} 1 & \text{for} \quad 0 \leq z \leq 1 \\ 0 & \text{elsewhere} \end{cases} \tag{B.1}$$

2. sort the numbers z_i into numbers z_i' with ascending order.

$$0 \leq z_i' \leq z_{1+1}' \leq 1 \tag{B.2}$$

3. compute N differences and assign to probabilities p_i.

$$p_i = \begin{cases} z_1' & : \quad i = 1 \\ z_i' - z_{i-1}' & : \quad 1 < i < N \\ 1 - z_{N-1}' & : \quad i = N \end{cases} \tag{B.3}$$

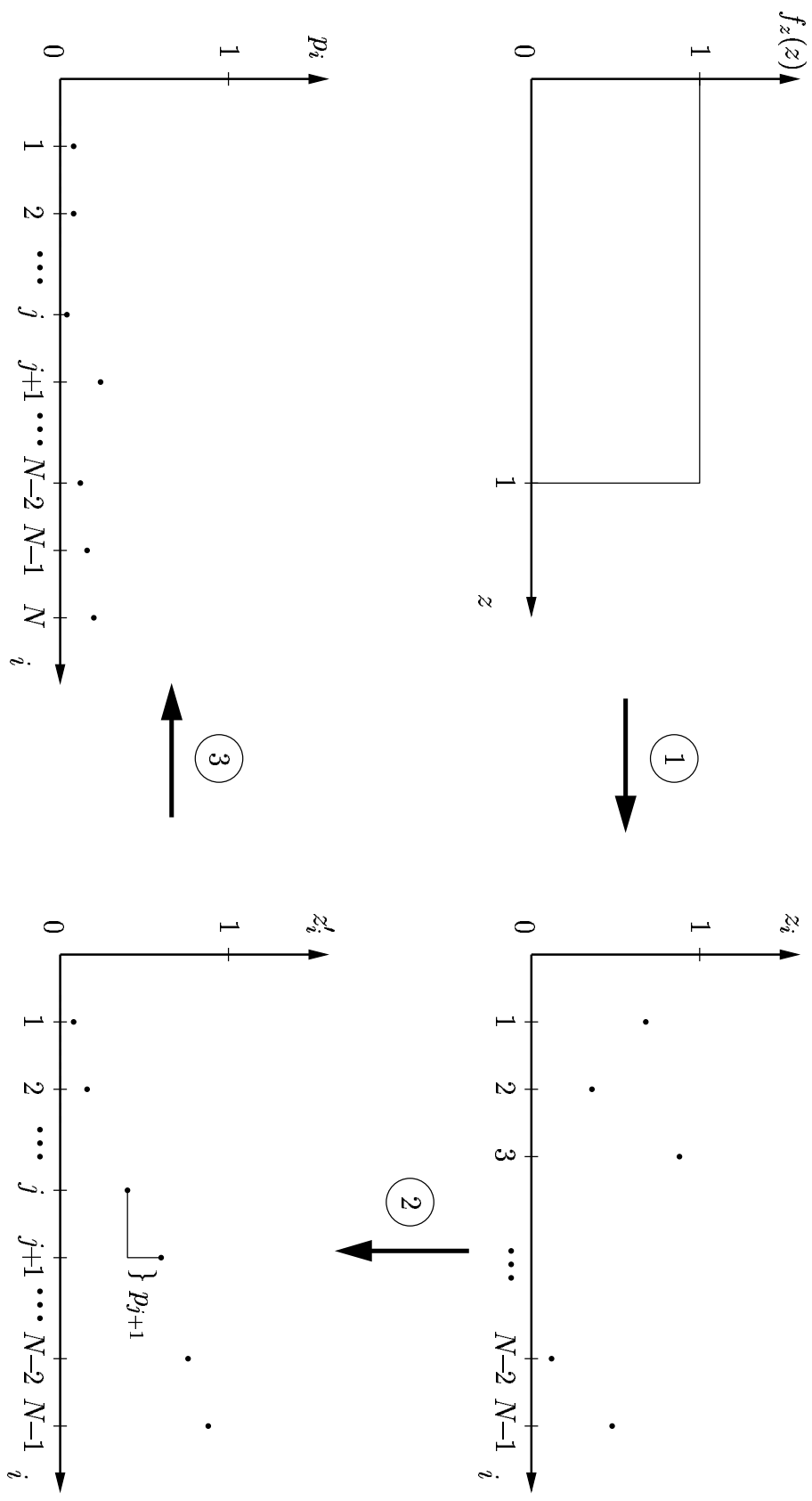


Figure B.1: *Process for generating probabilities*

Appendix C

Memory Management and Data Transfer in Computer Systems

The task of transferring data has an almost fractal character, since it occurs not only between computers but of course also in various levels inside of a computer, among its components. We will briefly outline the problems arising in this field. For this purpose we start at the fundamental von-Neumann architecture of computers and quickly proceed to today's computer systems that still follow their predecessor's basic principle.

The early computer proposed by von Neumann[1] included three components: a central processing unit (CPU), a fast random access memory (RAM) and a slow persistent storage.

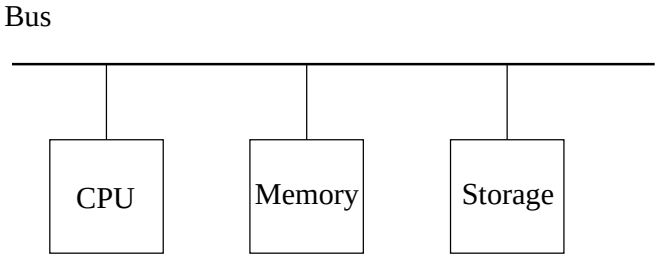


Figure C.1: *Components of basic von Neumann computer*

Both program code and data is represented in binary format. After transferring a program from the storage into the memory, the CPU's control unit executes the basic machine cycle:

- **FETCH** an instruction from memory to the CPU.

[1] John von Neumann, Hungarian Mathematician, ∗ 28 December 1903; † 8 February 1957, recognized for his ground-breaking contributions to mathematics, quantum theory and computer science as one of last century's most brilliant scientists

- **DECODE** the instruction in the CPU.
- **EXECUTE** the instruction.

Following the instructions, the CPU loads data from the memory into its registers. The CPU's arithmetic and logic unit (ALU) performs a binary operations on the data after which the data is kept in the CPU's registers or stored in the memory. Only a few registers exist in the CPU to store blocks of data bits (*words*). The size of a process, consisting of a program's data and instructions is limited by the capacity of the computer's main memory. While still working according to von-Neumann's basic principle, modern computers use techniques to allow for programs that are significantly larger than the memory of the computers they run on. These techniques are applied in the implementations of the computer's operating systems. As we will see an operating system's task of managing the memory of a computer system involves some problems similar to those occurring in effectively using communication resources for information access. A comprehensive treatment of memory management can be found in [Tan87], therefore only a very basic introduction is given here.

C.1 Paging and Swapping

The slow persistent storage mentioned above (today typically a hard-disk drive) has far higher capacity than the main memory. An obvious method to facilitate large programs is therefore to load only the share of data and instructions into the memory that are currently manipulated. If data or instructions have to be transferred to the CPU that are not in the memory, the sequence of the program is interrupted, they data currently in the memory is transferred to the disk, the needed data is transferred from the disk to the memory and the execution of the program can commence. The size of processes may grow far beyond the size of the memory with this technique. This achievement comes with a serious drawback. While the capacity of a hard-disk is usually larger than the main memory's by orders of magnitude, reading and writing from a hard-disk is slower by orders of magnitude. It is beneficial to partition the memory in small logical blocks, called *pages*. The technique to dynamically load the necessary blocks into the memory is called *Paging*. To improve performance a separate memory management unit (MMU) can pro-actively fetch and restore the pages before they are needed by the CPU, thus reducing the CPU's idle time. Except for sporadic user input the sequence of required pages is totally deterministic and could, in theory, be pre-determined, as any program follows some deterministic algorithm. Yet, the computational complexity to determine this sequence is similar to running the algorithm itself.

For most applications, with certain exceptions in real-time systems, deterministic pre-determination is not a sensible approach. Instead, the MMU continuously collects statistical data during the execution of the algorithm and tries to speculatively load the required blocks of data. Several replacement strategies have been developed.

Their performance is usually evaluated by comparing them with the optimal page replacement strategy which is determined by running sample programs in a simulator twice. On the first run the page reference information is collected and used in the second run. Most strategies (e.g. not-recently used (NRU), first-in-first out (FIFO), second chance, least recently used (LRU), working set model) take the very limited resources for storing statistics and calculating metrics into account and try to come as close as possible to the optimum replacement strategy within their constraints.

Closely related to paging are the effects that occur when the memory and CPU are shared by multiple processes resulting in the need to dynamically store and retrieve a process' data from disk to memory. This technique is called *swapping*. For further interest we again refer to [Tan87].

C.2 Caching

While paging and swapping are techniques that are typically applied to cope with the limited size of a computer's main memory, the term *caching* is used for techniques applied to improve data transfer between storage and main memory (*disk-caching*) as well as between main memory and small but fast memory units located more closely to the systems CPU.

Usually operating systems do not optimize the action of paging, swapping and disk-caching in a joint process. In the past this lead to sometimes paradox constellations where the competition among these techniques rarely resulted in an optimum performance. Today caching between storage and main memory frequently employs dedicated memory onboard of disk drives and disk controllers.

Modern CPUs make use of a cascade of cache memories to accommodate processing speed and the speed of the system's main RAM. Currently, this cascade consists of three levels (L1, L2, L3), where L1 is the fastest and usually smallest memory. Typically L1 and L2 reside on the same die as the CPU[2]. The strategies used for managing the content of these caches are similar to the ones applied for paging and swapping.

[2]To illustrate the impressing data transfer rates we give some data on a typical representative i.e. an Intel Pentium IV processor. Its L2 cache's size is 256 KB. Its connection to the CPU core is 256 bit wide, operating at the core clock speed. For 1.5 GHz core clock speed, this results in 358 Gb/s data transfer rate.

List of Figures

List of Tables

Glossary

List of Acronyms

ALU	Arithmetic and Logic Unit
AP	Access Point
API	Application Programming Interface
BT	Bluetooth
CA	Client Application
CDMA	Code Division Multiple Access
CERN	European Organization for Nuclear Research
CP	Central Proxy
CPU	Central Processing Unit
CRLB	Cramer-Rao Lower Bound
CSD	Circuit Switched Data
C-SIE	Central Situation Inference Engine
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DSL	Digital Subscriber Line
GIF	Graphics Interchange Format
GGSN	GPRS gateway support node
GPS	Global Positioning System
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications (formerly Groupe Spécial Mobile)
HSCSD	High Speed Circuit Switched Data
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol (RFC 2616)

IETF Internet Engineering Task Force
IP Internet Protocol (RFC 791)
IR Infrared
IrDA Infrared Data Association

JPEG Joint Photographic Experts Group

LSP Local Service Point

MAC Media Access Control
MAP Maximum A Posteriori
MD Message Digest
MD5 Message Digest algorithm 5 (RFC 1321)
ML Maximum Likelihood
MLE Maximum Likelihood Estimator
MMSE Minimum Mean Square Error
MMU Memory Management Unit
MP Mobile Proxy
M-SIE Mobile Situation Inference Engine

NAT Network Address Translation (RFC 3022)

OS Operating System

PAN Personal Area Network
PARC Palo Alto Research Center
PC Prefetching Controller
PDA Personal Digital Assistant
PDF Probability Density Function
PLMN Public Land Mobile Network
PMF Probability Mass Function
PNG Portable Network Graphics
PPM Prediction by Partial Matching
PPP Point-to-Point Protocol (STD 0051, RFC 1661)
PSTN Public Switched Telephony Network

RAM Random Access Memory
RFC Request For Comment (IETF standardization document)
RLP Radio Link Protocol

RP	Resident Proxy
R-SIE	Resident Situation Inference Engine
SA	Server Application
SDL	Specification and Description Language
SDL-GR	Specification and Description Language – Graphical Representation
SHA-1	Secure Hash Algorithm 1 (RFC 3174)
SIE	Situation Inference Engine
SRAL	Short Range Adaptation Layer
TCP	Transmission Control Protocol (RFC 793)
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol (RFC 768)
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier (RFC 2396)
URL	Uniform Resource Locator (RFC 1738)
VPN	Virtual Private Network
WAP	Wireless Application Protocol
WGS 84	World Geodetic System of 1984
WID	Wireless Information Device
WLAN	Wireless Local Area Network
WML	Wireless Markup Language
WPAN	Wireless Personal Area Network
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language

List of Symbols and Operators

$\mathbf{0}_{n,n}$	all-zero matrix of dimension $n \times n$
$\boldsymbol{A}$	adjacency matrix
C_i	datarate for i-th document
$\boldsymbol{C}_k$	coverage region matrix of k-th access point
Δc_i	incremental cost to prefetch i-th document
D_i	i-th document
$D(\cdot\|\cdot)$	relative entropy or Kullback-Leibler distance
$\boldsymbol{D}_i$	diffusion matrix for i-th waypoint
$E\cdot$	expected value of
$\mathcal{E}$	set of edges (of topology graph)
$f_x(x)$	probability density function of continuous random variable x
$\mathcal{G}$	topology graph
$H(\boldsymbol{x})$	entropy of a discrete random variable $\boldsymbol{x}$
$H(\boldsymbol{x}, \boldsymbol{y})$	joint entropy of two discrete random variables $\boldsymbol{x}$ and $\boldsymbol{y}$
$H(\boldsymbol{x}\|\boldsymbol{y})$	conditional entropy of discrete random variable $\boldsymbol{x}$ given $\boldsymbol{y}$
$\boldsymbol{I}_{n,n}$	identity matrix of dimension $n \times n$
$I(\boldsymbol{x}; \boldsymbol{y})$	mutual information between two discrete random variables $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{L}$	layout map matrix
N_{Γ_j}	number of components of j-th aspect
N_σ	number of possible situations
N_c	number of central nodes
N_m	number of mobile nodes
N_r	number of resident nodes
N_s	number of server nodes
$N_\mathcal{W}$	number of waypoints
$p_x(x)$	probability mass function of discrete random variable x
p_i	i-th probability
$p_{i,j}$	transition probability from state i to *state* j
p_{th}	threshold probability for prefetching
$p_{\boldsymbol{x}}(x_i)$	probability mass function (PMF) of a discrete random variable $\boldsymbol{x}$
$P(\cdot)$	power set of
$\Pr\{\cdot\}$	probability of
$\Pr\{\cdot\|\cdot\}$	conditional probability
t_R	request time
T_w	waiting time
T_{w_0}	time after which all candidate documents have been prefetched
V	volume
V_i	volume of i-th document

$\mathcal{V}$	set of vertices (of topology graph)
X_σ	fundamental set of situation space
α	parameter of Zipf distribution
$\alpha_{\sigma_i,\sigma_j}$	situation specific symptom
$\alpha_{\gamma_{h,i},\gamma_{h,j}}$	component specific symptom
$\beta_{\sigma_j,i}$	consequence
Γ	aspect
Γ_j	j-th aspect
ϵ	estimation error
Θ	parameter to be estimated
$\kappa_{U,T}$	cost factor for waiting time
$\kappa_{N,T}$	cost factor for temporal use of network
$\kappa_{N,V}$	cost factor for transport of volume over network
σ	situation
σ_i	i-th situation
σ^*	elementary situation
τ	topology
$\times$	Cartesian product
$\lvert \cdot \rvert$	cardinality of
$\cap$	logical AND
$\cup$	logical OR

Bibliography

[AK02] Michael Angermann and Jens Kammann. Cost metrics for decision problems in wireless ad hoc networking. In *Proceedings IEEE CAS 2002*, Pasadena, USA, September 2002.

[AKL03] Michael Angermann, Jens Kammann, and Bruno Lami. A new mobility model based on maps. In *Proceedings of the IEEE Semiannual Vehicular Technology Conference*, Orlando, Florida, USA, October 2003.

[AKR+01] Michael Angermann, Jens Kammann, Patrick Robertson, Alexander Steingass, and Thomas Strang. Software representation for heterogeneous location data sources within a probabilistic framework. In *Proceedings of the International Symposium on Location Based Services for Cellular Users, LOCELLUS 2001*, Munich, February 2001.

[Ang99] Michael Angermann. Navigation capabilities of future mobile communication systems – will global navigation satellite systems become obsolete? *Proceeding of GNSS '99*, 1999.

[Ang02] Michael Angermann. Analysis of speculative prefetching. *ACM Mobile Computing and Communications Review*, 6(2):13–17, April 2002.

[Ang03] Michael Angermann. Differences in cost and benefit of prefetching in circuit-switched and packet-switched networks. In *Procedings of the 10th International Conference on Telecommunications IEEE ICT'2003*, Tahiti, Papeete, French Polynesia, February 2003.

[Bar01] Paul R. Barford. *Modeling, Mesurement and Performance of World Wide Web Transactions*. PhD thesis, Boston University, Graduate School of Arts and Science, 2001.

[BCF+99] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of the IEEE Infocom 1999*, pages 126–134, 1999.

[Bet01] Christian Bettstetter. Smooth is better than sharp: A random mobility for simulation wireless networks. In *Proceedings of the 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and and Mobile Systems (MSWiM'01)*, Rome, Italy, July 2001.

[BFA01] Steven Bennett, Adam Felton, and Rahmi Akçelik. Pedestran movement characteristics at signalised intersections. In *23rd Conference of Australian Institutes of Transport Research (CAITR 2001)*, Melbourne Australia, December 2001.

[BHS91] Ferenc Belina, Dieter Hogrefe, and Amardeo Sarma. *SDL with applications from protocol specification*. Prentice Hall International, 1991.

[BJ01] Thorsten Bohnenberger and Anthony Jameson. When policies are better than plans: Decision-theoretic planning of recommendation sequences. In James Lester, editor, *IUI 2001: International Conference on Intelligent User Interfaces*, pages 21–24. ACM, New York, 2001. Available from http://dfki.de/∼jameson/abs/BohnenbergerJ01.html.

[BVE99] Christian Bettstetter, Hans-Jörg Vögel, and Jörg Eberspächer. GSM Phase 2+ General Packet Radio Service GPRS: Architecture, protocols, and air interface. *IEEE Communications Surveys*, 2(3), 1999.

[CB98] Mark Crovella and Paul Barford. The network effects of prefetching. In *IEEE Infocom, San Francisco, CA*, 1998.

[CD01] I. Cooper and J. Dilley. Known HTTP proxy/caching problems. *Internet Draft*, April 2001.

[CER] http://public.web.cern.ch/public/.

[CK00] Guanling Chen and David Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000.

[CL00] Bradley P. Carlin and Thomas A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall, 2000.

[CMT01] I. Cooper, I. Melve, and G. Tomlinson. Internet web replication and caching taxonomy. *RFC 3040*, January 2001.

[CP01] Xinjie Chang and David W. Petr. A survey of pricing for integrated services networks. *Computer Communications*, 24(18):1808–1818, December 2001.

[CSEZ93] Ron Cocchi, Scott Shenker, Deborah Estrin, and Lixia Zhang. Pricing in computer networks: motivation, formulation, and example. *IEEE/ACM Transactions on Networking*, 1(6):614–627, 1993.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[CW84] John G. Cleary and Ian H. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32(4):396–402, April 1984.

[Dav01] Brian D. Davison. Assertion: Prefetching with GET is not good. In *Proceedings of the 6th International Workshop on Web Caching and Content Distribution*, June 20-22 2001.

[Dey00] Anind K. Dey. *Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.

[DMAC02] Anind K. Dey, Jennifer Mankoff, Gregory D. Abowd, and Scott Carter. Distributed mediation of ambiguous context in aware environments. In *Proceedings of the 15th Annual Symposium on User Interface Software and Technology (UIST 2002)*, pages 121–130, Paris, France, October 2002.

[FCLJ99] Li Fan, Pei Cao, Wei Lin, and Quinn Jacobson. Web prefetching between low-bandwidth clients and proxies: Potential and performance. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '99), Atlanta, GA*, May 1999.

[Flu95] Fançois Fluckiger. *Understanding Networked Multimedia: Applications and Technology*. Pearson, 1995.

[GCSR65] Andrew B. Gelman, John S. Carlin, Hal. S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, Boca Raton, 1965.

[Gil00] Donald Gillies. *Philosophical Theories of Probability*. Routledge, London, 2000.

[GJW99] Barbara Großmann-Hutter, Anthony Jameson, and Frank Wittig. Learning Bayesian networks with hidden variables for user modeling. In *Proceedings of the IJCAI 99 Workshop "Learning About Users"*, pages 29–34, Stockholm, 1999. Available from http://dfki.de/∼jameson/abs/Grossmann-HutterJW99.html.

[Goo65] Irving John Good. *The Estimation of Probabilities – An Essay on Modern Bayesian Methods*. The M.I.T. Press, Cambridge, 1965.

[GR65] I. S. Gradsteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, London, 1965.

[GT94] Kaj Grønbæk and Randall H. Trigg. Design issues for a dexter-based hypermedia system. *Communications of the ACM*, 37(2):40–49, 1994.

[HREM99] Ernst-Georg Haffner, Uwe Roth, Thomas Engel, and Christoph Meinel. A semi-random prediction scenario for user requests. In *Proceeding of APWeb '99, Asia Pacific International Web Conference, Hong Kong, China*, 1999.

[HS94] Frank Halasz and Mayer Schwartz. The dexter hypertext reference model. *Communications of the ACM*, 37(2):30–39, 1994.

[IX00] Tamer I. Ibrahim and Cheng-Zhong Xu. Neural net based pre-fetching to tolerate WWW latency. In *Proc. of the 20th IEEE Int'l Conf. on Distributed Computing Systems (ICDCS2000)*, April 2000.

[Jam95] Anthony Jameson. Logic is not enough: Why reasoning about another person's beliefs is reasoning under uncertainty. In Armin Laux and Heinrich Wansing, editors, *Knowledge and Belief in Philosophy and Artificial Intelligence*, pages 199–229. Akademie Verlag, Berlin, 1995. Available from http://dfki.de/∼jameson/abs/Jameson95.html.

[Jam01] Anthony Jameson. Modeling both the context and the user. *Personal and Ubiquitous Computing*, 5(1):29–33, 2001. Available from http://dfki.de/∼jameson/abs/Jameson01PT.html.

[Jay89] E. T. Jaynes. Probability theory as logic. In *Ninth Annual Workshop on Maximum Entropy and Bayesian Methods, August 14, Dartmouth College, New Hampshire*, August 1989. revised, corrected and extended 1994.

[JK97] Zhimei Jiang and Leonard Kleinrock. Prefetching links on the WWW. In *Proceedings of the ICC '97 Montreal, Canada*, pages 483–489, June 1997.

[JK98] Zhimei Jiang and Leonard Kleinrock. An adaptive network prefetch scheme. *IEEE Journal on Selected Areas in Communications*, April 1998.

[JSR] Expert Group JSR-82. Java APIs for Bluetooth. http://jcp.org/jsr/detail/082.jsp.

[Kay93] Steven M. Kay. *Fundamentals of Statistical Signal Processing – Estimation Theory.* Prentice Hall, New Jersey, 1993.

[KB02] Jens Kammann and Tim Blachnitzky. Split-proxy concept for application layer handover in mobile communication systems. In *Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002), Stockholm*, September 2002.

[Kel97] Frank Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997. (corrected version).

[Kle75] Leonard Kleinrock. *Queueing Systems, Volume I: Theory.* John Wiley & Sons, 1975.

[Kle76] Leonard Kleinrock. *Queueing Systems, Volume II: Computer Applications.* John Wiley & Sons, 1976.

[Kle02] Leonard Kleinrock. Creating a mathematical theory of computer networks. *Operations Research*, 50(1):125–131, January-February 2002.

[KPN96] Richard. L. Knoblauch, Martin T. Pietrucha, and Marsha Nitzburg. Field studies of pedestrian walking speed and start-up time. In *Transportation Research Record 1538*, pages 27–38, Washington, D.C., December 1996. National Research Council, Transportation Research Board.

[KYVD03] R. Kokku, P. Yalagandula, A. Venkateramani, and M. Dahlin. A non-interfering deployable web prefetching system. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, March 2003.

[Lee61] C. Y. Lee. An algorithm for path connections and its applications. *IRE Transactions on Electronic Computing, EC-10*, pages 346–365, 1961.

[Mah97] Bruce A. Mah. An empirical model of HTTP network traffic. In *INFOCOM (2)*, pages 592–600, 1997.

[Oes99] Bernd Oestereich. *Objectorientierte Softwareentwicklung.* Oldenbourg, München, 1999.

[OFMP+94] Anders Olsen, Ove Færgemand, Birger Møller-Pedersen, Rick Reed, and J. R. W. Smith. *Systems engineering using SDL-92.* Elsevier Science B.V., Amsterdam, 1994.

[RJB99] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The unified modeling language reference manual.* Addison Wesley Longman, Inc., Massachusetts, 1999.

[RRH01] Odd-Wiking Rahlff, Rolf Kenneth Rolfsen, and Jo Herstad. Using personal traces in context space: Towards context trace technology. *Personal and Ubiquitous Computing*, (5):50–53, 2001.

[SA93] G.K. Schmidt and K. Azarm. Mobile robot path planning and execution based on a diffusion equation strategy. *Advanced Robotics*, 7(5):479–490, 1993.

[Sat01] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, pages 10–17, August 2001.

[SAW] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications.

[Sch02] Albrecht Schmidt. *Ubiquitous Computing – Computing in Context*. PhD thesis, Computing Department, Lancaster University, U.K., November 2002.

[SE01] P. Srisuresh and K. Egevang. Traditional IP network address translator. *RFC 3022*, January 2001.

[Siv96] D.S. Sivia. *Data Analysis – A Bayesian Tutorial*. Oxford University Press, Oxford, New York, 1996.

[SSR89] Roberto Saracco, J. R. W. Smith, and Rick Reed. *Telecommunications Systems Engineering using SDL*. Elsevier Science B.V., Amsterdam, 1989.

[Tan87] Andrew S. Tanenbaum. *Operating Systems, Design and Implementation*. Prentice-Hall, 1987.

[Tan00] Andrew S. Tanenbaum. *Computernetzwerke*. Prentice-Hall, 2000.

[TKV97] Nor Jaidi Tuah, Mohan Kumar, and Svetha Venkatesh. A performance model of speculative prefetching in distributed information systems. In *Intl. Parallel Processing Symposium, San Juan, Puerto Rico*, 1997.

[TKV99] N. J. Tuah, M. Kumar, and S. Venkatesh. Performance model of speculative prefetching in distributed information systems. In *Proceedings IPPS/SPDP 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing*, April 1999.

[TLAC95] Carl Tait, Hui Lei, Swarup Acharya, and Henry Chang. Intelligent file hoarding for mobile computers. In *Proceeding of the First ACM Conference on Mobile Computing and Networking (Mobicom) '95, Berkeley*, 1995.

[Tua00] N. J. Tuah. *Performance Modelling of Caching and Prefetching for Information Access in Distributed Systems*. PhD thesis, Curtin University of Technology, 2000.

[TvS02] Andrew S. Tanenbaum and Maarten van Steen. *Distributed Systems Priciples and Paradigms*. Prentice-Hall, 2002.

[VK96] Jeffrey Scott Vitter and P. Krishnan. Optimal prefetching via data compression. *Journal of the ACM*, 43(5):771–793, 1996.

[vT68] Harry L. van Trees. *Detection, Estimation, and Modulation Theory*. John Wiley and Sons, New York, 1968.

[WC97] D. Wessels and K. Claffy. Internet cache protocol (ICP), version 2. *RFC 2186*, September 1997.

[Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, pages 66–75, September 1991.

[Wei93] Mark Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, 1993.

[WHFG92] Roy Want, Andy Hopper, Veronica Falco, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS) archive*, 10(1):91–102, January 1992.

[WSA+95] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis, and Mark Weiser. The parctab ubiquitous computing experiment. Technical report, 1995.

[XPMS01] George Xylomenes, George Polyzos, Petri Mähönen, and Mika Saaranen. TCP performance issues over wireless links. *IEEE Communications Magazine*, 39(4):52–58, 2001.

[YZL01] Quinag Yang, Haining Zhang, and Tianyi Li. Mining web logs for prediction models in WWW caching and prefetching. In *Proceeding of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '01, August 26-29, San Francisco*, 2001.

[Zha01] Haining Zhang. Improving performance on WWW using path-based predictive caching and prefetching. Master's thesis, Simon Fraser University, February 2001.

[ZL01] Baihua Zheng and Dik Lun Lee. Processing location-dependent queries in a multi-cell wireless enviroment. In *Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access*, pages 54–65. ACM Press, 2001.