Resource Sharing in Mobile Wireless Networks

Maria Papadopouli

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2002

Resource Sharing in Mobile Wireless Networks

Maria Papadopouli

Advisor: Henning Schulzrinne

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2002

ABSTRACT

Resource Sharing in Mobile Wireless Networks

Maria Papadopouli

Motivated by the intermittent connectivity that many mobile users experience, we have been investigating mechanisms to improve their access to data. We propose 7DS, a system that addresses the challenge of increasing data availability by providing a novel mechanism that enables wireless devices to share resources in a self-organizing manner, without the need for an infrastructure.

7DS is an architecture, a set of protocols, and an implementation enabling resource sharing among peers that are not necessarily connected to the Internet. Peers can be either mobile or stationary. The focus is on three facets of cooperation, namely information sharing, bandwidth sharing, and message relaying. In the information sharing facet, peers query, discover, and disseminate information. Hosts acquire the information from the cache of other peers. For message relaying, hosts forward messages to the Internet (when they gain Internet access) on behalf of other hosts. We investigate the bandwidth sharing in wireless LANs and in video-on-demand servers. When bandwidth sharing is enabled in a wireless LAN, the system allows a host to temporarily act as an application-based gateway and share its connection to the Internet. In the video-on-demand case, the server shares dynamically its disk bandwidth among the clients.

The system adapts its communication behavior based on the availability of energy and bandwidth. For the information sharing and message relaying, we model several schemes depending on their type of cooperation among hosts, querying mechanism, energy conservation, host density, and transmission power. We evaluate these schemes and their impact on information discovery and data availability via simulations. We also provide an analytical model for a baseline scheme and show that the analytical results on data dissemination are consistent with the simulation results. For the case of bandwidth sharing in wireless LANs, we design a lightweight protocol and present its benefits via simulations. For the bandwidth sharing in a video-on-demand multi-disk server, we present novel retrieval techniques that take advantage of layered multimedia information and replication to dynamically reallocate the disk bandwidth. We model a multi-disk environment and show its performance in the case of no replication, partial replication and full replication as a function of user access skew. Our scheduling algorithm for the retrieval of streams can double the disk bandwidth utilization of the server.

Contents

List of Figures

List of Tables

Acknowledgments

My experience as a graduate student at Columbia University in New York has been amazing. Receiving a doctorate is a wonderful occasion to thank the many people who have taught, supported, inspired, and amused me.

First and foremost, I would like to thank my advisor Henning Schulzrinne. Henning has been a wonderful advisor for me and feel privileged to work under his supervision. I am deeply indebted to him for giving me direction and freedom, for being so generous, kind and supportive, and for everything that I learned from him. I am amazed by his dedication to research, and his contributions as a researcher and an advisor.

I am also grateful to my former advisor Leana Golubchik, who introduced me to research, supported me, and has given me her friendship and encouragement. Chapter 6 is work that I did with her while she was at Columbia University.

I would like to thank Chatschik Bisdikian, Sal Stolfo, Vishal Misra, and Erich Nahum for serving my dissertation committee and for their comments in a preliminary version of this thesis.

I wish to thank all colleagues in the Computer Science Department at Columbia University who have generously invested their time to offer comments and sugges-

tions on various papers and technical presentations: Knarig Arabshian, Kevin Butler, Tiberiu Chelsea, Jonathan Lennox, Anargyros Papageorgiou, Kundan Singh, Krish Sridhar, Gong Su, Erez Zadok, Kazi Zaman, and Xiaotao Wu. I am grateful to Wenyu Jiang who was always available for helpful and clarifying discussions. Thanks algo go to the two M.S. students, Denis Abramov and Stelios Sidiroglou-Douskos, who collaborate with us and implemented a major part of the 7DS prototype.

The IRT group has been a nurturing and supportive environment. The students and visitors to the IRT lab have contributed to my experience by providing friendship as well as technical expertise throughout the years.

I am also thankful to Chatschik Bisdikian, Andrew Campbell, Angelos Keromytis, Mahmoud Naghshineh, Vishal Misra, and Ken Ross for their advices and valuable comments during my job search and interview process.

I am grateful to the research groups in IBM T.J. Watson Research Center that hosted me during the summer of 1995, 1999, and 2000. In particular, many thanks go to Mahmoud Naghishineh, Chatschik Bisdikian, Dan Dias, and Martin Kienzle for inviting me to work in their group and for creating a stimulating, research environment and a very friendly and supportive atmosphere. I really enjoyed being a student summer intern there. I am also thankful to Paul Castro for his collaboration during the summer 2000.

I would like to thank also Zvi Kedem from the Computer Science Department at New York University for his encouragement. The discussion we had in September of 1995 had an impact on my decision to continue my graduate studies.

Thanks go to the administration of the department Rosemary Addarich, Alice Cueba, Genevive Goubourn, Patricia Hervey, Mary Van Starrex, and Susan Tritto

who always assisted with a smile.

I am also indebted to my friends Alexandros Elefteriadis, Anargyros Papageorgiou, Suren Talla, and Kazi Zaman. They have influenced, adviced, and inspired me in many ways. I wish to thank all other friends who shared with me the joys and challenges of my life as a graduate student: Tiberiu Chelsea, Apostolos Dailianas, Taria Glinou, Tobias Hollerer, Emily Kotzamani, Marina Kotzamani, Ramesh Jakka, Andreas Prodromidis, Yiannis Stamos, Ed Watson, and Ela. They were source for vibrant conversations and amusement.

My love and gratitude go to my parents, Xakousti Plevraki-Papadopouli and Giorgo Papadopouli, and my sister Eva Papadopouli for their life-long love and support. My love also to my grandfather Emmanuel Plevraki who born in 1899 has touched three centuries with such joie de vivre. He actively participated in ethnic and political events in Greece and witnessed so vast political, social, economic, and technological changes. By now, he has his own mythology and almost a homeric presence in my life. This thesis is dedicated to them.

Finally, I would like to mention that funding for my graduate studies has been provided by an NSF CAREER award ANI-99-85325 given to Henning Schulzrinne.

Στους γονείς μου Ξακουστή και Γιώργο,

την αδελφή μου Εύα, και

τον παππού μου Εμμανουήλ Πλευράκη

To my parents Xakousti and Giorgo,

my sister Eva, and

my grandfather Emmanuel Plevraki

Chapter 1

Introduction

1.1 Motivation

Wireless devices are becoming smaller, more user friendly and more pervasive. They are not only carried by people, but are integrated into physical objects. These devices can be part of data-centric, mobile, ad hoc (without infrastructure) and sensor networks; they collect, measure, process, query, and relay information. The expansion of the Internet and wireless data communications have amplified this trend by making information easier to share and by increasing the amount of information that is shared. Wireless information access will become as important as voice communications, since people are beginning to heavily depend on on-line information. People access local and general news, traffic or weather reports, sports, maps, guide books, music, video files, and games [3, 15, 65]. For example, Table 1.1 shows the number of wireless users, Table 1.2 shows the popular PDAs in use today and Figures 1.3 and 1.2 show users of two wireless Internet service providers, Avantgo and Vindigo [4, 97], respectively. Similarly, there is growing interest in the transportation industry to support vehicles

with navigation tools and location-based services. Examples of such services are location tracking, maps, driver/trip task lists, traffic reports, address lookup, routing information, fleet tracking, inter-vehicle entertainment, streaming and collaborative applications.

	2000	**2002**	**2005**
United States			
Internet Users	135	169	214
Wireless Internet Users	2	18	83
Worldwide			
Internet Users	414	673	1,174
Wireless Internet Users	40	225	730
Western Europe			
Internet Users	95	148	246
Wireless Internet Users	7	59	168

Table 1.1: Internet Users vs. Wireless Users (millions). Source: eTForecasts (February 13, 2001) [31].

Wireless data communications have four fundamental constraints: power, bandwidth, information accuracy and personal privacy. The first two are fundamental constraints of the wireless arena. The third is intrinsic to the highly dynamic environment in which these networks & devices operate with errors, packet losses, redundancy, imprecision and limited capabilities of the mobile devices. The redundancy results from the large amount of information available in the web or in information servers. Information can be inherently imprecise and change dynamically. Personal privacy becomes essential as the communications environment becomes more complex and global. Due to these constraints, devices need to be self-configuring and adaptive to better utilize their resources and provide robustness without compromising user privacy. At the same time, the participants may have different requirements in terms

of information accuracy, capabilities (power or bandwidth constraints), and degree of trust or cooperation among each other.

	2000	**2001**	**2003**	**2005**	**2007**
Worldwide PDA	24,920	40,435	85,620	149,030	227,400
Worldwide Phone-PDA	230	900	6,350	21,400	48,800
U.S. PDA	12,345	18,510	35,165	56,550	80,645
U.S. Phone-PDA	negl.	46	1,350	5,950	14,250

Table 1.2: PDAs in use (thousands). Source: eTForecasts report on "Worldwide PDA Markets" [32].

Current work in distributed systems, traditional networking and mobile systems, and Internet protocols has not solved the problem of designing efficient and scalable mechanisms for information discovery that operate under these constraints. We specifically address how wireless devices can share their resources to increase the data availability in a mobile network.

In the next sections, we discuss the mobile information access and the environment these wireless devices operate in. We propose a system that aims to increase the data availability of mobile devices and state the main contributions of this thesis.

1.1.1 Definitions

Let us introduce some terminology that we will be using throughout the thesis.

A *sensor network* is typically composed of a mass of sensors and servers that control these sensors. An *ad hoc network* is a network of wireless devices without infrastructure.

Mobile information access is the underlying querying mechanism via which wireless device receive information from a source while mobile. The mechanism de-

scribes the architecture of the system and interactivity model. In respect to the architecture, it specifies the need of an infrastructure, if any, and the main components of the system that query for information or provide the information to mobile devices. The interactivity model indicates how synchronous and direct is the communication between the user that queries for some information and the component of the system that provides the information.

A *base station* or access point is a gateway with a radio transmitter/receiver that provides Internet access to hosts in its wireless range.

An *infostation* is a fixed (stationary) information server attached to a data repository and a wireless LAN. When a wireless device is in close proximity to an infostation, it can query the server and access the information.

A *peer-to-peer* system is a distributed system without any centralized control or infrastructure. The software running at each peer host is equivalent in functionality. The peer hosts share their resources and can dynamically decide to collaborate.

An environment is characterized by *spatial locality of queries and information*, when users request for location-dependent data and it is likely users in close geographic proximity to query for similar data.

1.1.2 Mobile information access

We classify mobile information access according to their dependency on an infrastructure and the interactivity model. Depending on the need of an infrastructure, there are three categories: wireless Internet via base stations, infostations, and peer-to-peer. The first two approaches need an infrastructure. Depending on the user interactivity, the information access is either synchronous or asynchronous. We describe the mobile

information access types in the next paragraphs.

1. Wireless Internet via base stations

The first approach provides "continuous" wireless Internet access; examples include CDPD, 3G wireless, IEEE 802.11, and two-way pagers [27, 77]. The wireless Internet is broadly defined by two types of wireless networks, namely wireless wide area network (WAN) and wireless local area network (LAN). The wireless WAN is a licensed, heavily regulated wireless network used by cell phones, wireless modems; examples include CDPD, 3G wireless and two-way pagers. Wireless WAN access is typically characterized by low bit rates and high delays.

The wireless LANs (e.g., IEEE 802.11, HiPerLAN, DECT) operate in unlicensed spectrum. Currently, this access mode has either sparse coverage, low cost and high speed (IEEE 802.11) or major-cities-only coverage and high cost (Metricom) or wider coverage, but extremely low rates and high costs (CDPD, RIM).

In several cities worldwide, nonprofit groups have installed IEEE 802.11b base stations to provide free wireless access to the Internet. Figure 1.1 illustrates these zones of base stations in New York City [101].

2. Infostations

The second approach provides information access via infostations. When a wireless device is in close proximity to an infostation, it can query the server and access the information. The infostations can be located at traffic lights, building entrances, cafes and airport lounges. An infostation can be connected to a network of infostations or to the Internet. It can act as a proxy, caching data and forwarding requests to other infostations or to the Internet.

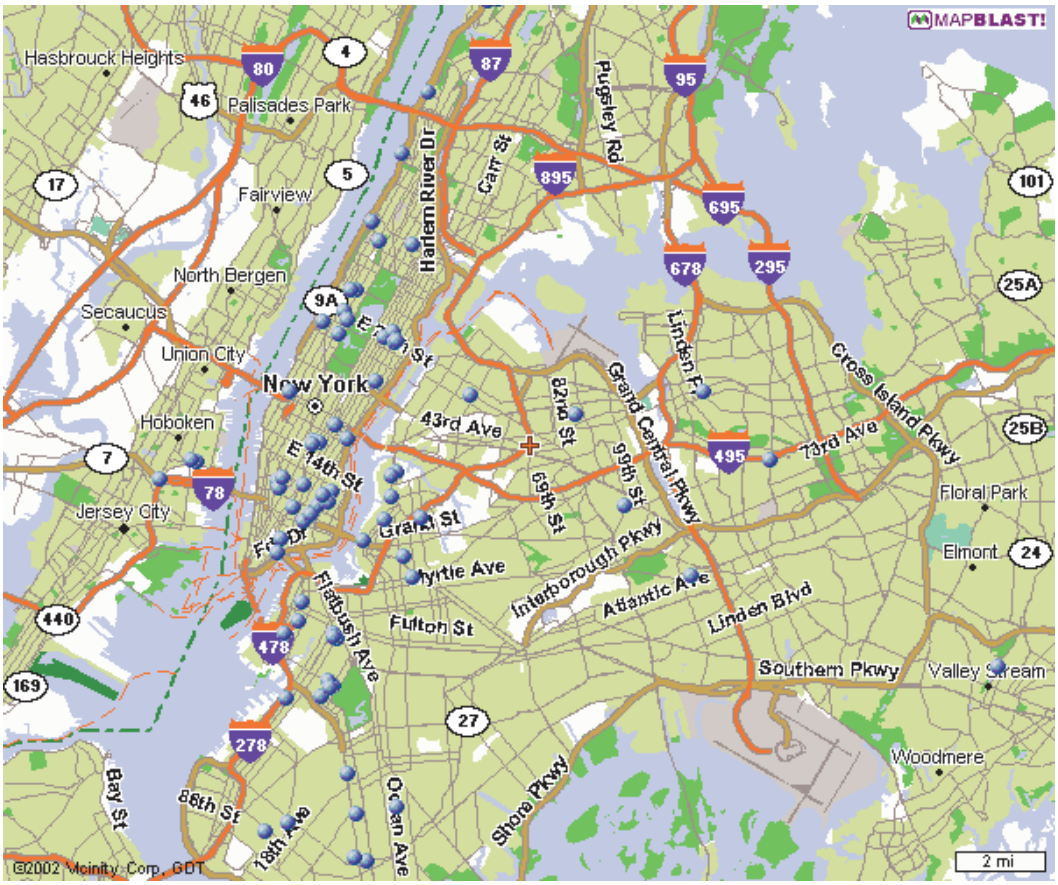

Figure 1.1: The New York City wireless public access points as of May 2002 [101]. The wireless access zones are depicted as "bullets".

3. Peer-to-Peer

We propose a third approach that does not need the support of any infrastructure (i.e., ad hoc), based on peer-to-peer resource sharing among wireless devices. In peer-to-peer mode, the participants share their resources dynamically based on a user-defined policy. The policy specifies the degree of cooperation, resource sharing, and functionality of peers. The peer-to-peer concept was originally introduced in the context of distributed systems and "reappeared" in 1999 with the widespread pop-

ularity of Napster[1]. As we discuss in Chapter 2.4, there is substantial peer-to-peer work in the file system and OS literature that is relevant. It includes the Ficus [68], JetFile [44] and Bayou [91] projects. All of them are replicated storage systems based on the peer-to-peer architecture. They are meant for a wide-scale, Internet-based use and they focus on issues related to update policies, data consistency, and reconciliation algorithms. Here, we target a different environment (of mobile wireless data access) and address different research issues.

Ad hoc mobile networks are based on a peer-to-peer mechanism for routing packets among the hosts. This multi-hop routing assumes a relatively high density of devices that are willing to cooperate with each other by routing packets. However, it is not always realistic to assume a connected network of cooperative devices. In this work, we introduce a peer-to-peer system in a new setting. Depending on the density of the mobile peer hosts, their network can be disconnected. We also vary the degree of cooperation of the peer hosts.

Depending on user interactivity, the information access is either synchronous (direct) or asynchronous (indirect using prefetching). For synchronous access, users directly (in real-time), specify their request for data and access the information from the web server or the infostation. In the asynchronous case, the mobile device acquires the data on behalf of the user without direct interaction with the user upon the receipt of an event (e.g., prior to the disconnection of the device or in the presence of an infostation).

The mobile device can access the information synchronously from the source or from the cache of the local device. Alternatively, the mobile device can access the

[1]Shawn Fanning started working in the Napster implementation between September 1998 and early 1999.

information from an infostation or another peer in an asynchronous or synchronous manner. In the asynchronous mode, the infostation may broadcast the data [50, 78]. The host can subscribe to a multicast channel of the infostation and receive the information. Another type of asynchronous access is prefetching or hoarding. The system prior to the disconnection of the device can prefetch the data from the file system or, in the general case, from an infostation. This form of prefetching, hoarding [56, 60], allows a mobile device, before disconnecting from the wired network, to prefetch data to increase the user's data availability while she/he remains disconnected and reintegrate upon reconnection. It aims to alleviate user perceived latencies. There are several hoarding strategies based on the detection of "file working sets" [94] or on capturing the semantic relationships among files in "semantic" distance measure. They addressed issues related to data consistency and targeted in a traditional file system setting. The system can locate the files prior to the disconnection. This mechanism is not adequate when the system cannot predict the information to be prefetched, when the mobile user searches for some new information while mobile or with very dynamic information.

Next, we describe the characteristics and issues of this environment.

1.1.3 Characteristics of the environment

There are four main characteristics of this dynamic, pervasive computing environment, namely heterogeneity of wireless devices and access methods, frequent disconnections and low bit rates, high spatial locality of information and queries, and heterogeneous application requirements on delay and accuracy.

Wireless transmission technology	Carrier
TDMA	AT&T, Digital PCS, Cingular networks CellularOne
GSM/GPRS	Omnipoint, Cingular, Voicestream Unicel, PinPoint Wireless
CDMA	AirTouch, Qwest, Bell Atlantic Mobile Sprint PCS, MCI WorldCom Wireless General Wireless, Verizon
CDPD	Digital PCS, GoAmerica, BellAtlantic/Nynex AT&T Verizon wireless, Omnisky
Pseudo-random FC-CDMA	Metricom

Table 1.3: U.S. wireless networks.

1. Heterogeneity of devices and access methods

We consider a setting of wireless devices with different capabilities, wireless access methods, degrees of trust and cooperation with each other. It includes handheld devices (e.g., iPAQs, palm pilots and mobile phones) with constrained memory and power, laptops or vehicular wireless systems with higher storage and power resources, and infostations with sufficient storage and no power constraints. The devices may be stationary or mobile. They are autonomous and not necessarily connected to the Internet. As we described earlier, there is a wide range of wireless networks encompassing infrared, wireless LAN (e.g., IEEE 802.11), 3G, Bluetooth, GSM, cable and satellite networks (e.g., Tables 1.4 and 1.3).

There is a variety of interaction types between them based on their capabilities and trust. Specifically, we distinguish two principal interaction types server-to-client (S-C) and peer-to-peer (P-P). We describe them in Section 1.2.

2. Changes in the bandwidth availability and loss of connectivity to the

Wireless technology	**Max bit rate**	**Frequency**	**Effective range**
Bluetooth	724 Kb/s	2.4 GHz	10 meters
Infrared	<4 Mb/s	$> 10^{14}$Hz	20 meters 100 meters 10 cm-2 meters
IEEE 802.11b	1 Mb/s 11 Mb/s	2.4 GHz	outdoors 550 meters indoors 50 meters outdoors 160 meters indoors 50 meters
3G (WCDMA)	144 K/s vehicle 384 kb/s pedestrian 1-2 Mb/s stationary	1.885 GHz- 2.2 GHz	
CDPD	19.2 Kb/s	1.8- 2.5 GHz	

Table 1.4: Bandwidth requirements, frequency, transmit power, and effective range for different wireless technologies.

Internet due to host mobility

Currently, mobile users can access information using the infrastructure of base stations (wireless LAN or WAN). Most wireless data WAN access are only available in major metro areas (such as, Vindigo [98] or RIM [83]). There are situations where a communication infrastructure is not available (such as in emergency situations, disaster relief, rescue teams, inside tunnel or subway). In other situations, there is an infrastructure, but it is overloaded or expensive to access. For instance, on September 11, 2001, after the terrorist attack in New York, it was difficult to access the communication infrastructure and the news web sites.

Given the exceedingly expensive license fees attained in recent government auctions of spectrum, the bandwidth expansion route is bound to be expensive. For example, European telecommunications giants spent $100 billion in 2000 for 3G li-

cense fees [43]. Similarly, the cost of tessellating a coverage area with a sufficient number of base stations or infostations coupled to the associated high speed wired infrastructure cost is prohibitive. For the next few years, continuous connectivity to the Internet will not be available at low cost for mobile users roaming a metropolitan area. The devices will continue to experience changes in the availability of bandwidth and frequent interruptions of connectivity due to host mobility.

3. High spatial locality of information and queries

The high spatial locality of information results from the type of services we expect a mobile user will run, namely location-dependent services, service discovery, news services and collaborative applications. For example, in an urban environment, such as a part of Manhattan during rush hours, the platform of a train, an airport, a commercial center, a corporation, or a campus we anticipate that the access patterns of the wireless devices will feature high spatial locality of information (such as local and general news, sports, train schedules, weather reports, maps, routes), service discovery queries and also popular information (such as music files or video games).

There are several wireless Internet service and information providers for handheld devices (Avantgo, Vindigo, Omnisky Corp.). For example, Avantgo regularly lists *The Wall Street Journal*, *The New York Times*, and *USA Today* as top ten user sites at www.avantgo.com/channels. Similarly, Vindigo licenses its technology to newspapers and hosts the service on behalf of its partners. Newspapers simply supply the listings in a structured format, periodically updating them. In a highway setting, vehicles with wireless capabilities will query for weather and traffic reports, maps, and routes.

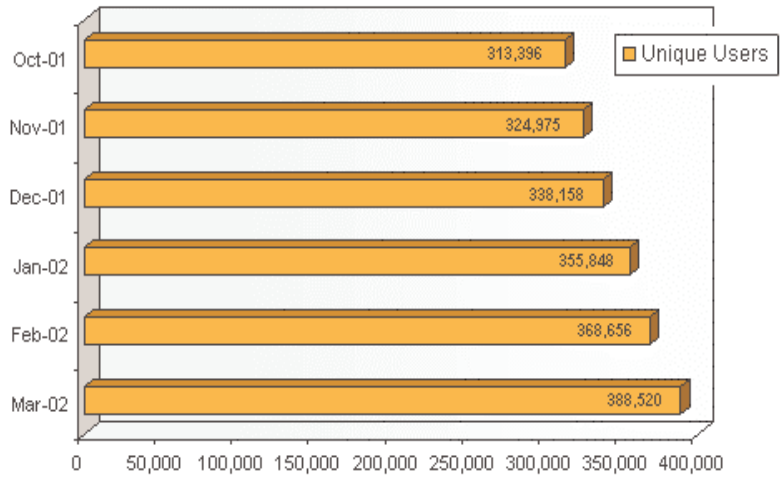

Figure 1.2: The unique number of Vindigo users that subscribe to the NYTimes news on-line information, respectively. Source: *The New York Times on the Web* [66].

4. Heterogeneous application requirements on delay and accuracy

Unlike voice communications, many wireless applications possess loose delay constraints (of the order of minutes). For example, tourists with a PDA camera that want to send pictures home can tolerate up to a few hours of delay, as long as the pictures finally reach their destination.

In many applications, users have flexible requirements on the information accuracy, freshness, precision, and media quality. For example, queries about the number of nearby taxis, closest Barnes & Noble stores or Internet cafes are inherently imprecise and may change dynamically. In other cases, users are flexible to get the information (e.g., images) in lower resolution as long as they get it fast.

1.1.4 Challenge of increasing information access

In the previous paragraphs, we described the different mobile access methods and their limitations in the environment we consider. Mobile users can access access information

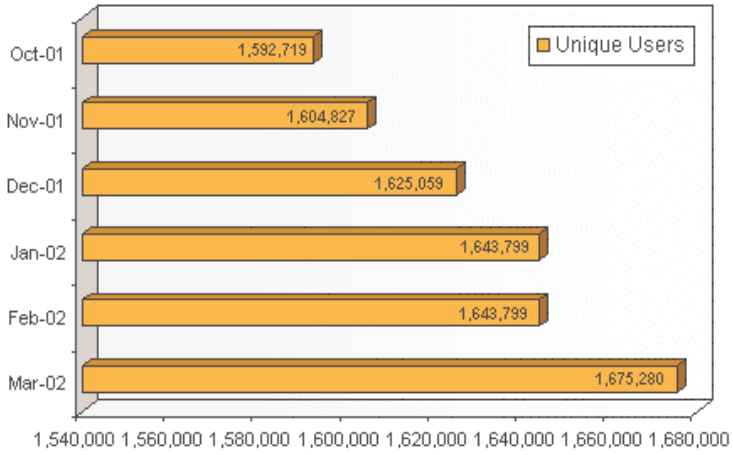

Figure 1.3: The unique number of Avantgo users that subscribe to the NYTimes news on-line information, respectively. Source: *The New York Times on the Web* [66].

using the infrastructure of base stations or the infostations, but experience frequent disconnection and low bit rates. Our main challenges are to accelerate the data availability and enhance the dissemination and discovery of information when hosts face changes in the availability of bandwidth and face the loss of connectivity to the Internet due to host mobility. We aim to investigate ways to enable these devices to share resources to increase their data access considering their power, bandwidth, and memory constraints.

1.2 Resource sharing using 7DS

We propose $7DS$[2] as a system that complements the three mobile information access approaches we described in the previous paragraph. 7DS is an architecture and set of protocols enabling resource sharing among peers that are not necessarily connected to

[2] "7DS" stands for "Seven Degrees of Separation", a variation on the "Six Degrees of Separation" hypothesis, which states that any human knows any other by six acquaintances or relatives. There is an analogy with our system, particularly, with respect to data recipients and the device with the "original" copy. We have not explored if a similar hypothesis is true here.

the Internet. We focus on three different facets of cooperation, namely, *data sharing, message relaying*, and *bandwidth sharing*.

7DS relays, searches and disseminates information, and shares bandwidth. It operates in a self-organizing manner, without the need for an infrastructure and exploits host mobility. It runs as an application on heterogeneous devices (with different capabilities) that are mobile or stationary. A 7DS-enabled device communicates with peers via a wireless LAN. We classify the 7DS-enabled devices in four categories: servers connected to the Internet, servers connected to other servers, autonomous caches (without connection to other servers or the Internet), handheld devices (power constrained, mobile). For example, a 7DS-enabled server can either be dual-homed device connected to the Internet or to a wired infrastructure of other servers or an autonomous server attached to a cache with access to a wireless LAN. Furthermore, a 7DS-enabled server can be mobile or stationary. An example of mobile server is a robot that roams a campus or a museum and disseminates information to users with handheld devices. When 7DS runs on handheld devices (e.g., PDAs), it will use energy conservation and collaboration methods different from the 7DS-enabled server. The 7DS-enabled handheld devices are sporadically connected to the Internet and 7DS can coexist with other data access methods (e.g., via wireless modem).

7DS hosts can interact either in peer-to-peer (P-P) or server-to-client (S-C) manner. In P-P mode, 7DS hosts cooperate with each other. S-C schemes operate in a more asymmetric fashion: there are some cooperative hosts (e.g., 7DS servers) that respond to queries and non-cooperative, resource constrained clients (e.g., 7DS-enabled PDAs). 7DS nodes can collaborate by data sharing, forwarding messages (such as, rebroadcasting queries and data or relaying messages to an Internet gate-

way) or caching popular data objects. For example, an autonomous 7DS server may monitor for frequently requested data, request them from other peers, and cache the data locally to serve future queries. The fixed information server (FIS) is the S-C scheme with fixed (stationary) server. 7DS is a generalization of the infostation concept. The infostation model is equivalent with the FIS mode.

In the information sharing facet, peers query, discover, and disseminate information. 7DS acquires data from other peers (in P-P) or from the infostation (S-C) within its wireless coverage using single-hop broadcast. The system takes advantage of the host mobility and periodically queries for data. A host, instead of operating with high transmission power to reach a base station or an infostation that is far away, forwards its messages or requests for data to its peers in close proximity. In that way, the hosts can conserve more power and better utilize the wireless bandwidth.

The system uses a simple energy conservation mechanism that periodically enables the network interface. During the *on* interval, 7DS hosts communicate with their peers. In its asynchronous mode, the on and *off* intervals are equal but not synchronized. In synchronous mode, the on and off intervals are synchronized among hosts, although not necessarily equal.

For bandwidth sharing, we assume that the system monitors its bandwidth capabilities and is able to compute its sustained bandwidth. We define the *sustained bandwidth* of a user, in a certain time period, as the rate at which he/she is "expected" to effectively receive data in that time period and which corresponds to a certain quality of service profile. The sustained bandwidth of a client may change due to host mobility or network congestion.

We distinguish two forms of bandwidth sharing based on the interaction among

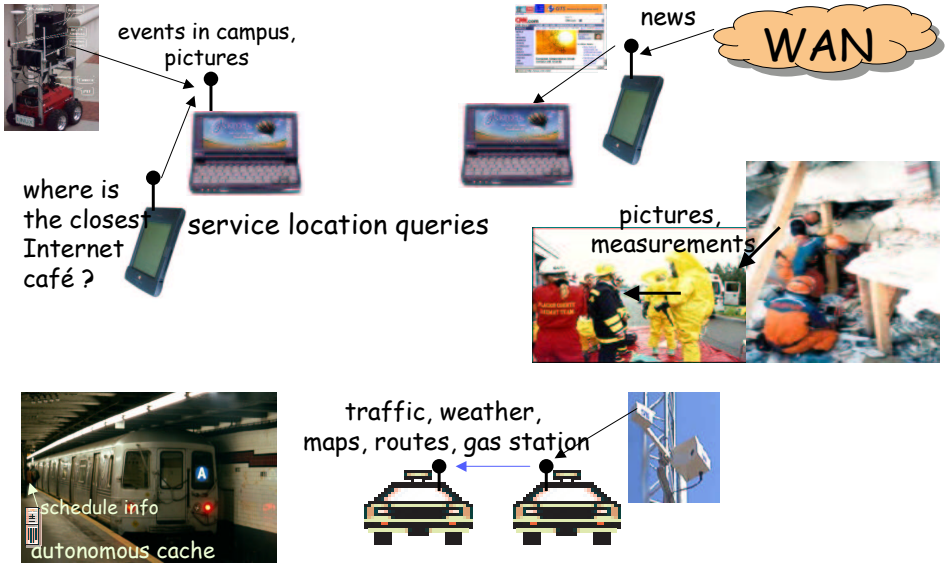

Figure 1.4: Examples of settings in which 7DS can be used: in a campus or a conference (top left), in an emergency site (right), at the platform of a subway station with an autonomous cache that runs 7DS and disseminates schedule information or news (bottom left), or in a rural area in which sensors measure traffic or weather reports and this information is disseminated among vehicles with wireless capabilities that run 7DS (bottom right).

the participants. In S-C mode, a multimedia server provides streaming data to clients with different service profiles and capabilities. The server operates in a multi-disk environment and dynamically reallocates its disk bandwidth to its clients by taking advantage of the *multiresolution property* and replication across disks. We assume that the multimedia objects are compressed at different layers of resolution. Using a subset of these layers, the client application can view the object in a lower resolution. Various video compression schemes, such as subband coding, MPEG-2, and MPEG-4, provide such a multiresolution property. In the general case, we have multimedia objects which are composed of different media objects and layers, each of them contributing to higher quality and accuracy of information. The server operates in a heterogeneous environment where not all the clients can take full advantage of the

highest quality of objects due to the constraints in the hardware of their devices, network, and access methods.

In P-P mode, when bandwidth sharing is enabled, the system allows a host to act as an application-layer gateway and share its connection to the Internet with other hosts. When a peer is unable to access the Internet, it may request other peers to act as gateway. Alternatively, hosts can buffer their messages, if they do not have Internet access. When message relaying is enabled, a host forwards its queued messages to another peer. Hosts also relay all their messages when they gain Internet access (via a gateway or base station). We use the term *gateway* to refer to a base station or stationary server that provides wireless Internet access. 7DS restricts the number of times it forwards a message to a gateway or another relay host.

The motivation of P-P mode is to exploit host mobility, better utilize the wireless throughput, and reduce the average delay that a message experiences until it reaches the Internet or acquires some information.

1.3 Contributions

In this thesis, we analyze the problems posed by the above challenges. We introduce a new general framework for mobile wireless data access. We design, implement, and evaluate several aspects of its performance with respect to the different mechanisms of cooperation it provides. We discuss in more detail the contributions in the following areas, namely information dissemination and message relaying (Section 1.3.1) and bandwidth sharing (Section 1.3.2).

1.3.1 Information dissemination and message relaying

In this new framework, we address the effect of wireless coverage range, density of devices, query mechanism, type of cooperation among hosts and their power conservation strategy on data dissemination. For example, we analyze how fast the information spreads in such setting if all nodes are cooperating with each other, and how the performance of data dissemination changes when only a few nodes cooperate (e.g., the 7DS-enabled servers). The performance of data dissemination is defined by the percentage of the nodes that acquire a data item over time, and the average delay that a node experiences until it receives the data. We compare the server-to-client and the peer-to-peer approaches and evaluate how the wireless coverage range, energy conservation, speed, density of devices and servers affect the data dissemination. We also investigate the message relaying, and find the number of times a host should relay a message to another host to reach the Internet. The investigation of these issues can also give insight for the design of a wireless information infrastructure in a metropolitan area.

7DS acquires the data from other peers within its wireless coverage using single-hop broadcast. Due to the highly dynamic environment and the type of information, 7DS does not try to establish permanent caching or service discovery mechanisms. Instead, we explore the *transient aspect of information dissemination*.

In our simulations, we consider variations of the P-P and S-C schemes as well as some hybrid ones. We consider a simple energy conservation mechanism that periodically enables the network interface. During the *on* interval, 7DS hosts communicate with their peers. In its asynchronous mode, the on and *off* intervals are equal but not synchronized. In synchronous mode, the on and off intervals are

synchronized among hosts, although not necessarily equal. We also vary the wireless range of the network interfaces.

We evaluate these approaches by measuring the percentage of hosts that acquire the data item as a function of time and their average delay. At the beginning of each experiment, only one 7DS host has the data item and the remaining hosts are interested in this data item. We also evaluate the probability that a message will finally reach the Internet and the impact of message relaying. We found that the density of the cooperative hosts, their mobility, and the transmission power have great impact on data dissemination. For a region with the same density of hosts, P-P outperforms S-C with no cooperation among the mobile devices. The simulations indicate that the probability a host querying a data object will acquire it by time t follows the function $1 - e^{-a\sqrt{t}}$ when using S-C mode with fixed server and no cooperation among the mobile devices (i.e., FIS). In case of high density of cooperative hosts, the data dissemination using P-P grows even faster.

We also discover two important scaling properties of data dissemination by expanding the area and varying the speed, the density of wireless coverage (i.e., average wireless coverage per space unit) of cooperative hosts, and the density of cooperative hosts (i.e., average number of cooperative hosts per space unit). First, the performance remains the same when we scale the area but keep the density of the cooperative hosts and transmission power fixed. Secondly, for a fixed wireless coverage density, the larger the density of cooperative hosts, the better the performance. In S-C, this implies that for the same wireless coverage density, it is more efficient to have a larger number of cooperative hosts with lower transmission power than fewer with higher transmission power. We can further generalize our simulation results

using these properties. These results can also assist in the design of wireless data infrastructures.

The contributions of this thesis regarding information dissemination and message relaying are as follows:

1. The design and implementation of 7DS, a novel system that enables information dissemination and sharing among mobile hosts in a peer-to-peer fashion.

2. An evaluation via extensive simulations of 7DS and the effects of the wireless coverage range, 7DS host density, querying mechanism, energy conservation, and cooperation strategy among the mobile hosts as a function of time.

3. Synchronous energy conservation, a mechanism that saves substantial energy, without degrading the efficiency of data dissemination.

4. An analytical model for FIS using theory from random walks and environments, and the kinetics of diffusion-controlled processes. The analytical results on data dissemination are consistent with the simulation results for FIS.

1.3.2 Bandwidth sharing

We investigate bandwidth sharing both in S-C and in P-P settings. For the P-P case, we propose an architecture and protocol that enables dual-homed hosts to act temporarily as gateways to the Internet for hosts that experience intermittent connectivity to the Internet. We design a lightweight protocol that discovers a gateway and enables hosts to share their connections to the Internet and enhance their quality of data. Collaborative applications with shared data motivate this system. The hosts, instead of requesting the data independently from each other, cooperate, and reduce

the replicated data. The main benefits of the network connection sharing protocol are:

1. The statistical multiplexing for bursty traffic that results in an increase of the bandwidth utilization of the WAN links.

2. In the case of shared data applications, the system reduces the amount of replicated data it acquires and increases the quality of service of the peers.

3. It can balance the load across gateways.

In the S-C version of bandwidth sharing, we consider a multimedia storage server that serves clients with different capabilities and requirements. We present a scalable multimedia server that provides statistical service guarantees and propose scheduling techniques for video retrieval that exploit the multiresolution property of compressed video streams. We present a novel retrieval technique that takes advantage of layered information and replication to dynamically reallocate the disk bandwidth. We model the multi-disk environment for different degrees of replication and measure the disk bandwidth utilization. We show how the system performs in the case of no replication, partial replication, and full replication as a function of user access skew. In the case of partial replication, only a part of the multimedia objects are replicated. The reallocation algorithm for full replication can double the disk utilization compared to the case of no replication.

1.4 Structure of thesis

This dissertation is organized as follows.

Chapter 2 gives an overview of the main components of 7DS and discusses related work. Chapter 3 describes in more detail the P-P and S-C models and presents simulation results. Chapter 4 discusses the modeling and analysis of FIS using kinetics of diffusion controlled processes. Chapter 5 introduces a form of bandwidth sharing in a wireless LAN, called network connection sharing, presents the architecture, and its performance evaluation. Chapter 6 discusses bandwidth sharing in video-on-demand servers. Finally, in Chapter 7, we summarize our results and discuss directions for future work.

Chapter 2

7DS Architecture for information sharing

This chapter focuses on the 7DS, an architecture and set of protocols that enables information sharing among peers. First, we describe the communication, cache and power conservation protocol, and its implementation. Then, we discuss mechanisms that stimulate cooperation among peers and provide security. Finally, Chapter 2.4 describes related work.

2.1 System architecture overview

We assume that 7DS is composed of mobile hosts that have a network connection to access the Internet, e.g., via a wireless modem or a base station, and are also capable of communicating with other hosts via a wireless LAN (e.g., IEEE 802.11). 7DS runs as an application on mobile hosts and communicates with other 7DS participants via a wireless LAN. We focus on information access from the Internet that takes place

through retrieval of data objects identified by URLs. When such access fails (for example, due to the loss of the Internet connection), 7DS tries to acquire the data from other 7DS peers. Figure 2.1 illustrates how 7DS operates. Mobile host A tries to access a data object (e.g. a web page). The local 7DS instance running on A detects that the host cannot listen to the Internet and tries to access the page from the peers in close proximity via the wireless LAN. Mobile host D has walked way and cannot listen to the query. Both hosts B and C receive the query. Host C has a copy of the data in its cache, and responds to A's query by sending the data.

2.1.1 7DS Messages

A 7DS query can be a request for a web page or a search with keywords specified by the user. 7DS uses three types of messages to communicate with other peers: queries, reports, and advertisements. A query consists of a set of attributes and their values, such as the URL of the web page, and the MAC address of the host that generated it. These two attributes, the URL and the MAC address, are also used as the query identifier. The system forms queries based on the URL of the data object it tries to acquire. 7DS maintains a query list, which also includes the URLs that the system predicts the user will visit in the next few hours. It multicasts these queries periodically via the wireless LAN to a predefined multicast group.

7DS may use different multicast groups for different queries. It determines the appropriate group in each situation either by hashing the URL of the requested data item, or by using application-specific criteria. In order to conserve more power, a host may listen to a subset of these groups depending on the data objects it is willing to share.

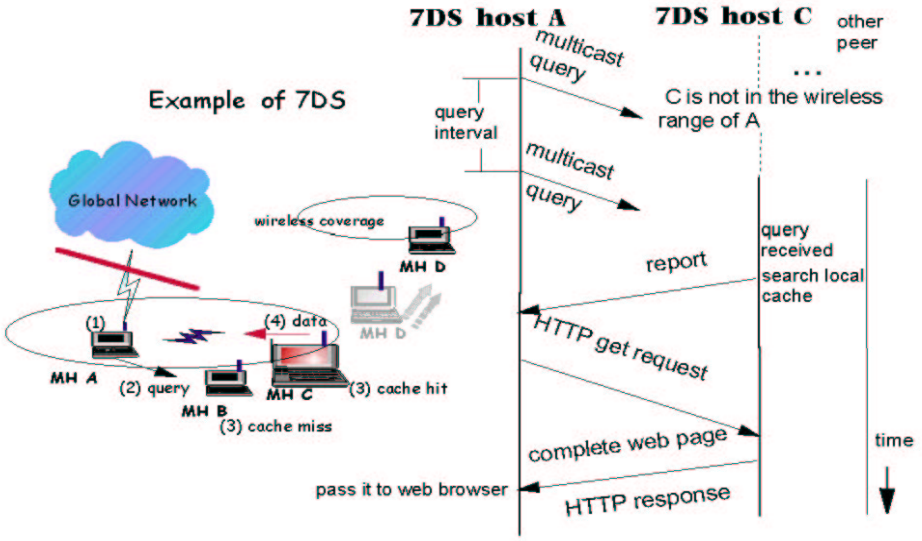

Figure 2.1: Example of information sharing using 7DS. The arrows show the message exchange for the 7DS communication. The ellipse denotes the wireless coverage of the associated host, the shaded signal of the wireless LAN and the non-shaded one of the (lost) connectivity to the Internet.

In both the prototype and simulations, we consider single-hop multicast, using the "ad hoc mode" of IEEE 802.11. After receiving a query, each 7DS peer searches its cache. If a host finds a match, it forms and broadcasts a report. The report describes the relevant data. After a defined interval, the querying 7DS host selects among the received reports the most relevant ones based on application-specific criteria, and then it initiates an HTTP GET request to the chosen host.

Advertisements are application-specific messages that announce the presence of 7DS-enabled servers. Power-constrained devices use a "passive" mode for participating in the system. In particular, they participate only when the expectation for data availability is high, such as when they receive an advertisement. A 7DS-enabled server periodically broadcasts such advertisements with an index of the information or a description of the application it supports. A 7DS host in passive mode sends the query directly to the server from which it received the advertisement. We call this "passive" querying, as opposed to active querying that takes place periodically until 7DS receives the data.

We use XML to describe 7DS messages. 7DS extracts the metadata from the queries received from other peers and performs an attribute-matching search in its local cache. The report includes an identifier that matches the identifier of its corresponding query, and a data description. The data description field contains the relevant information in the local cache of the peer that responds. The report message also contains some optional attributes with their values. These may include the original URL, the time the object of the data description field was cached locally, the time the original copy was created, its HTML title, size, and format. They may also include the quality of the wireless transmission (using the signal-to-noise ratio

value), the author, language, size, and content type of the object. Some of this information is inherently provided by web objects, while others require additional (application-specific) meta information.

The 7DS software displays the reports to the owner of the mobile host, or issues an HTTP GET request automatically (via the web client), using the local URL of the selected report to receive the complete object. Each 7DS node runs a miniature web server, which responds to the HTTP GET requests. The primary information propagation occurs through the use of caching rather than reliable state maintenance. It is not a goal of the current prototype to resolve inconsistency among copies of a data object. 7DS peers may have several objects matching a single query.

2.1.2 Cache management

7DS organizes and indexes the cache. Through a GUI, the user can view, browse, and manage the cache. In the current prototype, the content of the cache is displayed in a tree-like structure (Figure 2.2). We are extending the GUI to support grouping of the cache content by predefined categories and adding a search tool using the meta-data attributes of the stored objects. The user can set the access permissions for files and directories in the cache and specify the objects to be shared with other peers. To protect the user's privacy, the system only transmits reports or pages that correspond to publicly available objects. 7DS can encrypt a private object before transmission. Periodically, 7DS removes expired objects, updates the index with these changes, and also includes newly cached objects. In addition, the system may try to prefetch expired objects if specified in a user profile. Through the GUI, the user marks which pages need to be prefetched when they expire.

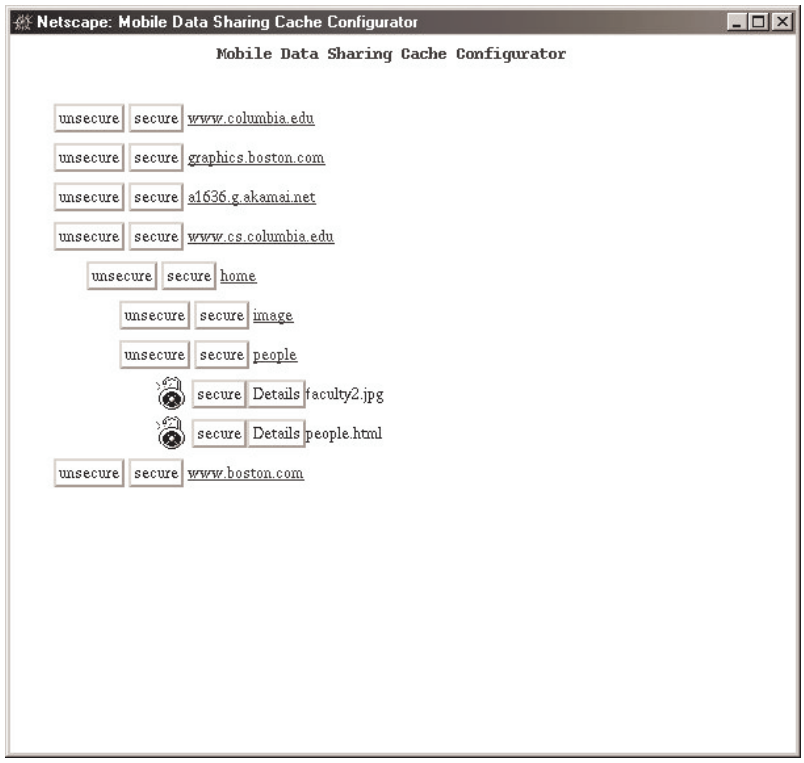

Figure 2.2: The cache manager GUI for setting up the permission of the cached objects for sharing with other peers.

2.1.3 Power conservation

Using a battery monitor and a power management protocol, 7DS aims to adapt communication to reduce energy consumption during idle periods, when there is low expectation for data or collaboration and when the battery life is below a threshold. Generally, prediction of data availability is a hard problem. To predict this, we currently use advertisements from the 7DS servers. When energy conservation is enabled, the mobile host periodically turns off its wireless LAN interface. The system can also adapt its communication with other 7DS peers by tuning several "thresholds" in the battery level. For example, it may set three values of the battery level: when the

battery level is above the highest value, the system can fully collaborate. Between the highest and second-highest value, the system only partially participates in the system. Below the third value, the system stops participating in the 7DS network. Usually, the degree of participation depends on the querying (active or passive, frequency interval) and type of collaboration (data sharing and forwarding support). 7DS is engaged entirely in the participation when it is in both active and passive modes, and supports data sharing and forwarding. In partial participation, the 7DS disables forwarding and switches from active to passive querying. The default setting is as follows: for battery levels above 75%, 7DS uses data sharing and active querying. Between 50% and 75%, it switches to passive querying. When the battery level falls bellow 50%, it stops participating in the system. The user can change this setting via a GUI.

In the second mode of energy conservation, 7DS nodes alternate between the *on* and *off* states of the network interface. During the interval that the network interface is on, 7DS communicates with the other hosts by sending queries, forwarding or receiving reports or data. The mobile host broadcasts a query at each *on* interval until it receives the data. In addition to that, the system can enable the *synchronous energy conservation* option. A group of cooperating hosts decides on a time interval to communicate, potentially with encrypted messages. These hosts can turn their network interface on and start participating in 7DS only during the agreed-upon time interval. We discuss and evaluate the synchronous energy conservation method in more detail in Chapter 3. This rendezvous-based approach can be used by peers to avoid malicious devices and conserve more power.

The protocol has also appeared in [74].

2.2 Encouraging cooperation

The communication and caching components are simple. However, the paradigm is powerful and triggers many challenging issues. It allows us to address some general questions on the performance of information dissemination among mobile devices, which is one of the main points of focus of this thesis (Chapter 3). It also triggers several design issues related to mechanisms that stimulate cooperation among peers, provide security, and efficiently utilize the wireless throughput and energy of the devices.

In peer-to-peer systems, systems willingness to cooperate is crucial. There are settings where hosts are naturally motivated to cooperate, since they belong to users or an infrastructure with common goals. For example, consider a setting in a corporation, a conference, a rescue operation, a home network, or a wireless networked vehicular environment. In these settings, we expect a number of users or wireless devices that share data and collaborate. However, other users may have less incentives to cooperate, especially when the devices are energy-constrained. Selfish users may give false promises about relaying messages or avoid responding to queries to save power. Malicious users may bombard the network with queries to drain the energy of other devices and/or prevent them from utilizing the bandwidth of the wireless LAN. As we mentioned previously, hosts can use the rendezvous-based approach to avoid a malicious user who keeps sending queries. There are ways to stimulate cooperation by providing incentives, financial rewards, or token-based mechanisms. The main motivation for these mechanisms is twofold:

1. Prevent denial-of-service attacks and devices from overloading the network;

2. Stimulate cooperation.

We discuss these mechanisms in Chapters 2.2.1 and 2.2.2.

2.2.1 Preventing denial-of-service attacks

A typical method of preventing denial of service attacks is "challenging" the host using *hash cash* [5]. When a 7DS host (e.g., host R) receives a query, before responding, it multicasts a challenge to the querier (e.g., host Q). This challenge forces Q to execute a non-trivial computational task (e.g., to discover the input in a hash function given the output and a part of the input), before the actual 7DS resource sharing takes place, as in Figure 2.3). By challenging the querier to spend some energy with each query, the system penalizes malicious users for overloading the network with queries. A potential problem arises when a responder cooperates with the malicious querier (e.g., by sending "trivial" challenges) or when the querier itself sends "trivial" challenges. The protocol can force responders to sign their message. In that way, other hosts in the wireless LAN can verify the source of the challenge.

1. Q sends query
2. R receives the query
3. R waits for a random time interval T
4. if no challenge for Q was multicast during T, R challenges Q
5. Q sends its response
6. R verifies Q response to the challenge

Figure 2.3: Responder R challenges querier Q to prevent denial-of-service attacks.

2.2.2 Micropayment mechanisms

The provision of security in 7DS becomes challenging due to its offline nature, the lack of a trustee entity (such as a server) and the power constraints of the devices.

Next we discuss two candidate micropayment mechanisms for 7DS, namely *electronic checks* (e-checks), and a *token-based* approach. We assume that the 7DS multicast query is free, but hosts pay to receive the complete data objects after selecting a report (as described in Section 2.1). In both of these micropayment mechanisms, nodes remunerate each other for the services they provide to each other. In this section, we only briefly describe the main ideas of these approaches. In the e-check approach, there is no need for trusted hardware. For that, we use the micropayment by Blaze *al* described in [9]. The token-based approach requires a tamper-resistant hardware module in each device for the management of tokens and cryptographic coding of messages. The use of tamper-proof hardware will increase the cost of hardware and the energy expenditure of the mobile device. It is part of future work to design them in more detail and evaluate them. In particular, we plan to investigate the trade-off between the robustness of the solution and its efficiency (computational complexity) and the anonymity requirements. The 7DS cooperation should not require complex cryptographic protocols and heavy computational effort that exceed the value of its service. More information on electronic micropayment schemes can be found in [19].

For both approaches, we describe the protocol that takes place between two hosts (e.g., Q and R), which includes an authentication, a micropayment, and an information exchange mechanism. We assume that a querier (host Q) has multicast a query, and host R has responded by sending a report with the relevant data in its

local cache. In its report, R also describes the amount of payment required to send the complete data.

2.2.3 Electronic checks

Let us give a brief description of the micropayment method in [9] and how we use it with 7DS. Hosts sign up for 7DS with a trustee entity or "bank". They get an amount of virtual currency as an electronic check (e-check) from that bank. The bank has an account limit for each host. Therefore, losses from uncollectible transactions are limited. The system with the e-check payment mechanism does not try to prevent losses. As is typical of credit models, we assume that there is a risk factor and the system can tolerate some loss. E-checks are cryptographically bound to the transaction, which prevents the forgery by another host that overhears the exchange of an electronic check. A public-key credential-based architecture is used. The bank acts as a trusted third party that can authenticate each other offline using appropriate credentials. Each host has its own public key, which is encoded in the credential, along with some restrictions. To minimize losses, the credentials are short-lived and thus frequently refreshed. 7DSs can download new credentials when hosts access the Internet. The bank can limit the amount of micropayment a host may send to a given host during a period of time. The number of credentials the bank issues to a host depends on its 7DS usage pattern, service, and the trustworthiness of the host. It is a future goal to investigate the tradeoff of reducing the loss and avoiding disruption of cooperation. The bank does not give new e-checks or extend the credit line to non-trustworthy hosts.

The two hosts Q and R authenticate each other and then verify each other's

1. R sends its credentials
2. Q verifies that R is known to the bank and it is authorized for 7DS
3. Q sends an e-check
4. Q waits for some time for the data from R, before sending a NACK to it
5. R verifies that e-check is genuine
6. if the e-check is genuine, R stores the e-check and sends the data to Q
7. If R receives a NACK from Q, it resends the data to Q

Figure 2.4: Electronic check payment for responding to a query: verification of credentials and e-check and information exchange.

capabilities: Q verifies that R is known to the bank and is authorized to charge Q's account for the particular type of transaction mentioned in the report. R verifies that Q is authorized by the bank to proceed with the specific transaction. When a transaction is completed, Q receives the web page and R receives an e-check from Q. The e-check is encoded as credentials that authorize payment for that specific transaction. Q creates the credential signed with its RSA key and sends it along with its credential to R. The credential contains information such as time issued, which can prevent double-depositing of the e-checks by R. To limit this risk, the system can constrain the amount of payment per responder during a time interval (e.g., refreshing time). Figure 2.4 describes the e-check payment for data between R and Q. There is no guarantee that R will transmit the data to Q after receiving Q's e-check.

The communication between the bank and the hosts can take place using established cryptographic protocols, such as IPsec. Periodically, hosts provide their collected e-checks to the bank. The bank uses this information to verify the transaction and to update the relevant accounts, that is, to increase R's account and to decrease Q's. The bank uses the same verification method that R used to check Q's

credentials. Also, the bank generates short-term credentials for the host over the secure link, with a new public key being refreshed every time.

An advantage of the e-checks is that there is no need for trusted hardware. On the other hand it limits cooperation (querying) with the account limit, the expiration of the e-checks, and the frequency of contact with the bank for uploading the received checks and getting new ones for use. The e-check system is designed to tolerate manageable losses, rather than preventing them. It does not provide anonymity.

2.2.4 Token-based micropayment approach

The token-based mechanism assumes the existence of tamper-proof secure hardware and a trustee agent that distributes some virtual currency or *tokens*. The secure device prevents the user from double spending. This mechanism was inspired by Buttyan *et al* [12], who they describe a similar protocol for loading tokens (or nuglets) for relaying messages in a mobile ad hoc network.

Hosts register with the trustee agent or "bank" and receive a number of tokens that they stores in their "purse". Tokens come in a single "denomination" and have no actual monetary value. The purse is a counter that resides in the secure hardware and indicates the wealth of the (7DS) host. 7DS systems use these tokens to pay hosts that respond to their queries. In order to prevent a node from illegitimately increasing its own counter, the counter is maintained by a secure module, i.e., a trusted and tamper-resistant hardware module in each node. The tokens that are loaded into the packet are protected from illegitimate modification and detachment from their original packet by cryptographic mechanisms.

We use the public key infrastructure with public key certificates to verify the

public key of a peer. In its secure module, each host keeps its own public and private key, a public key certificate from a certificate authority, and the counter.

The micropayment and data exchange take place after the querier successfully responds to the challenge. We use an authenticated key agreement protocol to establish a shared key between two hosts that want to run this micropayment mechanism, such as the authenticated Diffie-Hellman or Station-to-Station (STS) protocol [22]. Each time a host wants to send a query, it runs the STS protocol, so the parties' key pairs can be generated anew. The public keys are certified so that the parties can be authenticated. The STS protocol expires at some time, so for each query the hosts need to rerun it. An STS channel is established between the secure modules of R and Q. A shared key is generated between the two hosts. This shared key will be used to encrypt all messages exchanged between the two hosts. Using this secure module, the system prevents the host from double-spending.

On its secure module, querier Q runs the following operations (described in Figure 2.5) for requesting the complete data object. The responder R runs the steps described in Figure 2.6 on its secure module.

1. Return warning if counter is not sufficiently loaded
2. Verify R's public key certificate, if valid continue
3. Form query
4. Insert query in pending queries list
5. Send query to R
6. If no data sent for pending queries within a defined time interval,
   decrease counter and send NACK
7. If data received for pending query, decrease counter, send ACK

Figure 2.5: The above steps are run on the secure module of the querier Q.

1. Verify public key certificate. If valid, continue
2. Form response with data
3. Send data
4. If ACK received increase counter
5. If NACK received, increase counter and resend data

Figure 2.6: Operations running on the secure module of the responder R.

2.3 7DS information sharing system implementation

The prototype is written in Java. Initially, we used the Glimpse search engine [38]. Glimpse was a performance bottleneck, so we replaced Glimpse with Lucene [61]. Lucene provides incremental indexing, persistent and non-persistent operation, built-in lexical analyzer, and a small heap.

We have implemented a prototype on Linux, and also imported it on Windows and iPAQ. Details of the implementation can be found in [2].

The size of the 7DS prototype on Linux is 38089 lines of code (Table 2.1). Most of the HTTP client was not ours, but we have modified it. We used the following jar files: collections.jar, HTMLParser.jar, lucene-1.2-rc1.jar, xerces.jar, and xml4j.jar.

The system can specify the transmission power in dbm via *iwconfig*. Not all of the wireless cards support a settable transmission power. We use the Agere System Orinoco card that supports five levels, but other cards in the future should be more flexible.

7DS component	Number of lines
Caching	3282
GUIserver	2182
HTTPClient	24266
HTTPMethods	395
Misc	395
ProxyServer	1198
Startallservers	316
UDPMulticast	766
UDPUnicast	241
Webclient	342
WebServer	2205
Total	35588

Table 2.1: 7DS prototype on Linux.

Denis Abramov and Stelios Sidiroglou-Douskos implemented a major part of the 7DS prototype [2].

7DS sources and binaries are available at http://www.cs.columbia.edu/~maria/7ds/.

2.4 Related work

Napster [64] and Gnutella [40] are two systems that explore the cooperation among hosts and enable data sharing among users in a fixed wired network. The first focuses on sharing music files; the latter on any type of file. In contrast to Gnutella, a $7DS$ host does not need to discover its neighbors or maintain connections with them, but only multicasts its queries to a well-known multicast group. In addition, 7DS (in the default mode) restricts the query propagation to the wireless LAN. Unlike Napster, $7DS$ operates in a distributed fashion without the need for a central indexing server. Moreover, Napster requires user intervention for uploading files, whereas $7DS$ does

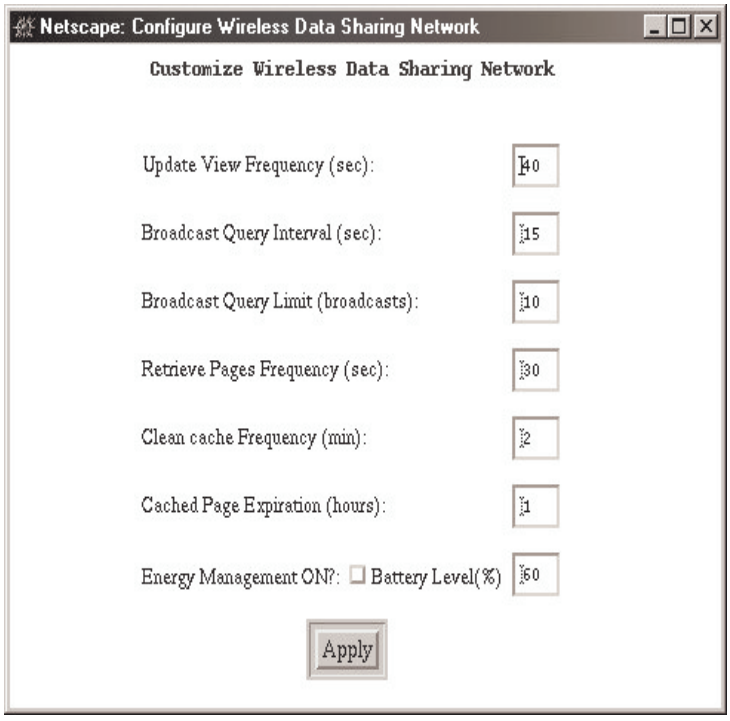

Figure 2.7: 7DS configuration. The user can change 7DS parameters via this GUI. For example, he can set the frequency that a query is broadcast (BroadcastQueryInterval) to 15 sec, or for web pages without any specified expiration field, the user can set a default one.

this automatically. Furthermore, our setting is orthogonal to the service discovery in the wide area network. In service discovery, there is typically an infrastructure of cooperative servers that create indices to locate data based on the queries and the content of the underlying data sources of their local domain [14].

Ad hoc and sensor networks typically assume a relatively high density of devices that results in a connected network, a host can access other hosts via multi-hop routing [11, 85, 103, 30]. They also assume cooperative nodes, part of the same infrastructure, that relay packets on behalf of other nodes. On the other hand, a $7DS$ network is rarely connected, and it can take minutes for one $7DS$ node to come in close proximity to another. As we mentioned, in our setting, peers have different

capabilities and cooperation strategies and they are not necessarily all cooperating with each other. Both in ad hoc and sensor networks, the emphasis is on routing protocols.

Infostations were first mentioned by Imielinski in the DataMan project [78]. Badrinath was among the first to propose an infrastructure for supplying information services, such as e-mail, fax, and web access by placing infostations at traffic lights and airport entrances. Infostations use a single server/multiple clients model in which the server broadcasts data items based on received queries. They mostly address issues related to efficient scheduling algorithms for the server broadcast that minimize the response delay and power consumption of mobile devices and efficiently utilize the bandwidth of the broadcasting channel [50, 78, 7]. Imielinski *et al* [50] investigate methods for accessing broadcast data in such a way that running time (which affects battery life) and access time (waiting time for data) are minimized. They demonstrate that providing an index or hashing based access to the data transmitted over the wireless can result in significant improvement in battery utilization. Barbara *et al* [7] propose and study a taxonomy of cache invalidation strategies and study the impact of clients' disconnection times on their performance.

In a context similar to ours, prefetching targeted for mobile users in a wide-area wireless network has been used in [104]. Tao Ye *et al* [104] consider an infostation deployment. They consider data representation in different levels of detail. Their prefetching algorithm uses location, route, and speed information to predict future data access. Their emphasis is on devising and evaluating techniques for building network-aware applications. They describe an intelligent prefetching algorithm for a map-on-the-move application that delivers maps, at the appropriate level of detail, on

demand for instantaneous route planning. When a mobile user enters the infostation coverage area, it prefetches a fixed amount of k bytes that corresponds to a map with a certain level of detail, where k depends on user speed. They investigate the effectiveness of infostations as compared to a traditional wide-area wireless network. There are two main differences of their setting with our FIS based schemes. First, in their environment, mobile clients are constantly connected to a low-speed wireless network. Devices use a high bandwidth link when they are within infostation coverage. Outside these regions, their requests are passed to the server via a conventional cellular base-station. In our case, the mobile hosts have no wide-area network access. Second, they investigate the effectiveness of (fixed) infostations compared to a traditional wide-area wireless network. For that, in their simulation study, they vary the infostation density and its coverage. In our case, we consider a fixed infostation (i.e., FIS) in the region of 1 km x 1 km, corresponding to low infostation density.

As we explained in Chapter 1, our focus is to investigate a different data access method, namely, peer-to-peer data sharing among mobile users. For its evaluation, we compare it to the access via an infostation. Also, we vary several parameters that have not been investigated in [104], including various mobility patterns, power conservation methods, and querying schemes. Their qualitative result, that having many infostations covering small ranges is a better topology than having few infostations covering large ranges, is consistent with ours.

Another project with similar goals to ours is Portolano [29]. They also aim to provide service discovery to mobile clients with intermittent connections. Their research exists in a hybrid world where they plan to leverage a wired infrastructure in addition to wireless links. It appears that their emphasis is on user interfaces that

allow mobile clients to discover the semantics of any service, and present an interface suited to the client's needs and resource limitations.

Kravets *et al* [57] present an innovative transport level protocol that achieves power savings by selectively choosing short periods of time to suspend communication and shut down the communication device. Their system queues data for future delivery during periods of communication suspension, and predicts when to restart communication. This work motivated us to consider schemes for power constrained devices, in which only in cases of high data availability, 7DSs query actively. In Section 3.3, we discuss this in more detail.

There is substantial peer-to-peer work in the file system and OS literature that is relevant, including the Ficus [68], JetFile [44], and Bayou [91] projects. All of them are replicated storage systems based on the peer-to-peer architecture. Ficus is a distributed file system meant for a wide-scale, Internet-based use. It supports replication using a single-copy availability optimistic update policy. Its main focus is on the consistency among the different copies and reconciliation algorithms to reliably detect concurrent updates and automatically restore consistency. Like Ficus, Bayou provides support for application-dependent resolution of conflicts. Unlike Ficus, it does not attempt to provide transparent conflict detection. JetFile requires file managers to join a multicast group for each file they actively use or serve. Our system targets a different environment and addresses different research issues. The primary concern of our work is the effect of the wireless coverage, collaboration strategy, and power conservation method in the data dissemination across mobile hosts, rather than consistent replication.

Flooding and gossiping (a variant on flooding that sends messages only to some

neighboring nodes instead of all) protocols have been also studied extensively. For example, Kulik *et al* [58] present a protocol for information dissemination in sensor networks. In their setting, the sensors are fixed and the network fully connected. They measure both the amount of data these protocols disseminate over time and the amount of energy they dissipate. Their system features meta-data negotiation prior to data exchange to ensure that the latter is necessary and desired, eliminating duplicate data transmissions, and with power resource awareness. They compare their work with more conventional gossiping and flooding approaches.

Grossglauser *et al* [45] show how the mobility can increase the capacity of mobile ad hoc wireless networks. They evaluate the average per-session throughput and asymptotic performance. On the other hand, the main focus of this thesis is on the transient behavior of the message relaying, and the impact of various parameters.

Davis *et al* [21] investigate the message relaying. Their main focus is on the additional storage at nodes as packets are stored, carried, and forwarded to the destination. They impose finite buffer sizes on hosts and investigate different packet dropping strategies. They show that the two drop strategies that perform best (among the ones they consider) are the Drop-Oldest and Drop-Least-Encountered. In the first, the packet that has been in the network longest is dropped. In the latter, the packet is dropped based on the estimated likelihood of delivery. For that, they use information about host location and movement.

Buttyan *et al* [12] consider a geodesic packet forwarding algorithm in order to evaluate micropayment mechanisms for message relaying in an ad hoc network. The geodesic algorithm assumes that the source of a packet knows the geographic position of the destination, its own geographic position, and that of its neighbors.

Before sending the packet, the source puts the coordinates of the destination in the packet's header. It forwards the packet to the neighbor closest to the destination. Each forwarding host performs the same operation. If the forwarding host does not have any neighbor that is closer to the destination than the host itself, then the packet is dropped. In their setting, the hosts are stationary.

2.5 Conclusions and future work

One important feature of our architecture is its easy deployment. The system can use any web browser to display received data information. It is transparent to wired and wireless networks as well as to different information providers that participate in the system. Also, 7DS is flexible enough to support different applications. It is able to form queries and application-specific criteria for the selection of the appropriate cached copies as long as these applications access their data using URLs. Once a user installs the 7DS software, it automatically configures itself with minimal manual intervention; the system does not require any registration for data distribution. The system is resilient to failures and inconsistencies that occur in this dynamic environment. 7DS is resource-aware and tries to utilize the constrained resources efficiently.

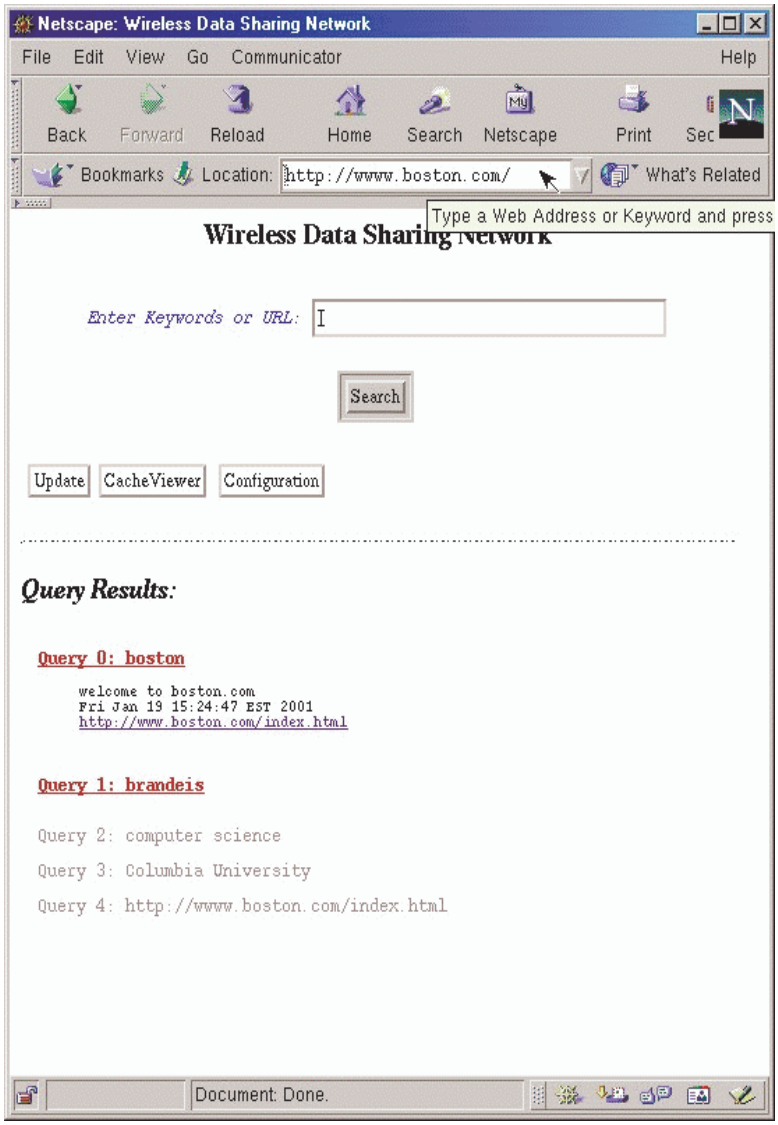

Figure 2.8: 7DS main GUI. In the upper part of the GUI, the user can enter a URL or form a keyword-based query or view the cache manager or configuration. Query results are in the lower part. In this example, queries 2, 3, and 4 are pending, whereas there are responses for query 0 and for query 1. The query 0 is "expanded", i.e., showing the report.

Chapter 3

Performance evaluation of information dissemination and message relaying

We evaluate via extensive simulations 7DS and the effects of the wireless coverage range, querying mechanism, 7DS host density, cooperation strategy among mobile hosts on the information dissemination and message relaying. In this chapter, we present the simulation model and the performance results.

3.1 Introduction

7DS host acquires the data from peers within its wireless coverage using single-hop multicast. Due to the highly dynamic environment and the type of information, $7DS$ does not try to establish permanent caching or service discovery mechanisms. Instead, we explore the transient aspect of information dissemination.

The performance analysis of information dissemination does not appear to be amenable to an analytical solution except for simplified settings with respect to the node layout, mobility pattern, and user interaction pattern. Also, there are no real traces available for the access patterns of mobile, wireless users which would be adequate for our purposes. Thus, to investigate these issues and also assess the efficiency of information dissemination via $7DS$, we perform a simulation-based study. In addition to the simulations in Chapter 4, we present our initial analytical results using diffusion-controlled processes theory. The simulations and analysis are not tied to $7DS$, and provide more general results on data dissemination. Recently, we begun using the actual testbed to measure the performance of the system. Earlier, this was not possible primarily due to cost reasons (e.g., hiring a large number of users to more accurately "approximate" the user's social behavior).

$7DS$ can operate in different modes based on the cooperation strategy among peers (data sharing, forwarding), energy conservation and query mechanism (active or passive querying). To investigate its performance, in particular the effect of transmission power, and the different modes of operation on data distribution, we evaluate P-P and S-C along with their variants. As we describe in Chapter 1.2, P-P and S-C are the two main interaction types among 7DS hosts. In P-P, 7DS hosts cooperate with each other. S-C schemes operate in a more asymmetric fashion: there are cooperative hosts that respond to queries and non-cooperative, resource constrained clients. 7DS hosts can collaborate by data sharing, forwarding messages (such as, "rebroadcasting" queries and data or relaying messages to an Internet gateway), or by caching popular data objects.

In the simulations, we fix the data object. For simplicity, we refer to the 7DS

hosts in these schemes as nodes or peers and the 7DS host that has the data originally in the S-C schemes as the server. At the beginning of each experiment, only one 7DS host has the data item of interest, and the remaining hosts are interested in this data item.

We consider a simple energy conservation mechanism that periodically enables the network interface. During the *on* interval, 7DS hosts communicate with their peers. In its asynchronous mode, the on and *off* intervals are equal (but not synchronized). In synchronous mode, the on and off intervals are synchronized among hosts, although not necessarily equal.

The wireless range of the network interfaces also varies. We evaluate these approaches by measuring the percentage of hosts that acquire the data item as a function of time, and their average delay.

In Chapter 3.4, we also evaluate the probability that a message will finally reach the Internet, and the impact of message relaying. We found that the density of the cooperative hosts, their mobility, and the transmission power have a great impact on data dissemination. For a region with the same density of hosts, P-P outperforms S-C with no cooperation among the mobile devices. The simulations indicate that the probability a host that queries for a data object will acquire it by time t follows the function $1 - e^{-a\sqrt{t}}$ when using S-C mode with fixed server and no cooperation among the mobile devices (i.e., FIS). In case of high density of cooperative hosts, the data dissemination using P-P grows even faster (proportional to $1 - e^{-at}$). For example, in P-P, in a setting of 15 hosts with wireless range of 230 m, after 25 minutes, 99% of the users will acquire the data compared to just 42% of the users in the FIS. For the same average delay of 6 minutes, a host using FIS will get the data with a 42%

probability, whereas using synchronous P, even in a setting of only five hosts per km^2, this probability is doubled. For lower transmission power, P-P outperforms FIS by 20% to 70%. In the case of only five hosts, the two approaches differ by 3% to 43%, depending on the transmission power.

We also present two important scaling properties of data dissemination by expanding the area and varying the speed, density of wireless coverage (i.e., average wireless coverage per space unit) of cooperative hosts, and density of cooperative hosts (i.e., average number of cooperative hosts per space unit). First, performance remains the same when we scale the area but keep the density of the cooperative hosts and transmission power fixed. Second, for fixed wireless coverage density, the larger the density of cooperative hosts, the better the performance. In S-C, this implies that for the same wireless coverage density, it is more efficient to have a larger number of cooperative hosts with lower transmission power than fewer with higher transmission power. We can further generalize our simulation results using these properties. Also, these results can assist in the design of wireless data infrastructures.

The results described in Chapter 3 have also appeared in [74, 75].

3.2 System models and operation modes

To investigate the performance of 7DS, in particular the effects of transmission power and the different modes of operation on data distribution, we evaluate P-P and S-C along with their variants. For simplicity, we refer to the 7DS hosts in these schemes as nodes or peers and the 7DS host that has the data originally in the S-C schemes as the server. In the P-P schemes, all nodes are mobile with active querying enabled. We simulate three variations depending on the type of cooperation, namely data sharing

(DS), forwarding (FW) and both data sharing and forwarding enabled (DS+FW). When forwarding is enabled, upon the receipt of a query or data, 7DS peers rebroadcast it, if they have not rebroadcast another message during the last 10 s. The last condition is a simple mechanism for preventing flooding in the network. For example, host A queries for some data and host B receives A's query. B rebroadcasts A's query, because it does not have any relevant data. Host C receives B's message and rebroadcasts A's query, since it does not have any relevant data. Host B receives the query rebroadcast by C, but it ignores it.

We separate the S-C schemes into the "straight" S-C scheme without any cooperation among clients (namely, FIS and MIS) and some hybrid ones with cooperative clients. In the FIS (MIS) scheme, there is a fixed (mobile) host with the data that acts as a server. The remaining nodes (clients) are mobile, non cooperative with active querying enabled, and without any energy conservation mechanism. They receive data only from the server. The hybrid schemes are with passive querying enabled and fixed server. In passive querying mode, the server sends an advertisement every 10 sec. Hosts send queries upon the receipt of an advertisement.

Model	**Cooperation**	**Options**	**Querying**
S-C	only server, server mobile/fixed (FIS/MIS)	DS (only server)	active
P-P	all hosts	DS, FW, DS+FW	active
Hybrid	fixed server, cooperative peers	DS, FW, DS+FW	passive

Table 3.1: Summary of the schemes with their querying mechanism.

Let us describe the main motivations for the comparisons made in the remaining section. The P-P vs. straight S-C comparison helps to understand the effect of the cooperation among mobile peers. The P-P and MIS vs. FIS show how mobility affects data dissemination. In particular, the MIS vs. FIS comparison investigates

the effect of server mobility on data dissemination.

3.2.1 Model assumptions

Nodes move in a 1000 m x 1000 m area according to the random waypoint mobility model [11]. This random walk-based model is frequently used for individual (pedestrian) movement [11, 85, 103]. The random waypoint breaks the movement of a mobile host into alternating motion and rest periods. Each mobile host starts from a different position and moves to a new randomly chosen destination. For each node, the initial and end points for each segment are distributed randomly across the area. Each node moves to its destination with a constant speed selected randomly from the interval (0 m/s, 1.5 m/s). When a mobile host reaches its destination, it pauses for a fixed amount of time, then chooses a new destination and speed (as in the previous step) and continues moving. Later in the section, we describe two scaling properties that allow us to show that our simulations are robust and to generalize the results when we expand the area or increase the user speed.

In our simulations we consider a two-dimensional world. In most settings we consider, it is not unrealistic to assume that information dissemination in three dimensions (e.g., among colleagues in a building that occupy several floors) can be viewed as information dissemination that takes place in several independent two-dimension planes due to high signal attenuation.

The query interval consists of an *on* and *off* interval. The broadcast is scheduled at a random time during the *on* interval. The asynchronous mode is the default energy conservation method. We explicitly denote the schemes having synchronous mode enabled with the word "sync". In schemes with no energy conservation, the

off interval is equal to 0 and the *on* and query interval are the same. The exchange of queries, reports, and advertisements takes place during the on interval. Generally, the transmission of the complete data object (such as a web page) is scheduled separately. For example, the dataholder may select a time or "rendezvous point" in which the HTTP transmission takes place, and include it in the report message. At that time, both the querier and dataholder set their network interface on, and the querier initiates the HTTP GET request (as described in Chapter 2). In the simulations, we concentrate only on the exchange of 7DS queries, reports, and advertisements. A cooperative dataholder responds to a query by sending the data item in the report. In this simulation study, we assume one data object, and all hosts in the area are interested in this data item. When a host receives a report for this data item, it becomes a dataholder. This simplification is reasonable in order to investigate the dominant parameters on data dissemination.

A *scenario* (file) defines the topology and movement of each host that participates in an experiment. We consider different number of hosts in the area. We used the RAN2 random generator function from the second edition of Numerical Recipes, and a new random seed for each scenario.

Later in this section, we scale the area and vary the density of the hosts and their wireless coverage. We emphasize that this host density does not necessarily represent the total number of hosts in that area, but just indicates the popularity of the defined data object. By varying the density of hosts, we study how data items of different popularity disseminate in such environment. We speculate that in an urban environment such as Lower Manhattan, near the platform of the train or subway stop in a rush hour, there will be from four to 25 wireless devices (carried by humans or

integrated into physical objects) that would be interested in getting the local and general news using PDAs or other wireless devices. A density of 25 hosts per km^2 corresponds to a very popular data item whereas a density of five hosts per km^2 corresponds to a more typical data object. We generate 300 different scenarios for different density values.

In each of these scenarios, the mobility pattern of each host is created using the mobility pattern we described, except in the FIS-based schemes, where the server is stationary. We run simulations using these scenarios, for the different schemes of Table 3.1.

3.2.2 Proposed model for wireless LANs

The wireless LAN is modeled as an IEEE 802.11b network interface. We use the ns-2 simulator [33] with implementation of mobility and wireless extensions contributed from the CMU Monarch project [1]. The majority of the WLAN products available are proprietary spread spectrum operating in the 900MHz and 2.4 GHz ISM frequency bands. Operation of the WLAN in unlicensed RF bands requires the of spread spectrum modulation to meet the requirements for operation in most countries. Both architectures are defined for operation in the 2.4 GHz frequency band typically occupying the 83 MHz of bandwidth from 2.400 GHz to 2.483 GHz. The raw bandwidth capabilities of the network interface is 2 Mb/s shared by all hosts in the wireless LAN. Hosts in the wireless LAN communicate with each other directly without the need of a base station. The frequency hopping systems in the 2400-2483.5 MHz band employ at least 75 hopping channels, all frequency hopping systems in the 5725-5850 MHz band, and all direct sequence systems: 1 Watt. All other frequency hopping systems

in the 2400-2483.5 MHz band: 0.125 Watts. We consider transmission powers of 281.8 mW (high), $\frac{281.8}{2^4}$ mW (medium) and $\frac{281.8}{2^8}$ mW (low). Assuming the two-ray ground reflection model these transmission powers correspond to ranges of approximately 230 m, 115 m ($\frac{230}{2}$ m)and 57.5 m ($\frac{230}{4}$ m), respectively. In the two-ray model the received power at a distance d is predicted by the formula [80]:

$$Pr_r(d) = \frac{P_t\ G_t\ G_r\ h_r^2\ h_t^2}{d^4} \tag{3.1}$$

where P_t is the power of transmitted signal, h_r and h_t are the heights of receiver and transmitter antenna respectively and G_r and G_t are the gains of signal at receiver and transmitter, respectively. Note that considering the simulation results on these

Number of hosts	**High transmission power**	**Medium transmission power**	**Low transmission power**
1	15	4	1
5	83	20	5
10	166	41	10
15	249	62	15
20	332	83	20
25	415	103	25

Table 3.2: The total wireless coverage density (%), i.e., $N\ \pi\ R_p^2/A$, where A is the area of the plane where N hosts are placed and R_p the radius of the coverage of host for transmission power p (high, medium, low) according to the two-ray propagation model 3.1. For example, medium transmission power $\frac{281.8}{2^4}$ mW corresponds to radius R_{medium} equal to 115 m. Here, A is equal to one square kilometer.

ranges and the scaling properties we describe later, we can generalize the performance of 7DS for different transmission powers.

Parameter	Value
Pause time	50 sec
Mobile user speed	(0,1.5) m/sec
Server advertisement interval	10 sec
Forward message interval	10 sec

Table 3.3: Simulation constants in 7DS.

3.3 Performance evaluation

3.3.1 Measurement of dataholders

We evaluate the effectiveness of our approaches by computing the percentage of nodes that acquire the information after a period of time. In the percentage we do not include the node that has the data item at the beginning of the simulation. We also compute the average delay until a mobile host receives the information from the time it sends the first query. For the computation of the average delay, for each simulation set we consider only the queriers that acquired the data by the end of the simulation and average their delay. We ran the 300 generated scenarios for each test and computed the average of the percentage of hosts that *become* dataholders by the end of each test. These are finite-horizon simulations, so that we do not have to deal with initialization bias. The default simulation time is 25 minutes. However, we also investigate data dissemination over time and vary the simulation time from 150 sec to 50 minutes. The 95% confidence interval for the average percentage of dataholders is within 0-11% of the computed average, with the variance tending to be higher for low host density.

Figures 3.1, 3.2, and 3.3 show the percentage of dataholders as a function of

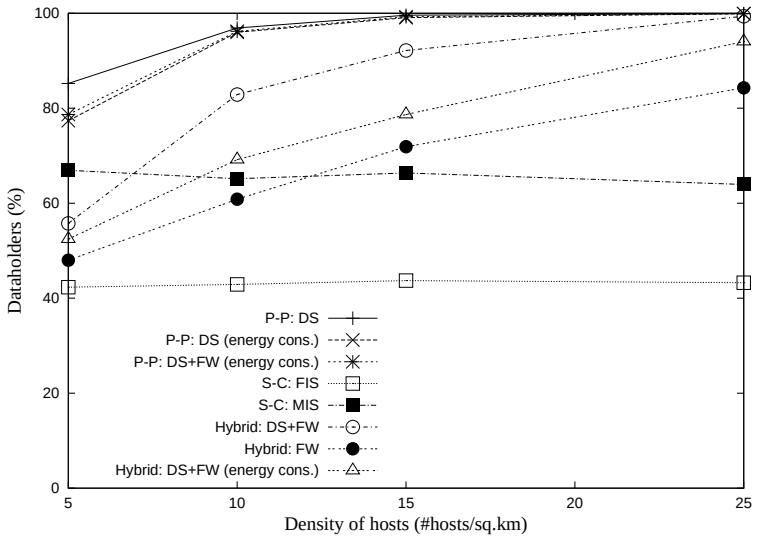

Figure 3.1: Percentage of dataholders after 25 minutes for high transmission power. The query interval is 15 sec.

the density of hosts for P-P and S-C schemes. In this set of simulations, the query interval is 15 sec. For high transmission power, as in Figure 3.1, 7DS proves to be an effective data dissemination tool. Even when the network is sparse, 77% of the users will acquire the data during the 25 minutes of the experiment. For networks with ten or more hosts, more than 96% of the users will acquire the data within 25 minutes. For host densities of 25 hosts per km^2, the probability of acquiring the data is very close to 100%. The P-P vs. FIS comparison illustrates the effect of data sharing among mobile peers. In Figure 3.1, in a setting of 25 hosts, P-P schemes outperform FIS by 55%. In particular, in P-P, 99.9% of hosts will acquire the data after 25 minutes, compare to 42% of the users in the FIS. For lower transmission power, P-P outperforms FIS by 20% to 70% (Figures 3.1, 3.2, and 3.3). The impact of data sharing among peers is also apparent in hybrid schemes. It is more evident

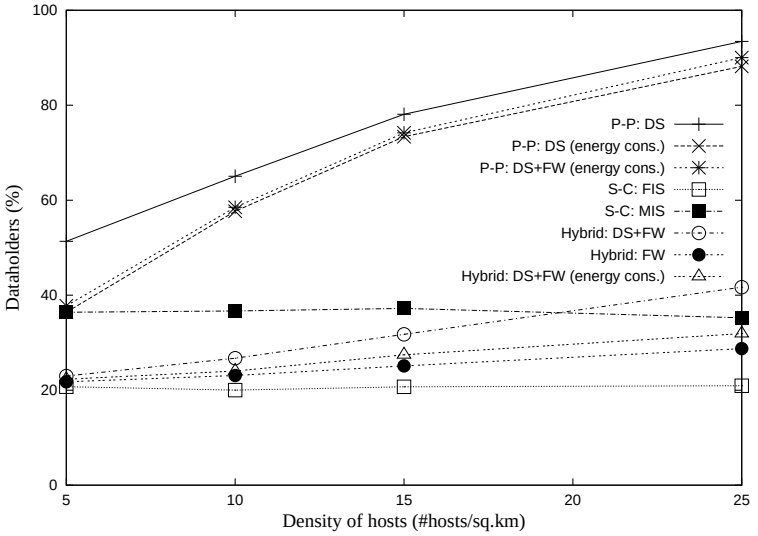

Figure 3.2: Percentage of dataholders after 25 minutes for medium transmission power. The query interval is 15 sec.

in the hybrid vs. S-C schemes for density of ten or more hosts per km^2 and medium or high transmission power.

Note that forwarding in addition to data sharing does not result in any further performance improvements. This is due to the low probability that a case such as the following occurs: There are a querier A and a dataholder C that cannot listen to each other, and a third host B that can communicate with both and forward data. Moreover, A will not acquire the data directly from a dataholder until the end of the test. We emphasize that this is also true for smaller simulation times, starting from 150 sec (just a few seconds after the hosts start querying). Figures 3.4, 3.5, 3.6, and 3.7 illustrate the effect of forwarding as a function of time in two settings of 10 hosts per square kilometer and 25 hosts per square kilometer, respectively. In a more dense setting of mobile hosts that forward messages independent of their own data

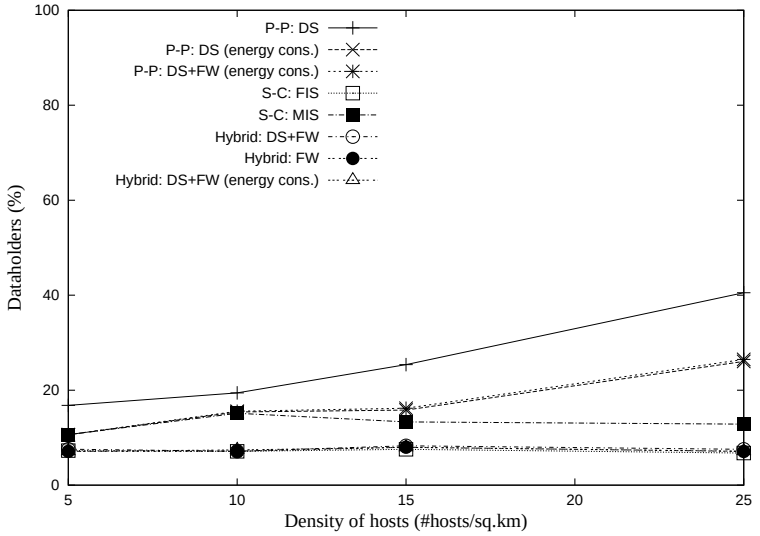

Figure 3.3: Percentage of dataholders after 25 minutes for low transmission power, respectively. The query interval is 15 sec.

interests, we expect forwarding to have a higher impact. As we mentioned earlier, in order to prevent flooding, when forwarding is enabled, upon the receipt of a query or data, 7DS peers rebroadcast it, if they have not rebroadcast another query or data during the last 10 s, respectively. This restricts also the effect of forwarding. The use of a routing protocol among the mobile hosts could potentially enhance the impact of forwarding. The impact of forwarding is more apparent in schemes with data sharing among peers disabled. For example, Figures 3.1, 3.2, and 3.3 show that hybrid schemes with forwarding enabled outperform FIS by 4%-40% depending on transmission power.

As we expect, the performance of both FIS and MIS remains constant as the number of hosts increases, since a data exchange takes place only when a querier is in close proximity to the server. In addition, notice that in Figures 3.1, 3.2, and 3.3,

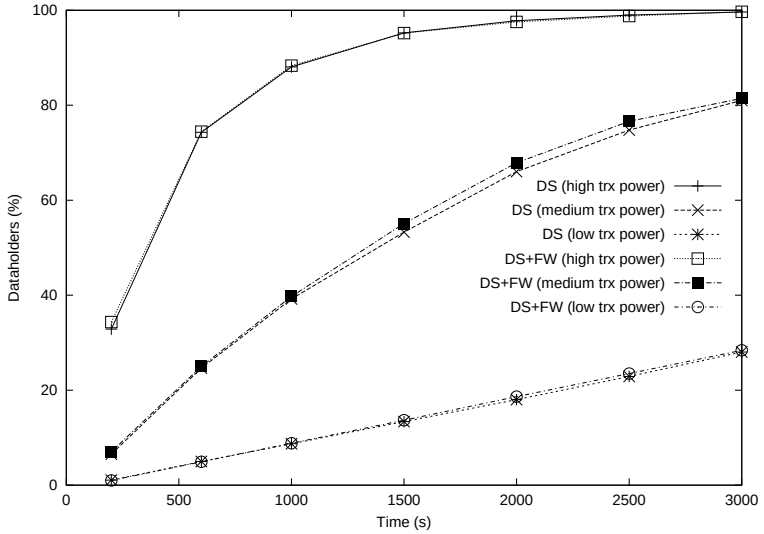

Figure 3.4: Effect of forwarding on peer-to-peer with data sharing enabled (DS). There are 10 hosts (one initial dataholder) per square kilometer.

MIS outperforms FIS by approximately 22%, 16%, and 6%, respectively. An intuitive explanation is based on the fact that, in MIS schemes, the relative speed of the server from the clients is larger than in FIS schemes where the server is fixed. Therefore, the mobile information server will meet with more hosts and disseminate the data faster. On the other hand, as we expect, the density of hosts affects the schemes that are based on peer-to-peer cooperation. As the number of hosts increases from five to 25 hosts, in P-P schemes with medium transmission power, the performance improves substantially.

3.3.2 Impact of energy conservation

Measurements of the energy consumption of the wireless network interfaces have shown that they consume substantial power even when they are idle (powered on but

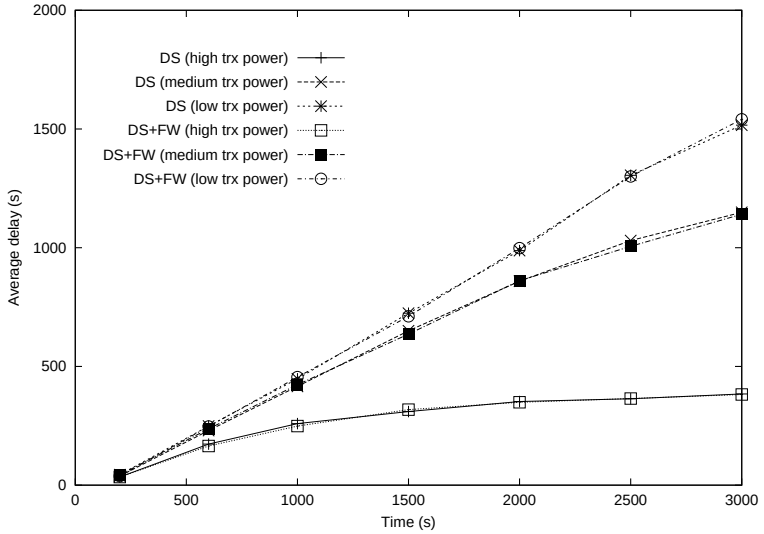

Figure 3.5: Effect of forwarding on peer-to-peer with data sharing enabled (DS). There are 10 hosts (one initial dataholder) per square kilometer.

not sending or receiving packets). Moreover, receiving packets costs marginally more energy than being idle [92]. Using the asynchronous energy conservation mechanism, there is a 50% energy savings, since the network interface is on only half the time. As Figures 3.1, 3.2, and 3.3 illustrate there is some degradation in data dissemination. This is due to the decrease of the time interval the hosts can communicate. If we keep the query interval constant and reduce the *on* interval, the smaller the *on* interval, the higher the energy savings. However, with smaller intervals, the degradation of data dissemination is larger. To prevent this degradation, we enable synchronous mode. In synchronous mode, the on and off intervals of all hosts are synchronized. As we show in Figures 3.8 and 3.9, when the synchronous mode is enabled, even with a small *on* interval, energy conservation does not cause any degradation of the data dissemination. More specifically, Figure 3.8 illustrates P-P schemes with data sharing

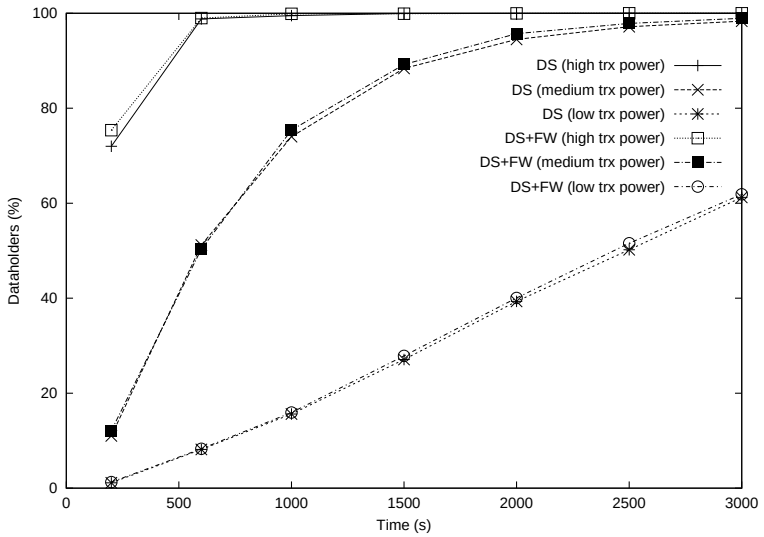

Figure 3.6: Effect of forwarding on peer-to-peer with data sharing enabled (DS). There are 25 hosts (one initial dataholder) per square kilometer.

and Figure 3.9 hybrid schemes with data sharing and forwarding. The query interval is 15 sec, in which, during the first 1.5 sec the network interface is on, and during the remaining time (13.5 sec), it switches off. In an ideal setting without packet losses and need for retransmission, the number of messages exchanged in the P-P schemes without energy conservation and the ones with synchronous energy conservation are the same. Therefore, the power spending on message receiving/sending is the same, whereas the period in the idle state is reduced (the network interface is on for only 10% of the time). The synchronous mode may result in a 90% reduction in energy dissipation. In general, hosts may query for different data items. In very dense settings retransmissions and packet losses may result in further energy spendings. It is part of our future work to investigate further the synchronous power conservation mode and the impact of retransmissions, packet loss, and on interval in such environment.

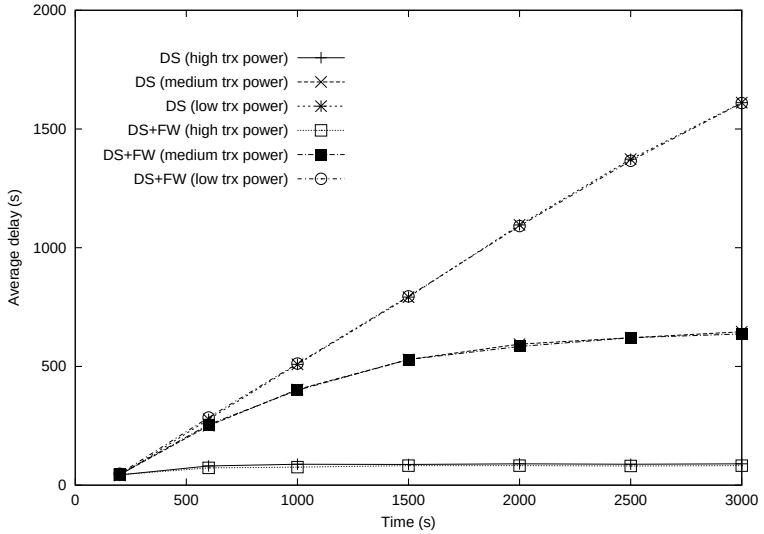

Figure 3.7: Effect of forwarding on peer-to-peer with data sharing enabled (DS). There are 25 hosts (one initial dataholder) per square kilometer.

3.3.3 Impact of query interval

We investigate the performance of the system as a function of the query interval using the asynchronous energy conservation method (i.e., the on interval is half the query interval and is not synchronized). The degradation in the FIS performance is relatively small compared to P-P schemes as the query interval increases. This is due to the high probability that a mobile host that gets in close proximity to a server acquires the data (i.e., there is sufficient time to broadcast a query and receive the data).

Figures 3.10 (a.1), (b.1), and (c.1) correspond to a relatively sparse network of five hosts per square kilometer, whereas Figures 3.10 (a.2), (b.2), and (c.2) correspond to a more dense network of 25 hosts/km^2. As Figure 3.10 illustrates, in P-P schemes the impact of the query interval can be more apparent. In a setting of 25 hosts with

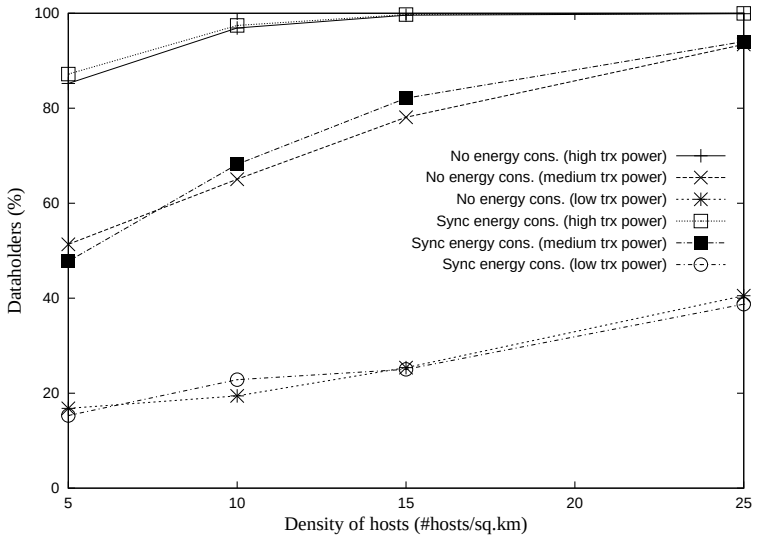

Figure 3.8: The impact of synchronous mode on data dissemination in a P-P with data sharing. Query interval is 15 sec and the on period in "sync" schemes is 1.5 sec. The simulation time is 25 minutes.

medium transmission power, data sharing, and no energy conservation, when the query interval increases from 15 sec to 3 minutes the degradation is approximately 30% and for five hosts, it reaches 50%. However, using the synchronous energy conservation, even when we maintain a low ratio of the on-interval (e.g., 5% with on interval to be 6 sec and query interval 2 minutes), we expect the degradation to be much weaker. We need to investigate further what is the optimal on and query interval, and when a mobile host need to switch to passive querying to utilize its battery more efficiently, taking also into consideration the average time that two hosts are in close proximity, and the traffic in the wireless LAN.

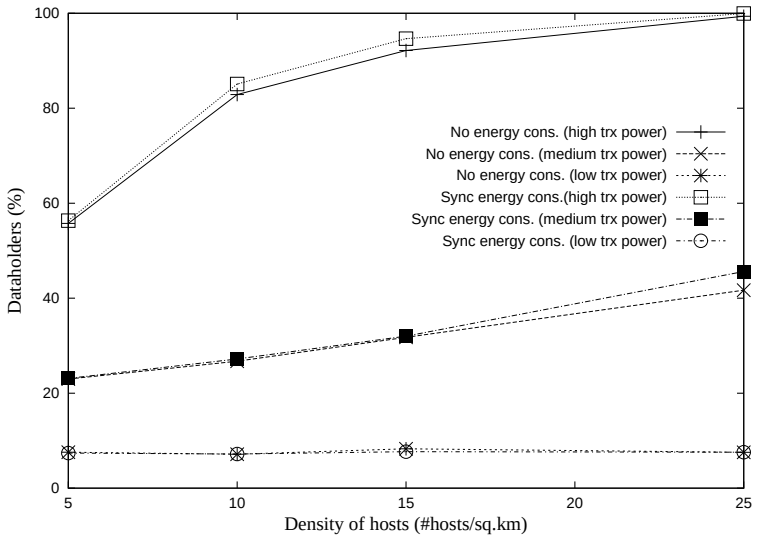

Figure 3.9: The impact of synchronous mode on data dissemination in a hybrid scheme with data sharing and forwarding scheme. Query interval is 15 sec and the on period in "sync" schemes is 1.5 sec. The simulation time is 25 minutes.

3.3.4 Measurement of average delay

As we mentioned earlier, we measure the average delay a host experiences from the first query until it receives the data. For each test, we compute the average delay of the nodes that acquired the data by the end of simulation. Note that for the computation of the average delay, we only consider the hosts that received the data by the end of the simulation. Then, we take the average over all 300 sets, excluding the ones without new dataholders. First, let us fix the simulation time to 25 minutes and compare P-P and FIS schemes in terms of average delay for the same probability of acquiring the data. In P-P with data sharing and no energy conservation, for high transmission power (Figure 3.13), the average delay is as high as 6 minutes for sparse networks and drops to 77 sec for dense networks (Figure 3.13). In the case of low transmission power, it reaches 13 minutes. Using FIS, the average delay is constant

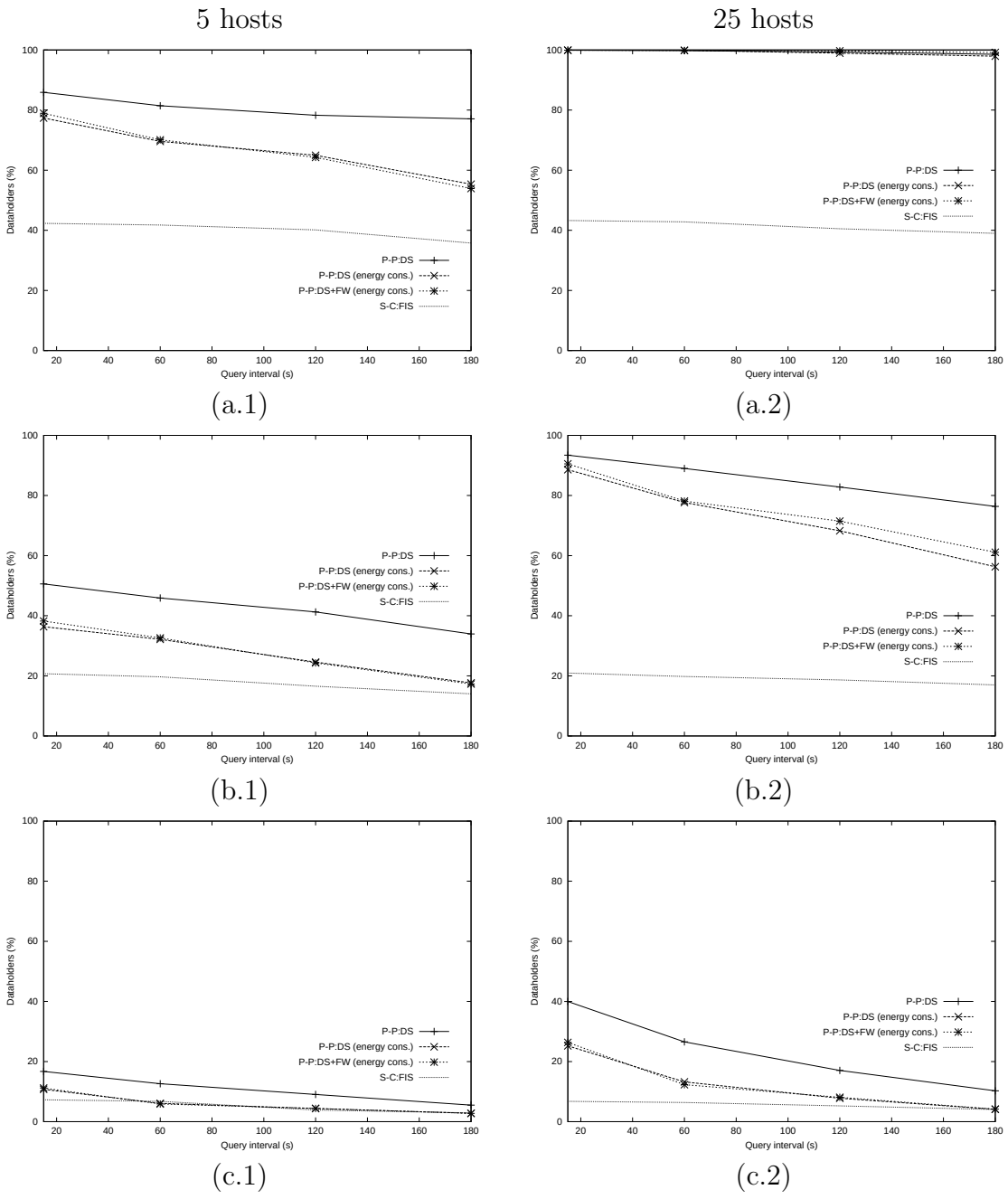

Figure 3.10: Percentage of data holders as a function of the query interval. The first and second column correspond to scenarios with 5 hosts per km^2 and 25 hosts per km^2, respectively. Figures (a), (b), and (c) correspond to a high, medium and low transmission power, respectively. Schemes with energy conservation enabled use the sync mode.

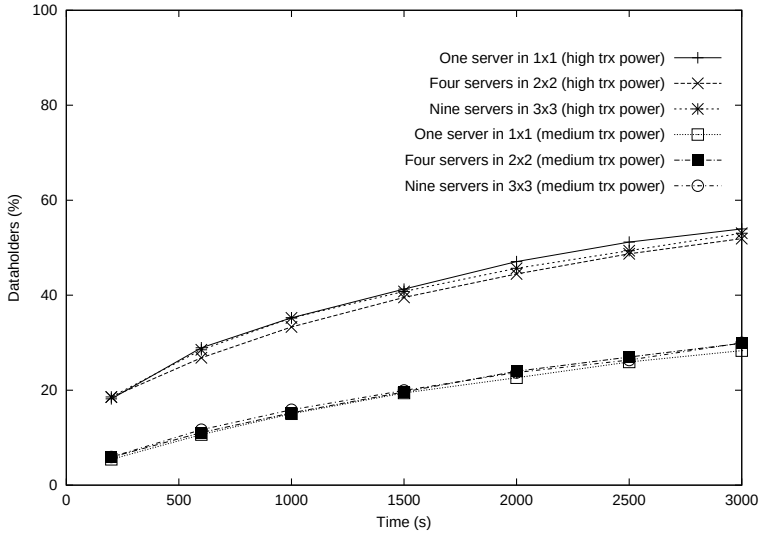

Figure 3.11: Percentage of dataholders of the fixed information server (FIS) schemes as a function of the simulation time. "AxA" indicates the size of the area in square kilometers. For example, the curve with the circle corresponds to a FIS scheme with nine servers per 3 km x 3 km (all hosts transmitting with medium power).

over the number of hosts in the area. For high transmission power, it is 6 minutes, while for low transmission power it reaches 9 minutes. So, (sync) P-P with data sharing, even in the case of low host density, performs better than FIS. For the same average delay to acquire the data (6 minutes), the probability to acquire the data in the P-P doubles. This becomes clear when we compare P-P in Figure 3.8 and 3.13 (five peers with high transmission power) and FIS in Figure 3.14 (e.g., the one server in 1x1 with high transmission power).

For the Figures 3.14, 3.15, 3.16, 3.17, and 3.18, we have combined the simulation results for the probability a host acquires the data, and the average delay it experiences as functions of time. For example, in the case of one server in a 1 km x 1 km area with high transmission power in Figure 3.14, we use the results

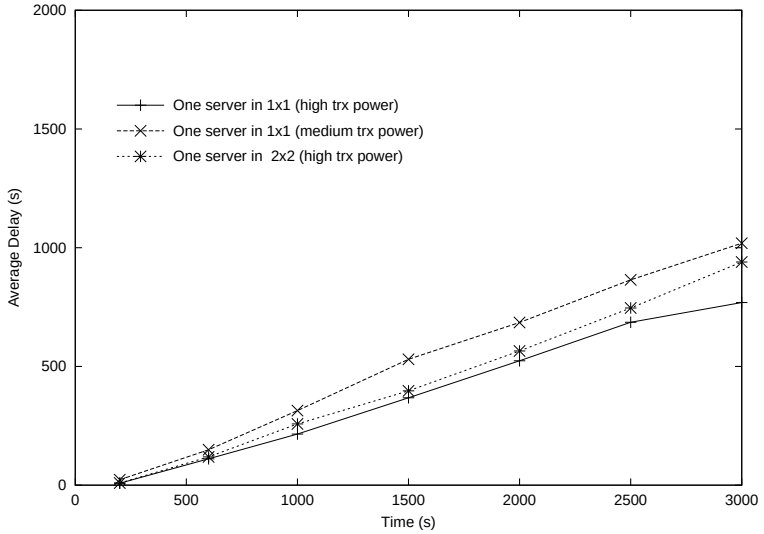

Figure 3.12: Average delay of the fixed information server (FIS) schemes as a function of the simulation time. "AxA" indicates the size of the area in square kilometers. For example, the curve with the circle corresponds to a FIS scheme with nine servers per 3 km x 3 km (all hosts transmitting with medium power).

of Figures 3.11 and 3.12 that correspond to one server in a 1 km x 1 km area with high transmission power. For that, we find in Figure 3.11 at which time t a given percentage of dataholders p is reached, and then in Figure 3.12, the average delay d that hosts who received the data by that time t have experienced (since their first query was sent). The graph in Figure 3.14 consists of these pairs (p,d). The percentage of hosts that acquire the data in P-P with high transmission power reaches 40% with an average delay of 135 sec. With the same delay and using FIS, 30% of hosts will acquire the data (Figure 3.14). With FIS, a 40% probability of acquiring data corresponds to an average delay of 6 minutes (Figure 3.14) whereas using (sync) P-P this probability doubles, even for a low density cooperative host setting (Figure 3.13 and 3.8). For a higher average delay of 10 minutes, 85% hosts will acquire the data

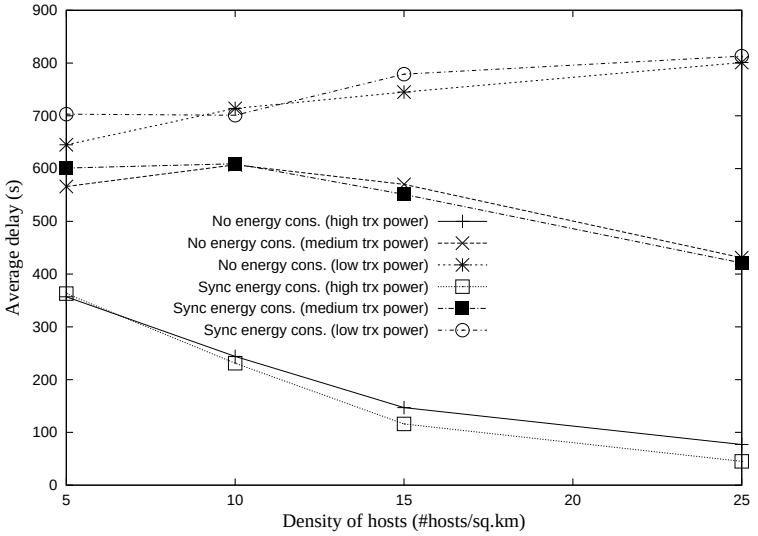

Figure 3.13: The average delay for the P-P with data sharing. Query interval is 15 sec and the on period in sync schemes is 1.5 sec. The simulation time is 25 minutes.

using P-P (Figures 3.23 and 3.25 for five hosts at simulation time equal to 2000 sec), and 50% using FIS (Figure 3.14). In the case of medium transmission power, with an average delay of 315 sec, a host will get the data with a probability of 15% and 22% using FIS (four servers in 2x2 scheme in Figure 3.16) and P-P (one initial dataholder and five cooperative hosts in 1x1 scheme in Figure 3.17), respectively.

3.3.5 Scaling properties of data dissemination

Let us now discuss the scaling properties and generalize our performance results. First, we focus on expanding the area but keeping the movement pattern the same. In both P-P and FIS schemes, when we expand the area but keep the density of hosts and their transmission power fixed, the performance of data dissemination remains the same. This indicates that our simulation results are robust. For example, Figures

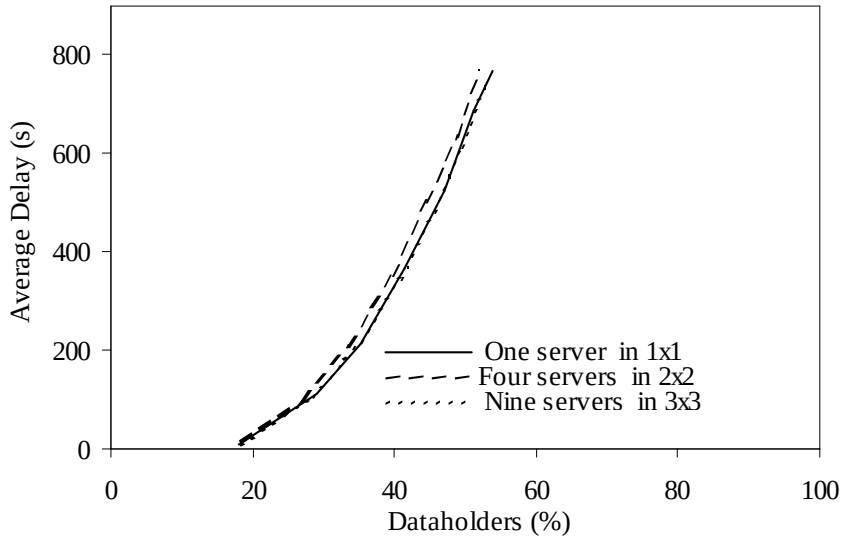

Figure 3.14: The average delay to receive the data as a function of the probability to acquire it in FIS for high transmission power. The "AxA" indicates the size of the area where the server and mobile clients are placed (in square kilometers).

3.14 and 3.15 show this scaling property in FIS, for high and medium transmission power, respectively. Specifically in FIS, it is sufficient to fix only the density of the servers, since only the servers cooperate. Let $p(t)$ denote the probability a host will acquire the data by time t. Figure 3.19 shows the probability that a host will *not* acquire the data by time t, i.e., $1-p(t)$, or *survival probability* on a logarithmic scale. This figure shows the percentage of data holders at time t using the transformation $(\log(1-p(t)))^2$. Their shape indicates that $p(t)$ in FIS follows the $1-e^{-\sqrt{at}}$. In P-P settings (e.g., P-P with data sharing and energy conservation) $p(t)$ grows faster than in FIS. Our simulation results indicate that the P-P with data sharing and energy conservation can be approximated by the function $1-e^{-at}$, especially for less dense settings, such as those with fewer than 20 hosts per km^2 transmitting with high or medium power. In the above function, the constant a depends on the density of FIS servers. For very dense settings, this probability grows even faster.

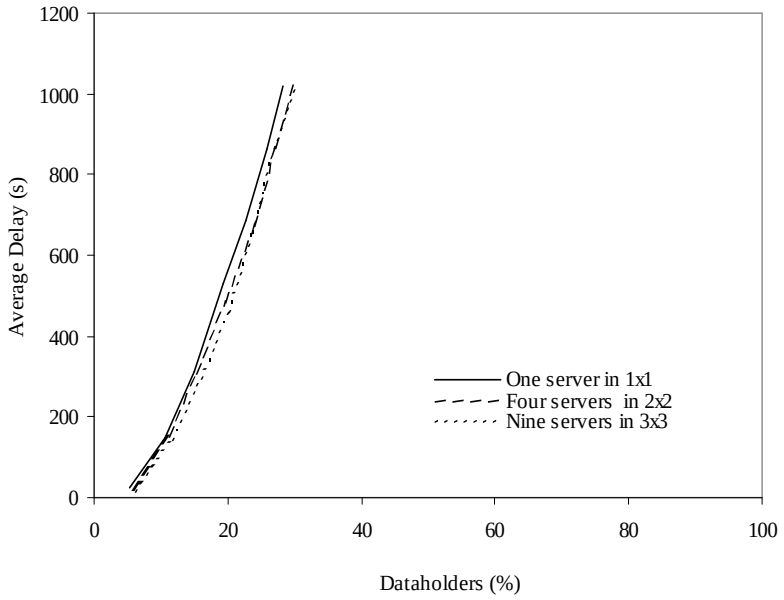

Figure 3.15: The average delay to receive the data as a function of the probability to acquire it in FIS for medium transmission power.

Another important scaling property is related to the effect of density of co-operative hosts vs. their wireless coverage density. Assuming the same total area of wireless coverage, we investigate the impact of host density for both the P-P and FIS schemes. Particularly in FIS, this can be viewed as a design decision. Figure 3.20 illustrates two possible deployments of servers with the same wireless data coverage, assuming an ideal transmission model with the power inverse to the square of distance. The density of servers in the left is higher than that in the right, but they have lower transmission power. For both settings, we assume the same mobility pattern. Figure 3.20 depicts a host moving with fixed speed v and traveling on a line segment during an interval. The setting of the larger number of servers with lower transmission power is more effective in terms of power spendings and wireless throughput utilization. We found that for fixed total wireless coverage, the higher the cooperative host density, the better the performance. Simulations indicate that

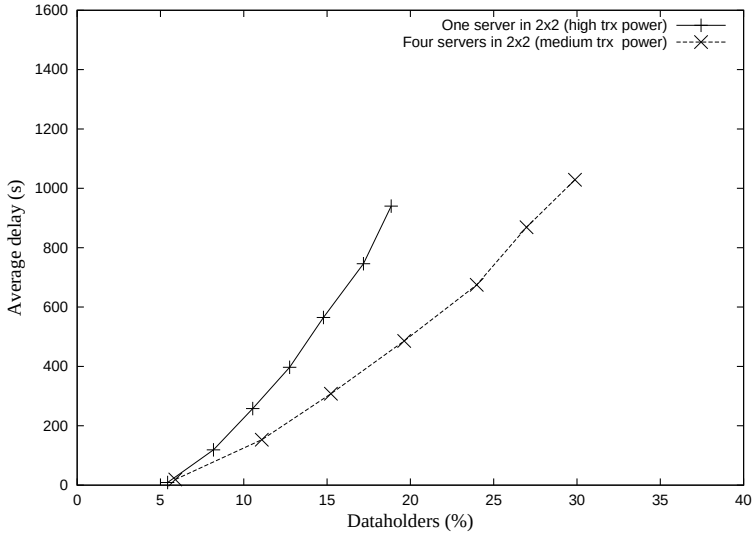

Figure 3.16: The average delay to receive the data as a function of the probability to acquire it in FIS. The "AxA" indicates the size of the area where the server and mobile clients are placed (in square kilometers).

this is true with both the FIS and the P-P schemes. An intuitive explanation is that, in Figure 3.20, the two deployments become equivalent by "scaling down" the left scheme (to match the right one). But after this "scaling", it is as if the speed of the hosts at the left scheme doubles. That is, the left setting is the same as the right one, in terms of area, transmission power of the servers, and server density, but with the hosts moving faster. Therefore, the probability a host will get into the coverage area of a server increases.

Figure 3.16 compares two FIS settings with the same total wireless coverage density of cooperative hosts (servers). The first includes one server in a 2 km x 2 km area with high transmission power and the latter four servers in a 2 km x 2 km area with medium transmission power. The case with higher density of servers performs better. For example, for a 20% probability of acquiring the data, the FIS scheme

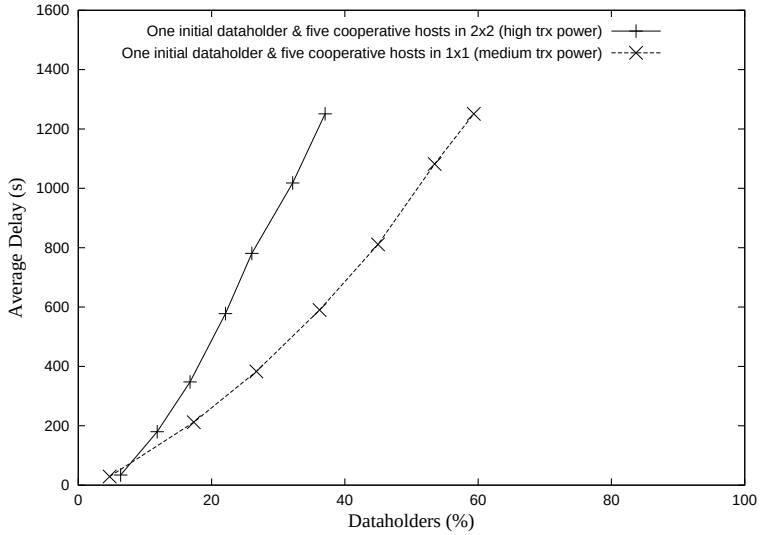

Figure 3.17: The average delay to receive the data as a function of the probability to acquire it in P-P with data sharing schemes. The "AxA" indicates the size of the area (in square kilometers).

with higher density of servers produces an average delay of 500 s. For the same wireless coverage, but lower density of servers, the average doubles. Figures 3.17 and 3.18 illustrate similar results in P-P schemes for different host densities. For a 40% probability of acquiring the data, the average delay is 600 sec in the higher density of hosts setting (5 hosts/km^2) whereas in a lower density setting, it doubles. Note that when we scale the speed of the mobile hosts and fix the mobility pattern and host density, we can compute the performance of data dissemination from the previous setting.

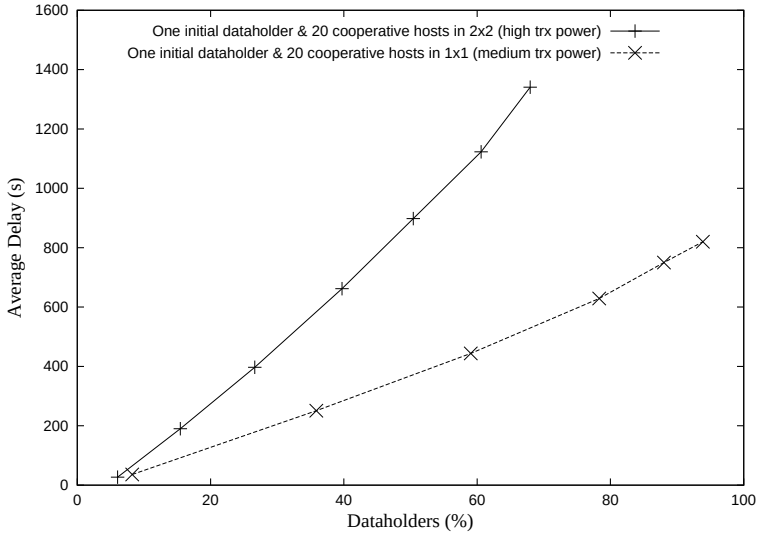

Figure 3.18: The average delay to receive the data as a function of the probability to acquire it in P-P with data sharing schemes. The "AxA" indicates the size of the area (in square kilometers).

3.4 Message relaying

As we discussed in the Chapter 1, message relaying is another facet of cooperation among mobile hosts. We assume hosts generate messages and buffer them locally when there is no Internet access. When a host gains access by reaching the wireless coverage area of a gateway, it relays these messages to the gateway. A host may relay its own messages to a peer when forwarding is enabled. We investigate the impact of message relaying on the probability that a message will reach a gateway and on the average delay from the time the message was created until it reaches a gateway.

To avoid a message explosion, we impose two restrictions. First, a host relays all queued messages to a gateway, but only its own messages to another peer. That is, a given message reaches the Internet via at most two hops. Secondly, we restrict

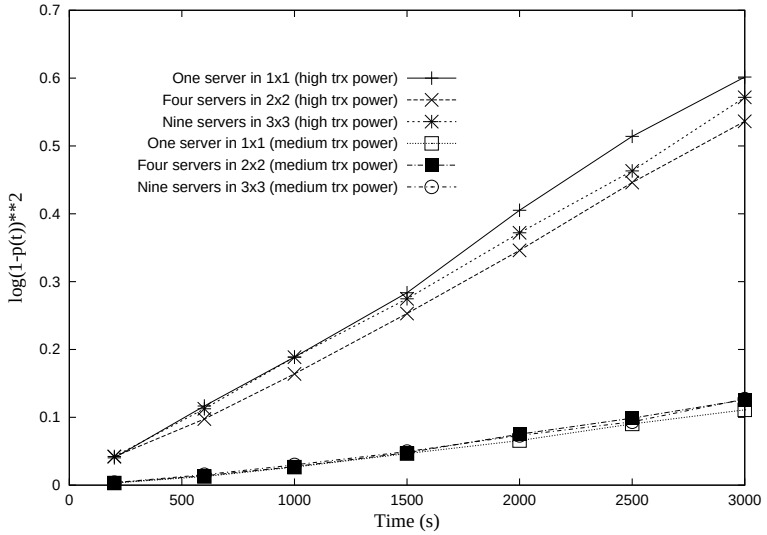

Figure 3.19: Performance of the fixed information server (FIS) schemes as a function of the simulation time.

the number of times a host may relay a given message. When a host has queued messages for relaying, it queries for a gateway or a relay host. A host that receives these queries may respond. Upon the receipt of such response, the querier forwards the queued messages to that host. Those messages need to satisfy the above two restrictions. In addition, a host transmits the same message only once to another host. The gateways periodically advertise their presence. Upon the receipt of such advertisement, a host forwards all the queued messages to the gateway.

Figures 3.21 and 3.22 illustrate the probability that a host will reach the Internet as a function of the host density. We assume one gateway per km^2 area. Figure 3.21 shows the percentage of the messages generated at each host during an interval (here is 25 minutes) that reach the Internet and the impact of forwarding to relay nodes. We assume that the hosts generate messages with a constant rate of one message every three minutes. The average number of buffered messages at each host

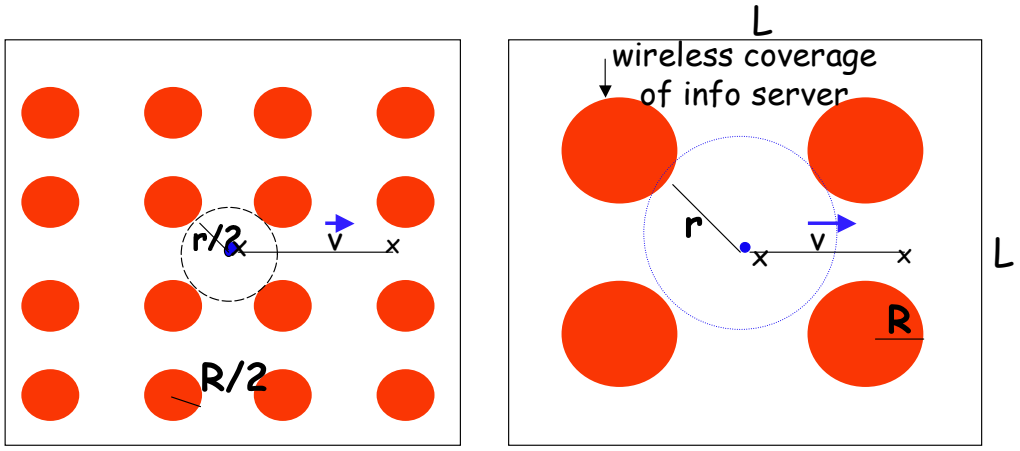

Figure 3.20: Investigating the scaling properties of data dissemination. The dark disks depict the wireless coverage of a host. For fixed wireless coverage, the larger the density of cooperative hosts, the more efficient the data dissemination.

is five. For high host density, forwarding doubles the percentage of messages that reach the Internet. Note in Figure 3.21 that forwarding a message to more than one relay node does not substantially improve the performance (FW6 vs. FW1 schemes). Figure 3.22 illustrates the probability that a message will reach the Internet within 25 minutes from the time the message was created on the source host. Note that when there is no forwarding, the probability that a message will reach a gateway is the same as the probability that the host will reach a gateway. Essentially, this probability is the same as the probability that a host will acquire the data in FIS for a gateway density equal to the server density in FIS. As in Figure 3.21, this probability increases when forwarding is enabled.

In a setting with a very low transmission power corresponding to a range of 8 meters (e.g., Bluetooth), and with high density of hosts (such as 100 hosts/km^2), after 2.5 hours, 5% of the messages generated during the first 25 minutes will reach a gateway directly and 38% of them via another relay host. This corresponds to a

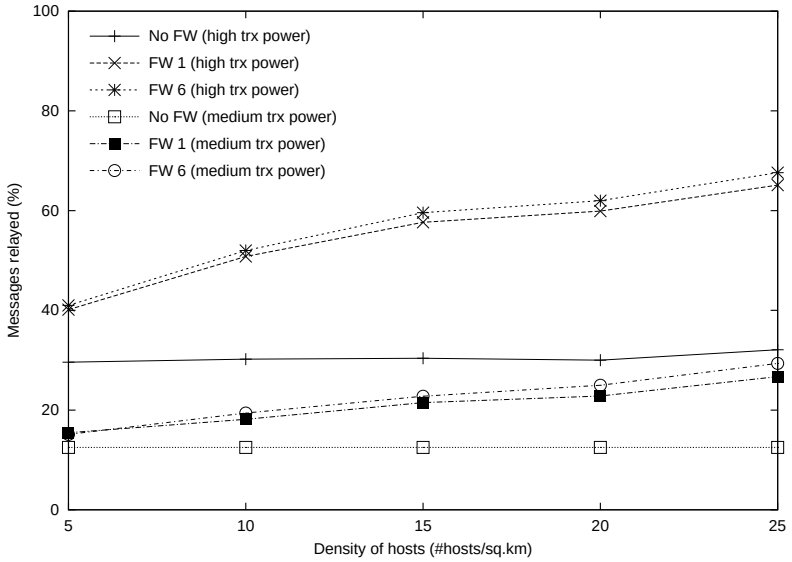

Figure 3.21: Percentage of the messages generated at each host that reach the Internet after 25 minutes in an area with one gateway per km^2. We use the notation "FWa" to indicate the maximum number of copies for each message, a, that a 7DS host relays to other nodes.

setting with forwarding enabled and forwarding number equal to 20, so that a given message can be relayed to at most 20 hosts. For a forwarding number equal to six, the percentage becomes 21%.

3.5 Summary

In this chapter, we discuss the performance of 7DS via simulations and the impact of the transmission power, host density, cooperation, query interval, and querying mechanism on the effectiveness of information dissemination and message relaying. Our results lead to the following conclusions:

P-P schemes outperform S-C schemes. The results indicate that the probability

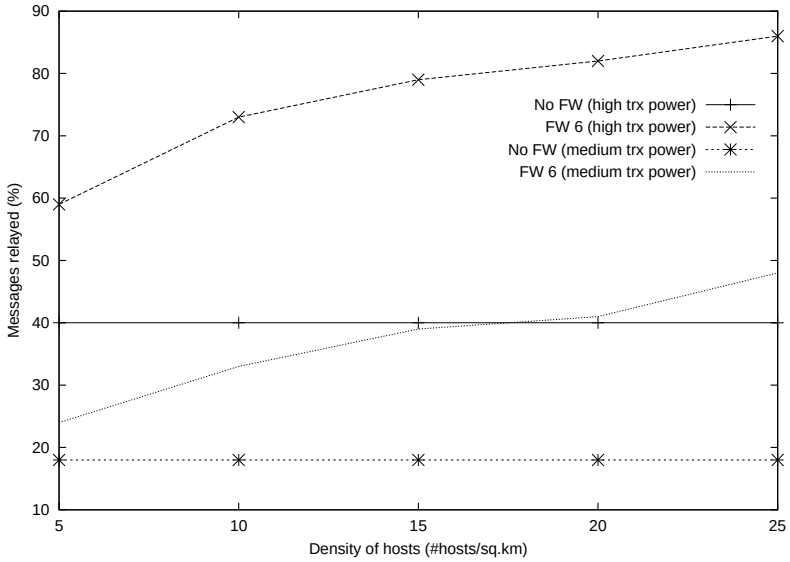

Figure 3.22: Probability that a message will reach the Internet within 25 minutes from the time it was generated on the source host. We assume an area with one gateway per km^2. We use the notation "FWa" to indicate the maximum number of copies for each message, a, that a 7DS host relays to other nodes.

that a host that queries for a data object will acquire it by time t using FIS and P-P, follows the $1 - e^{-a\sqrt{t}}$ and $1 - e^{-at}$, respectively. In case of high density of cooperative hosts, data dissemination using P-P grows even faster. Generally, the difference becomes more prominent in cases of medium or low transmission power, with more than ten hosts. In our setting with ten or more hosts per km^2, P-P provides 60% or higher probability for acquiring the data item to hosts that move in the area for 25 minutes and transmit with medium or high power. This probability is two to three times higher than in FIS. In some of the cases, difference in their average delays is negligible, in other cases FIS has lower average delay (with a maximum difference of 100 sec).

Forwarding (i.e., rebroadcasting 7DS messages upon their receipt) in addition

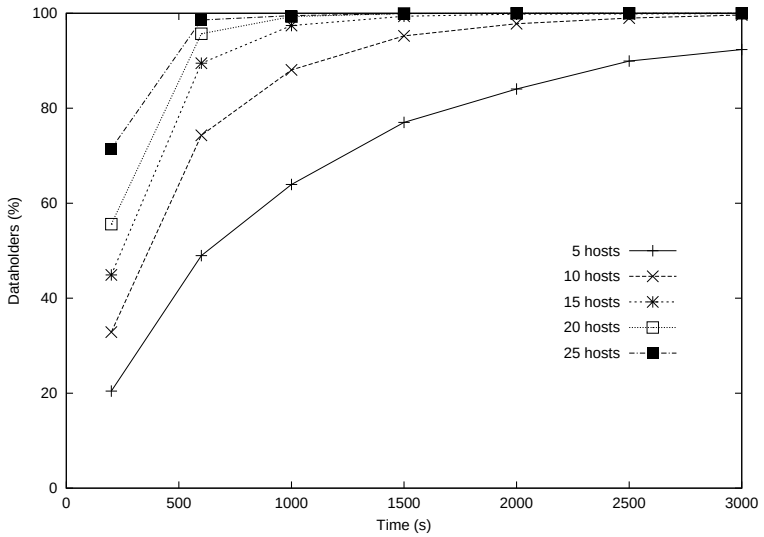

Figure 3.23: Performance of the peer-to-peer with data sharing and energy conservation enabled (DS) as a function of the simulation time for high transmission power. Each scheme has N cooperative hosts ($N = 5, \ldots, 25$) in a square kilometer area. Initially, one of them is dataholder.

to data sharing does not result in any performance improvements.

The query interval has negligible effect on S-C schemes.

The synchronous energy conservation method is beneficial. It increases the power savings without degrading the data dissemination.

Performance remains the same when we scale up the area, but keep the density of cooperative dataholders and their transmission range fixed.

Dominant parameters are the density of cooperative hosts and their wireless coverage density. Also, mobility can contribute to higher data dissemination. For a given wireless coverage density, the higher the density of cooperative hosts, the better the performance. For example, in both the P-P and FIS schemes, for the same wireless coverage, it is more efficient to have a larger number of servers with lower

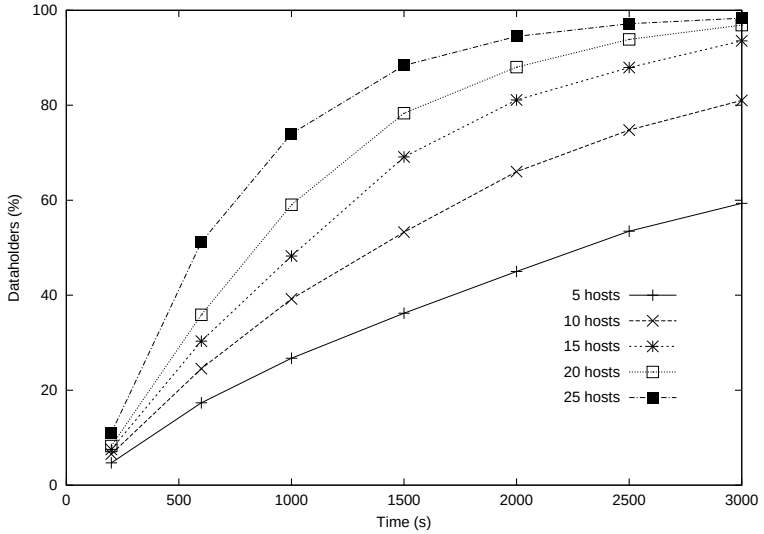

Figure 3.24: Performance of the peer-to-peer with data sharing and energy conservation enabled (DS) as a function of the simulation time for medium transmission power. Each scheme has N cooperative hosts ($N = 5, \ldots, 25$) in a square kilometer area. Initially, one of them is dataholder.

transmission power than fewer with higher transmission power.

Message relaying can increase the data access by exploiting host mobility. In the next chapter, we present our initial analytical results using diffusion-controlled processes theory.

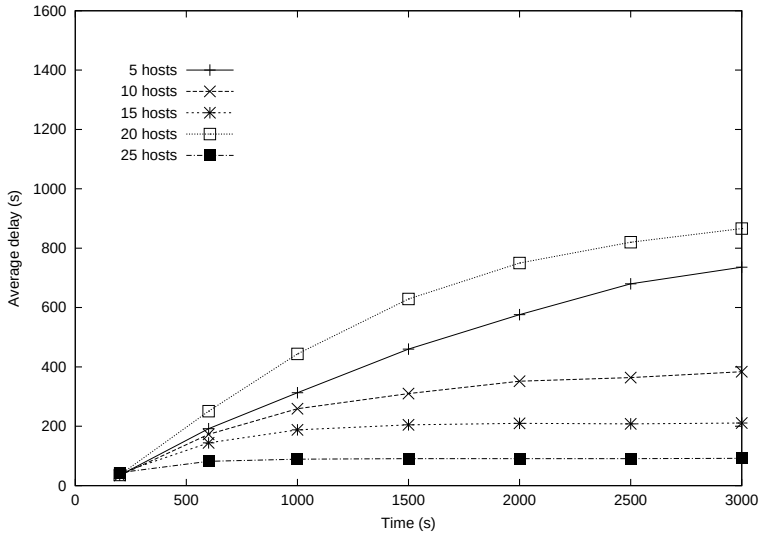

Figure 3.25: Average delay of the DS with energy conservation scheme as a function of simulation time (for various cooperative host densities and high transmission power). These hosts are in a square kilometer area.

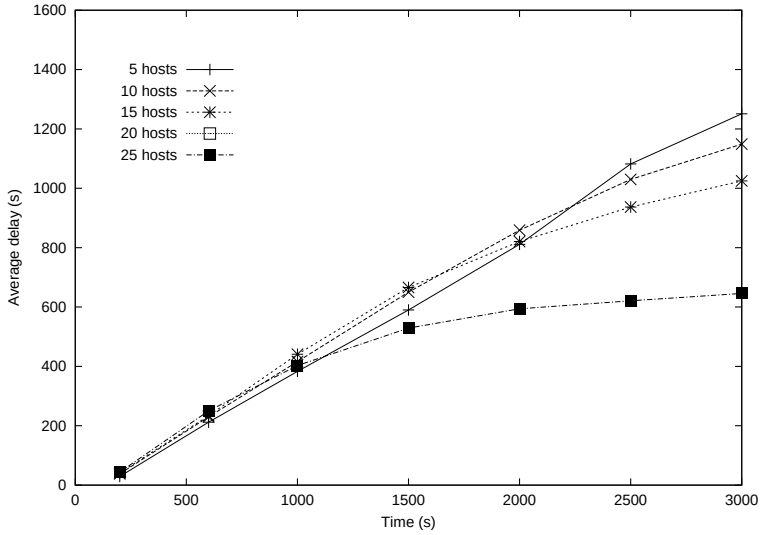

Figure 3.26: Average delay of the DS with energy conservation scheme as a function of simulation time (for various cooperative host densities and medium transmission power). These hosts are in a square kilometer area.

Chapter 4

Analysis of information dissemination

This section discusses our initial efforts to study data dissemination analytically and generalize further our results. Section 4.2 describes a simplistic epidemic model, and Section 4.3 contributes to a novel approach to model data dissemination borrowed from particle kinetics and diffusion-controlled processes.

4.1 Introduction and related work

Gossiping algorithms have been studied analytically. For example, [81] assumes a system where the nodes are placed on a line. They present an optimal algorithm for broadcasting, and compute the expected number of time steps required for it to complete. Other studies on information dissemination have used percolation theory [25] or epidemic models. In percolation theory, the nodes are typically placed on a lattice. When the shape theorem [25] holds for a particular setting (with respect to

the node layout, mobility or interaction pattern), it provides elegant techniques to estimate many properties, such as the expected time for a message to spread among all nodes. The shape theorem typically assumes a system in two-dimensional integers, in which each lattice site is either empty (0) or occupied (1), and in which the set A_t of occupied sites at time t grows and attains a limiting geometry. However, these models are not adequate for our setting, since they typically assume stationary nodes. To the best of our knowledge, there is no analytical work for the setting we described.

The epidemic model has also appeared in [73] and the diffusion-controlled process model in [74, 75].

4.2 Simple epidemic model for data propagation

As we mentioned, $7DS$ aims to prefetch and disseminate data for mobile hosts not necessarily connected to the Internet. Its effectiveness as a data dissemination and prefetching tool depends on a variety of parameters, such as 7DS node density in a certain region, node mobility, transmission power, cooperation strategy, querying mode, and energy conservation. It does not appear to be amenable to an analytical solution except for simplified versions. For example, if we assume that in any time interval h, any given dataholder will transmit data to a querier with probability $h\alpha + o(h)$, then the problem becomes much easier. More specifically, we can use a simple epidemic model described in [84] to compute the expected delay for the message to be propagated to the population of an area.

For the epidemic model, let us assume a population of N $7DS$ peers that at time 0 consists of one dataholder (the "infected" node) and $N-1$ queriers (the "susceptibles"). We assume that once a peer acquires the data, the data will stored

locally forever. Also, we suppose that in any time interval h any given dataholder will transmit data to a querier with probability $h\alpha + o(h)$. If we let $X(t)$ denote the number of data holders in the population at time t, the process $\{X(t),\ 0 \leq t\}$ is a pure birth process with

$$\lambda_k = \left\{ \begin{array}{ll} (N-k)N\alpha & k = 1, .., N-1 \\ 0 & \text{otherwise} \end{array} \right\} \tag{4.1}$$

That is, when there are k dataholders, each of the remaining mobiles will get the data at rate $k\alpha$. If T denotes the time until the data has spread among all the mobiles, then T can be represented as

$$T = \sum_{i=1}^{N-1} T_i, \tag{4.2}$$

where T_i is the time to go from i to $i+1$ dataholders. As the T_i are independent exponential random variables with respective rates $\lambda_i = (m-i)i\alpha$, $i = 1, .., m-1$, we see that

$$E[T] = \frac{1}{\alpha} \sum_{i=1}^{N-1} \frac{1}{i(N-1)}. \tag{4.3}$$

4.3 Data dissemination as a diffusion-controlled process

This section discusses our initial efforts to study the data dissemination analytically and to further generalize our results. It contributes a novel approach to model data dissemination. We also address the main theoretical results and challenges. The models are based on diffusion-controlled process that uses theory from random walks and environment [48], and the kinetics of diffusion-controlled chemical processes [67].

In particular, we use the diffusion in a medium with randomly distributed static traps to model the FIS scheme.

Let us first define the static trapping model. Particles of type C perform diffusive motion in d-dimension space. Particles of type S ("sinks" or traps) are static and randomly distributed in space. Particles C are absorbed on particles S when they step onto them. The basic trapping model assumes traps of infinite capacity. The diffusion controlled processes focus on the survival probability, that is the probability that a particle will not get trapped as a function of time.

For Rosenstock's trapping model in d dimensions (with a genuinely d-dimensional, unbiased walk of finite mean-square displacement per step), it has been shown that the large-n behavior of the survival probability

$$\log(\phi_n) \approx -\alpha[\log(\frac{1}{1-q})]^{2/(d+2)} n^{d/(d+2)} \tag{4.4}$$

In Eq. 4.4, α is a lattice-dependent constant, and q denotes the concentration of the independently distributed, irreversible traps.

One question is when Eq. 4.4 is a useful approximation. All previous analyses of this question have relied on some form of simulation, but so far there was no information available on the range of validity of Eq. 4.4. In the Letter [46], Havlin *et al* present evidence suggesting that Eq. 4.4 is a useful approximation when

$$\rho > 10 \tag{4.5}$$

where ρ is the scaling function

$$\rho = (ln\frac{1}{1-q})^{\frac{2}{d+2}} n^{\frac{d}{d+2}}. \tag{4.6}$$

This value of ρ corresponds to a survival probability equal to 10^{-13} in both $d = 2$ and $d = 3$ dimensions. They argue that pure simulation techniques will always lead to an

exponential decay at sufficiently long times, rather than to the correct decay given by the theoretically-proven Eq. 4.4. Their evidence for the new lower value of ρ (or higher value of $P(n)$) is based on two numerical techniques that they have developed. One of them is practical for high trap concentrations only ($0.9 \leq q$). This case of high trap concentrations is similar to our case.

As we mentioned in Section 3.2, in FIS information sharing takes place among the server and the querier. When a $7DS$ querier comes in close proximity to the server, it acquires the data. It is easy to draw the analogy: the traps are the stationary information servers, the particles C are the queriers, and the trapping is essentially receiving the data. We model the stationary information servers as traps and the mobile peers as particles C. When a host acquires the data, it stops participating in the system, and is considered "trapped" with respect to the model. Figure 4.1

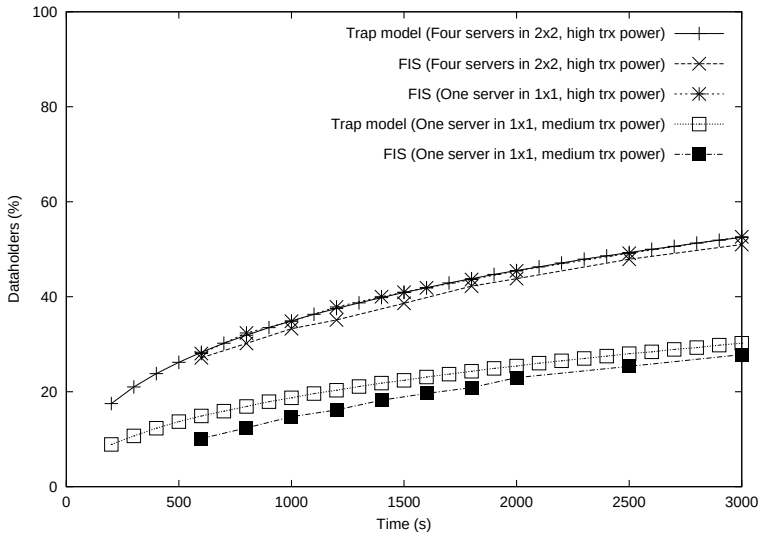

Figure 4.1: Simulation (FIS) and analytical trapping model (Trap) results. The "AxA" indicates the size of the area in which the servers are placed (in square kilometers).

illustrates the analytical and simulation results for data dissemination as a function of time. The analytical results for trapping model are derived from Eq. 4.4 (Rosenstock's trapping model) for high and medium transmission power.

We define q as $\pi\ R^2 N_{servers}/A^2$, where $N_{servers}$ is the number of servers placed in an area of size AxA and R is the wireless coverage equal to 230 m and 115 m for high and medium wireless coverage, respectively. For the simulation results on FIS in Figure 4.1, we use the FIS simulations we described in Section 3.3. Note that, using Eq. 4.4, the term $1-\phi_n$ expresses the fraction of hosts that acquire the data at time n. As Figure 4.1 illustrates, our simulations are consistent with Eq. 4.4 (in two dimensions) for α equal to 0.021. That is, using Eq. 4.4, the $(1-\phi_n)*100\%$ match our simulation results for the percentage of dataholders at time n for the FIS scheme we described. In a setting of one stationary host per square kilometer with high transmission power that corresponds to a range of 230 m, the server concentration is $c = 0.1661$ and the criteria $\rho > 10$ is equivalent to $n > 550$.

An attractive feature of the diffusion-controlled processes in the context of our research is that it can provide elegant tools and methodology to investigate data dissemination for different server distributions. Also, we are currently exploring how we can use it to model other types of interaction (S-C and P-P schemes) and incorporate parameters such as the expiration of data objects.

Chapter 5

Network connection sharing in wireless LANs

In this chapter, we focus on the third facet of cooperation, namely bandwidth sharing in a wireless LAN. We motivate network connection sharing and present the main components of the architecture and its performance evaluation via simulations.

5.1 Introduction

Current wide-area network wireless deployment is characterized by intermittent connectivity, low bit rates, and high delays. These constraints provide a strong incentive to better utilize the user's local resources, in order to achieve better quality of service (QoS), and higher data availability. The characteristics of collaborative applications led us to the natural extension of data sharing to network connection sharing. The central idea is that 7DS peers share their network connections in order to improve their service, increase the data availability, and potentially introduce other benefits,

as described in the following paragraphs.

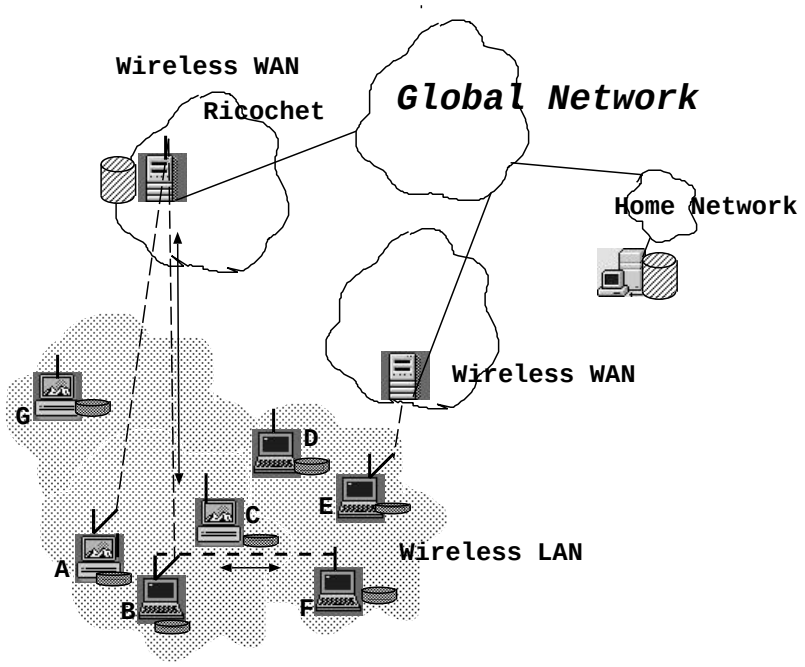

Figure 5.1: Description of the environment: Hosts share a wireless LAN. Some of them have a wireless WAN connection to access the Internet.

Figure 5.1 illustrates an environment for network connection sharing: there is a group of 7DS hosts (A, B, C, D, E, F, and G) in close proximity. As we assumed in Chapter 1.2, 7DS communicates with peers via a wireless LAN (e.g., IEEE 802.11b). Some of them (A, B, and E) have an additional network interface that provides them with access to the Internet via a wireless WAN connection. It would be typical in the near future to support a wireless WAN connection of 100 kb/s [27, 39, 63, 59]. Currently, there are wireless WAN modems of (approximately) 45 kb/s. A shared 2-11 Mb/s wireless LAN is typical (e.g., [99]).

We envision this system to be especially applicable in cases where users meet in a conference or meeting (e.g., at an IETF meeting), or in a train, and want to gain

Internet access. They decide to cooperate and share or lend their resources in order to facilitate a common need and potentially create other gains. The motivations for connection sharing are as follows:

- utilization of temporarily idle WAN connections,
- exploitation of statistical multiplexing for bursty traffic, and
- reduction in the transmission of replicated data that belong to "shared" (collaborative) applications.

In this chapter, we focus on the first two motivations. We discuss the third one briefly in Section 5.5.

When a host is connected to the network, there are periods when the connection is idle, such as when the user is reading a page that was downloaded earlier. Very often, when a user does not use her connection temporarily, she does not disconnect from the network. While her connection is idle, another peer in the ad hoc network may use this mobile device as a *gateway* to the global network. Consider another example, in which the group members videoconference with some other colleagues over the Internet (as in Figure 5.1), or view the news from a server. It is unnecessary to transmit the data as multiple streams with the same content. Instead, it is sufficient for one of the hosts with access to the Internet to receive the stream via its wireless WAN connection and multicast it to the rest of the group via the wireless LAN. Alternatively, in a home network or a wireless network in a vehicle, where a few devices have Internet access and the rest can use them to access the Internet. This host temporarily acts as a *gateway*. Throughout the chapter, the term gateway denotes any host that acts *temporarily* as a gateway for other hosts in the group. It must

have access to the Internet and a wireless LAN interface. The other hosts either do not have a connection to the Internet, or have a connection which is saturated at that instant. Also, we use the term gateway connection to refer to the wireless WAN connection of the gateway.

As a result, hosts with a wireless WAN connection in this environment temporarily act as gateways, unlike traditional networks where the routers are fixed in place. Another difference between this environment and that of traditional networks is the lack of mechanisms for directing flows to different routers based on criteria such as bandwidth availability. Network connection sharing does not change if instead of wireless WAN connections, there are wired WAN connections, such as ADSL (or cable modem) lines.

Under the connection sharing mechanism, the gains for the hosts with no wireless WAN connection are obvious. However, even the hosts with a wireless WAN connection can potentially benefit. As we describe in greater detail in Section 5.5, when users are receiving the same data, such as participants in a multicast discussion, connection sharing results in better QoS. The bandwidth requirement for the transmission of all layers of a multimedia object[1] is usually much higher than the capacity of a single wireless WAN connection. However, if hosts collaborate and use the aggregate bandwidth of their connections for the layered multimedia transmission, the video quality can be increased dramatically.

As we discussed in Chapter 2.2, the willingness of systems to cooperate is crucial in peer-to-peer systems. There are settings, such as in a corporation or a

[1]The design of tools for video conferencing services, conference controllers, and QoS control mechanisms is the focus of papers such as [90, 23]. L. Wu *et al* [102] and S. Floyd *et al* [35] investigate layered video transmission.

conference among colleagues, or in rescue operations, where hosts are naturally motivated to cooperate and share their bandwidth, since they belong to users or an infrastructure with common goals. In some other settings, though, users have fewer incentives to cooperate, especially when the cooperation drains their battery energy. In other cases, the owner of the connection may receive financial benefits through a renting or rewarding mechanism. A host may lend a part of its connection, depending on the bandwidth availability and the bandwidth requirements of the flow that need to be served. Pricing issues may therefore have an important effect on the system operation. A variety of different pricing arrangements[2] depending on the setup and the users' relation (degrees of collaboration) are possible and make connection sharing desirable, despite the cost and energy consumption requirements of keeping them active. The relatively high power consumption when transmitting data may constrain the deployment of connection sharing. On the other hand, the power consumption for wireless modems is decreasing, and the number of electrical outlets is increasing in places where we expect the system to be used (such as conference rooms, trains, and airports). In addition, note that wireless WAN access is generally more expensive (e.g., higher subscription fees) than local-area access (such as infrared, Bluetooth, WaveLAN). Thus, if a mobile user accesses the network infrequently, "leasing" a temporary gateway is more efficient.

In this chapter, we concentrate on the basic components of the architecture and study its performance. Our main contributions are the design of a novel system that provides dynamic resource sharing among collaborating hosts, and its performance evaluation. The four main components of the system are admission control

[2]An example of such pricing arrangement would be a "bandwidth co-op" scheme.

at the gateways, a mechanism that assists hosts with selecting a gateway while ensuring traffic balancing across the gateways, a traffic measurement mechanism at a gateway, and a mechanism to announce the gateway availability. We measure the performance of the system by simulation. Specifically, we consider a time snapshot in which a fixed number of gateways provide their wireless WAN connection to serve hosts in the wireless LAN that request access. The requests correspond to various services and generate control and data traffic in the wireless LAN and at the gateway connections. We measure the bandwidth utilization (and the gains from statistical multiplexing), and the packet dropping rates at each gateway connection. We found that the bandwidth utilization varies from 21% to 82%, and the dropping rates from 0% to 10%, depending on the traffic model characteristics.

The traffic overhead due to the control messages exchanged in order to enable the sharing is very low. It contributes around 0.9 kb/s to 1.8 kb/s to the wireless LAN (compared to the wireless LAN bandwidth capacity that ranges from 1.2 Mb/s to 11 Mb/s, depending on the technology). Section 5.3.4 evaluates the protocol overhead. Finally, the selection mechanism that the hosts use to choose a gateway achieves load balancing across the gateways. The load balancing metric, as defined in Eq. 5.1, ranges from 1.7% to 4.6%.

This network connection sharing protocol has also appeared in [72].

The remainder of this chapter is organized as follows. In Section 5.2, we discuss briefly related work. Section 5.3 gives an overview of the connection sharing system. Section 5.3.1 describes the measurement of the gateway traffic and the announcement policy. In Section 5.3.2, we present the gateway selection mechanism that ensures load balancing across the gateways. Section 5.3.3 discusses the admission control policy at

the gateways and Section 5.3.4 evaluates the protocol overhead. Section 5.4 presents simulation results. Finally, in Section 5.5, we summarize our conclusions and discuss directions for future work.

5.2 Related work

There has been work on the deployment of a combination of wireless networks of different technologies. For example, Stemm and Katz [93] considered a hierarchy of network interfaces that included combinations of wireless network interfaces, spanning in-room, in-building, campus, metropolitan, and regional cell sizes. Their main objective was to enable a user to roam among multiple wireless networks in a manner that was transparent to applications and would reduce the handoff disruption. They focused on performance issues for vertical handoffs, i.e., handoffs between base stations that were using different wireless network technologies.

The MosquitoNet project [105] addressed the multiple connectivity management on mobile hosts, i.e., the need to support multiple packet delivery methods simultaneously, and the use of multiple network devices for both availability and efficiency reasons. Multiple interfaces were not available at any point in time; just the "best" interface that is selected according to a specific policy. Goals similar to those of Stanford's MosquitoNet, InfoPad, and Daidalus project (e.g., [6]) were also discussed in [51]. While these groups focused more on mobile IP implementations, Inouye *et al* [51] dealt more with dynamic reconfiguration policies.

A large amount of work focuses on routing protocols to support mobility, and some on ad hoc mobile networks [11, 85, 103]. Broch *et al* [10] focus on routing protocols in a setting similar to ours. In their setting, hosts also have multiple network

interfaces. They describe a technique that allows a single ad hoc network to span across heterogeneous link layers. It enables the use of heterogeneous interfaces, the integration of an ad hoc network into the Internet as a subnet and the movement of mobile nodes into and out of an ad hoc network using Mobile IP. However, to the best of our knowledge, there is no paper in the wireless environment we describe that allows collaborating hosts to share their wireless WAN connections to increase the data availability and QoS while guaranteeing a load balancing across the gateways. Under this network connection sharing framework, we plan to exploit further the nature of collaborative applications that support scalable multimedia streaming data, such as layered video.

5.3 Overview of connection sharing

Before proceeding with the overview of the connection sharing system, let us state our assumptions for the setup:

- In general, hosts and base stations can operate in the system as gateways. In this setup, we assume that all of the gateways are hosts with wireless WAN connections of the same bandwidth. The system is not restricted to support only hosts with the same bandwidth capabilities, but we assume this for simplicity of exposition.

- All hosts that participate in the system have a wireless LAN interface. There is an well-known multicast address designated for the network connection sharing protocol. Hosts can listen to that multicast group.

- A host may leave the multicast group or stop acting as a gateway without prior

notification.

- The system treats all the gateways uniformly. As we describe in Section 5.3.2, we aim at a selection policy which guarantees load balancing across the gateways.

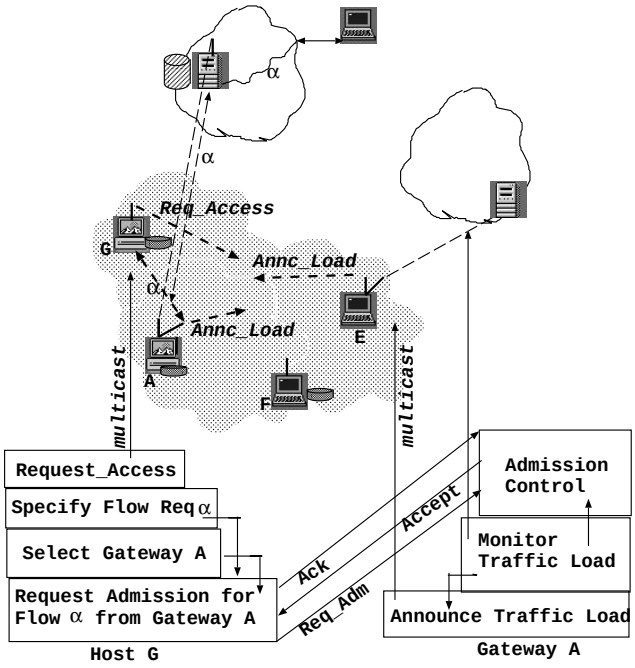

Figure 5.2: Overview of the communication protocol that enables network connection sharing.

As we mentioned in Section 5.1 and illustrated in Figure 5.2, hosts create a wireless LAN dynamically and communicate in order to collaborate and share their network connections. They communicate by sending messages and listening to a well-known multicast address. They multicast requests for accessing the Internet. The gateways multicast announcements of their measured traffic load and availability. In order for a gateway to share its wireless WAN connection with a host, the gateway needs to

decide if there are sufficient resources for a given flow. This decision is made through admission control. Admission control messages are unicast via the wireless LAN. We emphasize that the need for admission control depends on the resource sharing and cooperation characteristics. For network connection sharing without any guarantees, the host may operate temporarily as a gateway in a best-effort fashion. In this case, no admission control is required. However, if there is a pricing mechanism that charges the user who rents the resource, then some form of admission control is needed.

Figure 5.2 gives an overview of the communication protocol that takes place among the group members and enables connection sharing. For example, host G requests access over the Internet for flow α with peak rate r_α. It queries for an available gateway by sending a Request Access multicast message. As we describe in Section 5.3.1, the gateways (here, hosts A and E) announce the measured traffic on their wireless WAN link. G waits for time T_c to collect the gateway announcements and selects a gateway. Let us assume that it selects gateway A. Then, G sends a Request Admission message directly to A to share A's wireless WAN connection. In this unicast message, G includes the peak rate of the flow, r_α. Upon receiving a Request Admission, the gateway decides to accept or refuse to serve the flow. Host A sends the decision to host G and G sends back an acknowledgement.

We discuss the main components of the system in more detail:

1. Traffic load estimation of a gateway, i.e., the bandwidth utilization of the wireless WAN connection over a sampling period (in Section 5.3.1).

2. The gateway policy for announcing traffic load (in Section 5.3.1).

3. The criteria the hosts use to select a gateway (in Section 5.3.2).

4. The admission control mechanism at the gateway (in Section 5.3.3).

There are some additional architectural issues closely related to the security mechanism and the pricing arrangement for realizing the network connection sharing or leasing. In this thesis, we concentrate on the basic components of the architecture. Security and pricing issues are topics of future work.

5.3.1 Measurement and announcement of gateway traffic

Each gateway is capable of estimating the load of the wireless WAN periodically. The gateway computes an average load every sampling period S (typically a few hundreds of ms). The most recent estimated average load is the value of the traffic load that a gateway announces.

There are two possible announcement policies for the gateway traffic load:

- **Gateway advertisement policy (A)**: The gateway periodically multicasts its traffic load to the group every T_a sec.
- **Request-initiated policy (R)**: The gateway multicasts its estimated traffic load that corresponds to the last sampling *only* in response to a Request Access message.

The purpose of announcing the traffic load is to let the hosts know about the available gateways and select the appropriate gateway to share its connection to the Internet. As we discuss further in Section 5.3.2, the selection assists in load balancing the traffic across the gateways. We need to emphasize that the selection of the gateway is *not* an admission control mechanism. It indicates which gateway the querier should contact for admission control. In the future, we plan to include pricing information in these

announcement messages as part of a pricing mechanism. This would enable a leasing or bandwidth co-op scheme for the network connection sharing.

5.3.2 Gateway selection mechanism

As we mentioned, a host may request access to the Internet for a specific service. For that, it selects a gateway by listening to the multicast announcements of the gateways to obtain their estimated traffic load. The selection must be made while ensuring load balancing requirements across the gateways. In this work, the load balancing criteria is the reduction of the maximum difference in the average load over a time period (snapshot), τ, across the gateways. Specifically, the load balancing metric we consider is

$$\sigma \;=\; \frac{\max_i \{L_i(\tau)\} - \min_i \{L_i(\tau)\}}{b} \tag{5.1}$$

where $L_i(\tau)$ is the average traffic measured at the gateway i over the time interval τ, and b is the bandwidth capacity of the gateway connection which is the bandwidth of the wireless WAN link.

We assume no knowledge of the arrivals of future request or their duration. The problem of minimizing σ is a hard due to its on-line nature and the burstiness of traffic. We suggest a *greedy* algorithm and show through simulation results that we can achieve a fairly balanced system for different types of traffic. A host that requests access to the Internet chooses the *least-loaded gateway*, based on the traffic load value included in the most recent announcement from the gateways. Its low operational cost, simplicity, and good performance make the greedy algorithm an attractive choice for the system. We investigate its performance through simulations

for a variety of traffic models such as the the exponential and Pareto distributions. For both exponential and Pareto distributions, the greedy algorithm performs well: σ ranges from 1.7% to 4.6% (as defined in Eq. 5.1). Section 5.4 presents the traffic models and the results in detail.

Lastly, we note that in general, Eq. 5.1 is not a representative metric for load balancing, since it does not capture the potential skew of the load across the gateways. It has been used mostly to express a fairness criteria. However, in cases where its value is very small, as it is in our simulations, it also ensures that the system is load balanced.

5.3.3 Admission control

The gateway may provide some service guarantees to ensure that sufficient resources are available to serve the flows. For that, the system applies admission control. The criteria to choose an admission control mechanism are:

- low complexity, easy implementation, and low operational cost,
- high bandwidth utilization,
- designed for adaptive, real-time applications that can tolerate variance in packet delays and some packet loss.

Notice that due to the dynamic nature of the system where gateways may walk away, an admission control with strict quality-of-service (QoS) guarantees does not match with the characteristics of this system. The admission control algorithm we choose for the system is the Measured Sum algorithm by Jamin *et al* [54]. The Measured Sum algorithm has low operational cost, promises high bandwidth utilization, and does not

make strict guarantees. In that paper, Jamin *et al* discuss several measurement-based admission control algorithms [3].

In the Measured Sum algorithm, each gateway uses measurements to estimate the load of existing traffic. A gateway admits the new flow requested by a host if the current load plus the peak rate of the new flow is less than or equal to the gateway's desired bandwidth utilization. That is,

$$\hat{v} + r_\alpha \leq u\beta \tag{5.2}$$

where u is a user-defined utilization target, β is the bandwidth capacity of the gateway, $\hat{v}$ the measured load of existing traffic, and r_α the peak rate requested by flow α.

As mentioned in Section 5.3, each gateway is capable of periodically estimating the load of the connection, a point-to-point link, that uses to access the Internet. Specifically, it computes an average load every sampling period S. At the end of a measurement window T_m, the gateway uses the highest average from the just ended T_m as the load estimate for the next T_m window. When a new flow is admitted to the network, the estimate is increased by the parameters of the new request to reflect the worst-cast expectations, and then the measurement window is restarted. If a newly-computed average is above the estimate, the estimate is immediately raised to the new average. At the end of every T_m, the estimate is adjusted to the actual load measured in the previous T_m. As expected, a smaller S gives higher maximal averages, resulting in a more conservative admission control algorithm. A larger T_m keeps a longer measurement history, again resulting in a more conservative admission control algorithm, as we illustrate through simulations in Section 5.4, Table 5.4. If a flow is admitted, it is served by that gateway until its completion or premature

[3]The simulation study on network connection sharing has benefited from [54].

termination when the gateway or host leaves. If a flow is rejected, the querier merely drops it, as opposed to queuing it and retrying later. That is, the system performs in a "drop call lost" fashion rather than "drop call retry".

In Section 5.4, we run simulations to investigate the performance of "Measured Sum" in this system.

5.3.4 Connection sharing protocol overhead

The overhead of the protocol is caused by the control messages that are exchanged to coordinate resource sharing. It includes traffic announcements (for the A or R policies), the request for access (Request Access), and the admission control (Request Admission, Accept/Reject, and Ack) messages. Note that these messages contribute *only* to the traffic on the wireless LAN.

Let B^P_{proto}, where $P \in \{A, R\}$, be the average overhead in bandwidth, n_g be the average number of gateways that participate in the system (simultaneously), pkt be the packet size, b be the bandwidth of the gateway connection, and f be the aggregate (i.e., generated from *all* participants) flow interarrival time.

$$B^P_{proto} = B_{reqacc} + B_{adm} + B^P_{annc} \tag{5.3}$$

where $B_{reqacc} = \frac{pkt}{f}$ and $B_{adm} = \frac{3pkt}{f}$.

The overhead of the announcement policy is:

$$B^P_{annc} = \begin{cases} \frac{n_g pkt}{f} & \text{if P=R} \\ \frac{n_g pkt}{T_a} & \text{if P=A} \end{cases}$$

As expected, the difference in the overhead depends on the interval values and the aggregate flow interarrival time. Note also that the flow across the gateways will not

saturate the wireless LAN bandwidth network as long as $n_g \leq \frac{B-B_{proto}}{b}$. From this, we can compute the maximum number of gateways in the group , n_g^{max},

$$n_g \leq \frac{B - \frac{4pkt}{f}}{b + \frac{pkt}{f}} \Rightarrow n_g^{max} = \lfloor \frac{B - \frac{4pkt}{f}}{b + \frac{pkt}{f}} \rfloor \ if\ P = R \tag{5.4}$$

$$n_g \leq \frac{B - \frac{4pkt}{T_a}}{b + \frac{pkt}{T_a}} \Rightarrow n_g^{max} = \lfloor \frac{B - \frac{4pkt}{T_a}}{b + \frac{pkt}{T_a}} \rfloor \ if\ P = A \tag{5.5}$$

From Eqs. 5.5 and 5.4, given a typical range of values of B, b, f and T_a, we see that $\frac{B}{b}$ is the dominant term in determining the value of n_g^{max}.

5.4 Performance evaluation

We consider a time snapshot in which a fixed number of gateways operate. Hosts request access to the Internet from the gateways. The requests correspond to various services and generate data traffic, i.e., flows, in the wireless LAN and at the gateway connections. We use the ns-2 simulator [33] to quantify the performance of the system. The performance measurements include the bandwidth utilization and the packet dropping rates at each gateway connection, the protocol overhead, and the load balancing characteristics across the gateways. The simulation is parametrized on the source flow traffic model, the bandwidth and total link delay of the wireless WAN connection, the bandwidth and total link delay of the wireless LAN, the simulation time, the measurement time, the number of gateways, the aggregate (i.e., generated from *all* the participants) flow interarrival time, the aimed bandwidth utilization of the gateway connection, the interval size, and the sampling period for measuring the traffic at the gateway connection.

5.4.1 Traffic models

First we describe the simulation parameters and the motivations for their values. The hosts generate *homogeneous* data traffic, each CBR, Pareto, or exponential, and the same flow interarrival time. Our main focus is on Pareto and exponential data traffic, since they more accurately approximate the actual measured traffic. We also run a few tests for CBR data traffic. Willinger *et al* [100] modeled measured Ethernet LAN traffic[4] with well-known on/off source models, such as Pareto. Paxson *et al* [76] showed that network traffic often exhibits long-range dependence (LRD), with the implications that congested periods can be quite long, and a slight increase in the number of active connections can result in a large increase in the packet loss rate. Each Pareto traffic flow itself does not generate LRD. However, the aggregation of Pareto traffic results in LRD.

- Exponential: on/off model with exponentially distributed on and off times. During each on period, an exponentially distributed random number of packets are generated at a fixed rate, with an average off time and an average on time.

- Pareto distribution: during each ON period of the Pareto flow, packets are generated at peak rate, an average burst, and an average idle time. According to [100], the shape parameter of the Pareto distributed off and on times covers the interval $(1, 2)$. The shape-parameter-estimate of the OFF period stays mostly below 1.5. In our simulations, the shape parameter for both the ON and OFF periods is 1.2.

[4]This data set includes traffic due to applications, such as ftp, e-mail, WWW, and Mbone [100].

5.4.2 Wireless access models

Emerging third generation networks (3G) [39] aim at supporting user bit rates of up to 144 kb/s with wide mobility and coverage and up to 2 Mb/s with local mobility and coverage. We simulate the wireless WAN connection as a link of bandwidth of 100 kb/s, and total link delay in the range of 100 msec or 165 msec [17, 59]. The total link delay is the sum of MAC delay, link layer delay, and propagation delay.

The wireless LAN in our testbed is RadioLAN [79] or WaveLAN [99]. We ran some actual tests to estimate their bandwidth capabilities. The tests involve two laptops located indoors, each with a PCMCIA card, placed at a distance varying from 2-30 meters. These two hosts are the only participants of the wireless LAN. The measurements include ftp transfer and bandwidth estimation of a link using *pathchar* [53] and *hop_speed* [55]. The highest value of the RadioLAN link between the two laptops capacity measured was 5.8 Mb/s, running the ftp transfer test. We repeated ten ftp transfers of a large, MPEG-1 file of 33.5 MB from one host to another. The 5.8 Mb/s corresponds to the average bit rate of these tests. The *hop_speed* estimates the bandwidth to be 4.8 Mb/s. We repeated the tests for the WaveLAN (note that this is pre-IEEE802.11). Using *hop_speed*, we found the bandwidth of the link to be 1.2 Mb/s. In our simulations, the wireless LAN has a bandwidth of 2 Mb/s, link delay of 64 μs, and a CSMA/CA MAC layer.

We assume no failures or disconnection occur during the snapshot of the test. The group size and the number of gateways n_g remain fixed during that period, i.e., there are no changes due to gateway arrivals or departures. We experiment with n_g values of three and ten. In all of the simulations, the announcement policy used is R. Also, in all of the simulations, the aggregate flow interarrival time follows an

exponential distribution. The aimed bandwidth utilization of the gateway connection is 95%. Throughout these tests, the packet size is fixed at 100 bytes and the buffer size at the gateway connection is fixed at 160 packets. The bandwidth and total link delay of the wireless WAN connection is 100 kb/s and 165 msec, respectively.

The measurement time indicates when we start measuring the link utilization and the dropping rates. As recommended in [54], Pareto on/off sources require a longer warmup period and a longer simulation time for the LRD effect to be seen. Thus we run them, if not otherwise specified, for a total simulation time of 18,000 sec and measurement time of 10,000 sec. The exponential sources run for a total simulation time of 3,000 sec and measurement time of 1,500 sec.

5.4.3 On constant bit rate (CBR) traffic

The CBR source has a rate of 64 kb/s. The snapshot of the test is [1,500 sec, 3,000 sec] (warmup period of 1,500 sec). The aggregate flow interarrival mean is 600 msec. The holding time of the flows follow the Pareto distribution with a mean of 300 sec and shape parameter of 2.5. Table 5.1 presents the measurements on the bandwidth utilization of the wireless WAN. In both the cases the packet dropping rate in the gateway link is 0%.

Number of gateways	**Utilization (%) at each gateway**
10	$63.9, 63.4, 63.3, 63, 63.2, 63.7, 63.2, 63.4, 63, 63.6$
3	$64, 63.4, 63.4$

Table 5.1: Bandwidth utilization at each gateway in the case of CBR traffic. The bitrate is 64 kb/s, the flow interarrival time is 0.6 sec, and the mean holding time is 300 sec.

We obtain the confidence interval [62] for the average bandwidth utilization of each gateway, the packet dropping rates and the load balancing metric σ (as defined in Eq. 5.1), for a system with Pareto and exponential traffic. The wireless LAN consists of six hosts, three of them acting as gateways. Each gateway connection has a bandwidth capacity of 100 kb/s, and the total link delay is 165 ms. The measuring interval time is 3 sec, and the sampling period is 400 msec. We repeat the simulations 64 times in the Pareto case and 100 times in the exponential case, each time with a different random number seed. The two cases differ only in the data traffic model and the flow holding time. In the exponential case, the generated flows follow an exponential distribution with peak rate of 64 kb/s, average on time of 312 msec, and idle time of 325 msec. The holding time follows an exponential distribution, with a mean equal to 300 sec. In the Pareto case, the traffic follows a Pareto distribution with peak rate of 64 kb/s, a shape parameter of 1.2, mean bursty time equal to 312 msec, and mean idle time of 325 msec. The holding time for Pareto traffic follows a Pareto distribution with an average of 300 sec and shape parameter equal to 2.5 [24, 82].

Before proceeding with the exposition of our results, let us first show how we measure the load balancing metric σ (as defined in Eq. 5.1) in the simulations: At the end of each test, we compute the average utilization of each gateway connection over a time interval $[MeasT,\ SimT]$, $L_i([MeasT,\ SimT])$, for $i = 1, 2, 3$. From that, we find the maximum difference in the traffic across the three gateways and compute σ according to Eq. 5.1. We repeat the tests 64 times for the Pareto case and 100 times for the exponential case, each time with a different seed. From these values, we compute the confidence interval for the load balancing metric.

5.4.4 On Pareto traffic

Table 5.2 illustrates the packet dropping rate and link bandwidth utilization for each gateway. We run simulations for aggregate flow interarrival mean of 600 msec or 6 sec. The 99% confidence interval for the load balancing of the system, σ (as defined in Eq. 5.1), is $[1.63\%, 2.52\%]$ when the aggregate flow interarrival mean is 600 msec and $[3.4\%, 4.6\%]$ when the aggregate flow interarrival mean is 6 sec.

The packet dropping rate is very high, for example, around 10% when the aggregate flow interarrival mean is 0.6 sec. As previously mentioned, the queue at each gateway connection is 160 packets or 16 KB. We conjecture [5] that in our simulations, $\mathcal{P}_q$, where

$$\mathcal{P}_q \;=\; \frac{k_1}{\beta_q^{\alpha-1}} \tag{5.6}$$

shows how the packet losses behave on a queue of size β_q. In Eq. 5.6, k_1 is a constant and α is the shape parameter of the Pareto traffic (equal to 1.2). By increasing the queue size (β_q) by 32% (i.e., β_q is equal to 512 KB), the packet losses are cut in half to 5%.

5.4.5 On exponential traffic

In Table 5.3 we illustrate, for each gateway, the packet dropping rate and link bandwidth utilization. The aggregate flow interarrival mean is 600 msec. The 99% confidence interval of the load balancing of the system, σ (as defined in Eq. 5.1), is $[1.71\%, 2.12\%]$.

[5] After a discussion with Prof. Predrag Jelenkovic.

Aggregate F_{int}	**Gateway 1**	**Gateway 2**	**Gateway 3**
	Link utilization (%)	Link utilization (%)	Link utilization (%)
0.6 sec	$[81.14, 82.08]$	$[80.81, 81.62]$	$[80.49, 81.24]$
6 sec	$[64.7, 65.8]$	$[63.3, 64.7]$	$[62.6, 64.0]$
	Dropping pkt rate (%)	Dropping pkt rate (%)	Dropping pkt rate (%)
0.6 sec	$[9.62, 10.23]$	$[9.64, 10.29]$	$[9.53, 10.12]$
6 sec	$[3.7, 4.1]$	$[3.4, 3.9]$	$[3.3, 3.7]$

Table 5.2: Link utilization and dropping packet rates at each gateway. We consider Pareto traffic with bitrate 64 kb/s, average idle time 325 msec, and average burst time 315 ms. F_{int} is the mean aggregate flow interarrival time. The confidence interval is 99%.

Gateway 1	**Gateway 2**	**Gateway 3**
Link utilization (%)	Link utilization (%)	Link utilization (%)
$[66.68, 69.13]$	$[65.85, 68.04]$	$[65.32, 67.63]$
Dropping pkt rate (%)	Dropping pkt rate (%)	Dropping pkt rate (%)
0	$3 * 10^{-3}$	$2 * 10^{-3}$

Table 5.3: Link utilization and dropping packet rates at each gateway. We consider exponential traffic with peak bitrate equal to 64 kb/s. The aggregate flow interarrival mean is 0.6 sec. The confidence interval 99%.

Therefore, in both the Pareto and exponential case, the greedy algorithm performs well: The σ ranges from 1.7% to 4.6%, with 0% perfect load balancing.

The results in Tables 5.2 and 5.3 indicate that the admission control aggressively schedules the Pareto flows, which results in higher bandwidth utilization, at the cost of higher packet dropping rates (LRD effect). In some tests, the dropping rate is around 10%, which is an unacceptable level for many services. In case of exponential flows, keeping the same aggregate flow interarrival time, the bandwidth utilization is lower than in the Pareto case with lower packet losses.

(T_m,S)	Link utilization (%)	Load balancing (%)	Dropping pkt rate (%)
(60,400)	$[30.74, 31.43]$	$[2.29, 3.52]$	$[0.0676, 0.12]$
(30,400)	$[36.49, 37.28]$	$[2.61, 3.76]$	$[0.19, 0.28]$
(3,400)	$[80.81, 81.62]$	$[1.63, 2.52]$	$[9.64, 10.29]$

Table 5.4: Performance over time intervals (T_m(s),S(ms): link utilization and dropping packet rates at each gateway and load balancing across gateways. We consider Pareto traffic with peak bit rate 64 kb/s, an average idle time 325 msec and an average burst time of 315 ms. The holding time has mean equal to 300 sec. The aggregate flow interarrival is 600 ms.

Table 5.4 shows the packet dropping rates and link utilization for a range of values of T_m. Table 5.4 also includes the load balancing measurements of the system. Hence, by increasing the interval T_m and keeping the sampling period fixed, the admission control policy becomes more conservative, since the gateway uses a longer time period for its traffic measurement. As we describe in Section 5.3.3, the gateway estimates its load as the maximum over the averages (that it computes for each sample during that period). For larger T_m, the system estimates a higher utilization and therefore becomes more conservative. As expected, it results in lower packet dropping rates.

Let us compute the protocol overhead in the wireless LAN for n_g of three and ten. As before, the announcement policy is R. According to Eq. 5.3, the overhead depends on the *aggregate* flow interarrival mean. For an aggregate flow interarrival mean of 6 sec, and packet size of 100 bytes, the traffic overhead contributes 0.93 kb/s to the wireless LAN. Similarly, for a system with 10 gateways, the protocol overhead is 1.86 kb/s. For an aggregate flow interarrival with a mean equal to 600 msec, the protocol overhead increases by exactly a factor of 10.

5.5 Conclusions and future work

In summary, we presented a framework that enables collaborating mobile hosts to share their network connections in order to increase their QoS and data availability. We discussed the basic components of the system and analyzed their performance through simulation results. The connection sharing across the hosts is characterized by a bandwidth utilization varying from $21\% - 82\%$ and a packet dropping rate from $0\% - 10\%$ depending on the system parameters as we described in Section 5.4. The gateway selection mechanism achieves load balancing across the gateways. The greedy algorithm performs well: the load balancing metric, as defined in Eq. 5.1, ranges from 1.7 to 4.6% (where 0% means perfect load balancing).

We have implemented a prototype that operates as a multicast application. The users issue requests for Internet access by sending multicast messages to the group. The users with available network connections can view the requests and may explicitly select which ones to respond to, as described in Section 5.3. A desirable extension of the current prototype is to apply a more generic approach that requires less user interaction. We are considering possible extension of the IGMPv3 and ICMP router advertisements. Most of the multicast routers support IGMPv2 [34]. IGMPv3 provides the additional capability of joining a multicast group for a specific source [13]. Here, we describe a slightly more general case, in which join takes place in per "channel" or multimedia layer basis. More specifically, in case of multimedia applications that use layered video and audio, a gateway can be in charge of a given video layer or audio. A gateway joins a multicast group for a specific video layer or audio (source) and then multicasts it to the wireless LAN. In several multi-resolution compression schemes such as MPEG, subband codding, there are dependencies among

the layers of a video. For example, the layers of a video have a certain order and these schemes require the video receiver to decode a complete subset of consecutive layers starting from the first one. In those cases, we need to provide a control mechanism to coordinate the assignment of the video or audio source to available gateways.

As mentioned in Section 5.1, reduction in the transmission of replicated data is a motivation for connection sharing. As we illustrate in the following scenario, this results in better utilization of the bandwidth of the wireless WAN connections. In Figure 5.1, users need to teleconference with colleagues over the Internet. We assume the support of layered multimedia data sources. Host A joins the multicast discussion. Due to its low bit rate wireless WAN connection, A cannot receive more than one layer of the video stream. Thus, it listens to the first channel that transmits the first layer of the video (here S_1). Later, host E joins the discussion. Similarly, E can also afford only one layer of video. However, instead of listening to the first channel that corresponds to the first layer of the video, it listens to the second one, as soon E becomes aware of A. They forward to each other the layer just received via multicast from the other user. Both A and E decode the two layers, and the video quality increases substantially. This idea can be extended as more hosts join the multicast discussion. Note that if A and E broadcast the layers S_1 and S_2 respectively, then all the remaining hosts in the ad hoc network will be able to receive both the layers. A similar scenario applies in the case of channels with a different source.

This approach can be extended in a number of ways:

- A generalized selection policy enables the user to define preferences and restrictions in the choice of the gateway. Moreover, the selection policy needs to consider the dynamic changes in the environment and fault tolerance issues.

For example, it may periodically look for another gateway to have as backup in case the connectivity with the current gateway is broken due to mobility.

- As in information sharing (in Chapter 2.2), a micropayment mechanism is needed to stimulate cooperation and encourage hosts to lend their Internet access. A peer that accesses the Internet via a gateway periodically pings and pays the gateway using electronic checks or tokens during the service.

- Performance evaluation of the network connection discovery protocol enhanced with fault tolerance, a micropayment mechanism, and device mobility need to be done.

Chapter 6

Bandwidth sharing in video on demand

In this chapter, we investigate bandwidth sharing in S-C mode for a video-on-demand application. The server operates in a multi-disk environment and provides streaming data to clients with different service capabilities. We motivate the disk bandwidth sharing in this setting and discuss the disk model. The server shares the disk bandwidth among the clients based on their capabilities and service requirements. The server applies a scheduling algorithm for the retrieval of streams of clients. We present the scheduling algorithm and discuss how it can improve the disk bandwidth utilization of the server.

6.1 Introduction

In Chapter 1.2, we introduced the bandwidth sharing in S-C mode and described the environment of a video-on-demand streaming server that operates in a multi-disk

environment. We consider a heterogeneous environment where not all the clients can take full advantage of the highest quality of objects due to the constraints in the hardware of their devices, network, and access methods. Their capabilities and constraints may change dynamically. The server shares the disk bandwidth by dynamically reallocating it among the clients based on their current capabilities and service requirements.

The design of such large-scale systems involves several challenging tasks, including the satisfaction of real-time constraints of continuous delivery of video objects as well as quality of service (QoS) requirements. A great deal of work has been performed in the area of VOD server design [37, 36, 88, 87, 42]. We consider the retrieval of variable bit rate (VBR) video streams from a VOD storage-server in a *multi-disk environment*. An attractive feature of the use of VBR video, as opposed to constant bit rate (CBR) video, is the constant quality of video that can be provided throughout the duration of an object's display. However, VBR video exhibits significant variability in required bandwidth. This variability can affect resource utilization in the system and complicate scheduling of both the transmission of objects over a communication network and their retrieval from the storage subsystem. Several papers deal with the support of VBR video for networking applications [28], some can be extended for use in the storage-server domain. However, an essential difference in this case is that a storage server has *a priori* knowledge of the video traces which can aid in further improvements of the design.

In many multimedia systems different levels of quality of service (QoS) can be supported through the use of the multiresolution property of video (and/or images). Figure 6.1 illustrates an example of a layered image that uses the multiresolution

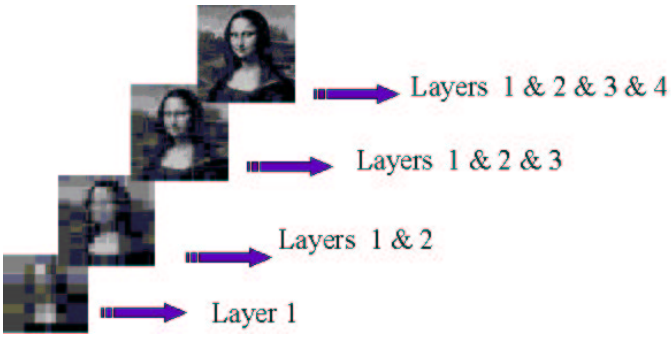

Figure 6.1: Example of a layered image. There is a hierarchy of images, each compressed with an additional layer that improves the resolution.

property. There is a hierarchy of images in different number of layers.

Below we present a scalable VOD server for a heterogeneous environment that provides statistical service guarantees, and propose scheduling techniques for retrieval of VBR video that exploit the multiresolution property of compressed video streams. An example of such a system includes a video server delivering videos over the Internet, with users potentially requesting service from different types of hosts. We define the *scalability* characteristic, as it applies to a VOD system, as the capability to adjust to changes in storage and network resources, and workload, as well as the capability to deliver video data at different levels of QoS guarantees.

In general, the video server faces fluctuations in workload, network congestion, and failure of server or network components. At the same time, the users, while being served, are subject to changes in their sustained bandwidth requirements due to network congestion, failure of network components, or host mobility. We define the *sustained bandwidth* of a user, in a certain time period, as the rate at which the user is expected to effectively receive the data in that time period, which corresponds to a certain video quality profile. The changes in the sustained bandwidth of a user might

be due, for instance to network congestion. Seshan et al [89] describe mechanisms for estimating the sustained bandwidth of a host.

The flexibility that scalable compression techniques provide, in adjusting the resolution (or rate) of a video stream at any point after the compressed video object has been generated, can be of great use in designing scalable video servers. Various video compression schemes, such as subband coding and MPEG-2, provide such a multiresolution property. The multiresolution property has been utilized, for instance, in previous work on dynamic adjustment of resolution of video streams being transmitted through a *communication network* [26] based on available network resources. Although this approach has produced good results in utilization of available communication network bandwidth, there are several difficulties with adapting it to solving similar problems with utilization of disk bandwidth resources in a *storage* subsystem. In [96], a framework is proposed for a layered substream abstraction with highly scalable compression algorithms to support this abstraction. Finally, simulations in [18] show that the use of scalable video with Laplacian and pyramidal coding can greatly increase the I/O bandwidth demand that can be sustained, and decrease the waiting time for start of new requests for video objects, as compared to the use of full-resolution non-scalable video.

We focus on the scheduling of data retrieval and approaches to resolving disk bandwidth congestion under expected and unexpected changes in I/O bandwidth demand. One difference from other related research is that the scheduling of the retrieval adapts to the storage availability and network resources, as well as to the user's bandwidth requirements, which may change dynamically. It is introduced in a dynamic rate-distortion context for a multi-disk environment.

Specifically, we consider techniques that dynamically adjust the resolution of video streams in progress, in order to adjust to fluctuations in workload and resource availability while satisfying given QoS constraints and utilizing system resources efficiently. We propose resolution adjustment and load balancing techniques which address: (a) different causes of fluctuations, which include VBR property of video objects, the use of VCR functionality, as well as failure of system components, (b) the extent and duration of overflow of disk bandwidth demand beyond the available resources, (c) predictability of future I/O bandwidth demand, and (d) variations in QoS requirements. The resolution adjustment algorithm deals with short-term fluctuations in the workload and takes advantage of the multiresolution property of video by adjusting the resolution of video streams in progress on a per-single-disk basis. The load-balancing algorithm, on the other hand, deals with long-term fluctuations in workload as well as extensive overflow of I/O bandwidth demand, and shifts the load between multiple disks in the system. This algorithm takes advantage of both the multiresolution property of video and replication techniques. Furthermore, it also deals with changes in users' sustained bandwidth requirements. We emphasize that the success of resolution adjustment techniques, in general, depends on the number of resolutions available, the workload on the system, and the variability of the VBR video streams. For instance, in the case of subband coding, a large number of layers can be supported, whereas in MPEG-2, the number of layers is restricted to a maximum of 3.

This work has also appeared in [71, 70].

6.2 System description and background

In this section we first present the disk model and discuss the data retrieval and disk bandwidth. We then briefly review the notion of scheduling of video requests in cycles or groups and then discuss data layout issues. Throughout the chapter we use the term *stream* to refer to the delivery of a given layered video object at a given time.

Notation	Description
L_d	set streams whose data blocks are stored on disk d
τ_{seek}	max seek time to move the disk head between two extreme inner and outer cylinders
τ_{rot}^{avg}	rotational latency
τ_{rot}	reading the necessary number of data blocks in a track
B_{track}	number of bytes per track
B	effective bandwidth of a single disk
τ_w	size of an interval (in units of time)
T_{cycle}	size of a cycle (in units of time)
$X_{i,d}(c)$	r.v. bandwidth requirement (full resolution) of stream S_i on disk d during cycle c
$X_{i,d}^k(c)$	r.v. bandwidth requirement of the first k layers of stream S_i on disk d during interval c
$fb_{i,d}(w)$	sustained bandwidth of stream S_i on disk d during interval w
μ	mean bandwidth requirement (for full resolution) of a stream
μ^L	mean bandwidth requirement that corresponds to retrieval of L multiresolution layers

Table 6.1: Notation and description of the parameters of the disk model.

6.2.1 Data retrieval and disk model

To achieve efficient use of available disk bandwidth, it is common to organize the scheduling of streams into (time) cycles or groups, as in [16]. In their simplest form,

cycle-based schemes deliver in each cycle the data that was read in the previous cycle. During each time period, data for each active stream is read from the disks into main memory while, concurrently, the data read during the previous cycle is transmitted over the network to display stations. The motivation for this organization is to provide opportunities for seek optimization [16]. Note that our bandwidth reallocation techniques are not restricted to cycle-based scheduling, but are presented in that context for simplicity of exposition.

In the remainder of this chapter, we will consider scheduling of data retrieval and overflow management in time intervals composed of an integral number of cycles. This slight generalization will allow us to control the scale on which bandwidth reallocation is performed, as explained in more detail in Section 6.2.4. Thus, an interval w starting in cycle c, i.e., $[c, c+\tau_w]$, whose size is τ_w (in *number of cycles*) is composed of consecutive, non-overlapping cycles (c, $c+1$,.., $c+\tau_w$), where continuous playback of a video with quality Q_L can be guaranteed if all blocks corresponding to

$$\{X^L_{i,d}(j), \quad \text{where } j \text{ is a cycle and } j \in [c, c+\tau_w]\} \tag{6.1}$$

have been retrieved (from disk d) by the end of that interval. $X^L_{i,d}(j)$ is a random variable that indicates the bandwidth requirement of stream S_i on disk d, during cycle j corresponding to video quality Q_L. Lastly, we define the mean bandwidth requirement of stream S_i on disk d during interval w corresponding to video quality Q_L as follows:

$$\mu^L_i(w) = X^L_{i,d}(w) = \frac{\sum_{j=c}^{c+\tau_w} X^L_{i,d}(j)}{\tau_w} \tag{6.2}$$

6.2.2 Disk Model

We now introduce a simple disk model used in our system. During each cycle of size T_{cycle}, each stream retrieves a variable number of blocks corresponding to its bandwidth requirement for that cycle. Given the cycle-based scheduling scheme, the amount of time needed to retrieve data blocks corresponding to x streams scheduled during that cycle includes a maximum seek τ_{seek}, which is the time to move the disk head between the extreme inner and outer cylinders of a disk, rotational latency τ_{rot}^{avg}, and the transfer time for each of the x streams that is attributable to reading the necessary number of data blocks, τ_{rot}. Note that the amount of data that has to be retrieved per stream in order to maintain continuity in the data delivery is $\mu\ T_{cycle}$, where μ is the *mean* bandwidth requirement of each stream. Then, the time to read this data is

$$\frac{T_{cycle}\ \mu}{B_{track}}\tau_{rot} + \tau_{rot}^{avg}, \tag{6.3}$$

where B_{track} is the number of bytes per track, and the constraint that there must be time in a cycle to read that much data for x streams is

$$\tau_{seek} + x(\frac{T_{cycle}\ \mu}{B_{track}}\tau_{rot} + \tau_{rot}^{avg}) \leq T_{cycle}.$$

We assume that

$$\tau_{rot}^{avg} = \frac{\tau_{rot}}{2}, \tag{6.4}$$

which then gives us a bound on the number of streams that can be serviced in one cycle on one disk: We have assumed that the data blocks of the same video object that correspond to one retrieval unit are stored contiguously on the disk. Therefore, the rotational latency corresponding to retrieval of the last fraction of the data block,

i.e., the one that does not necessarily make up an entire track is on the average $\frac{\tau_{rot}}{2}$. We have also assumed that zero latency reads [86] are possible for the full size tracks and that B_{track} is a constant parameter, i.e., we do not consider zones here.

$$x \leq \frac{T_{cycle} - \tau_{seek}}{0.5 + \frac{T_{cycle}\mu}{B_{track}}} \frac{1}{\tau_{rot}}.$$

The effective bandwidth of a disk is then $B = x\ \mu$.

6.2.3 Data layout and partial replication

Below, we briefly describe the basic notion of replication schemes in disk-based systems and present the data layout of our server. In general, a replication scheme keeps, for instance, two copies of each object, termed a primary copy and a backup copy, on two distinct disks in a system. The existing variations of replication schemes (in the context of traditional disk-based system) differ mostly in how/where they place the backup copy. This, of course, results in different performance and reliability characteristics of the disk subsystem.

We extend the notion of a traditional replication across disks scheme [8] for use in a multiresolution video server environment to instead support backup copies of videos in a lower resolution than the primary copy. That is, a backup copy of a video object is not necessarily identical to its primary copy; however, all resolution layers which are replicated are identical to their primary copy counterparts and correspond to a predetermined video quality level Q_L. Thus, the backup copy is composed of the first L layers of each video segment ($L = 1, .., max$). We refer to this form of replication as *partial* replication.

In general, replication provides opportunities for better disk bandwidth utilization, as will become more apparent in Section 6.3, since it provides: flexibility

to service a stream by partial simultaneous retrieval of data from multiple disks in the system, when there is not sufficient bandwidth to serve that stream from any one disk, and load-balancing opportunities, i.e., opportunities for shifting some of the load from one disk to another. Although much of the discussion on scheduling which follows is not restricted to a particular replication scheme, in Section 6.4 we present a scheme which is specific to *chained declustering* [47, 41]. We describe this particular replication scheme next.

In the traditional chained declustering layout, at any point in time, two physical copies of a data fragment, termed the primary and the backup copies are maintained. If the primary copy of a fragment resides on disk D_i, then the backup copy of that fragment resides on disk D_{i+1} mod N_d. N_d is the number of disks in the system. Under dynamic scheduling, a server can choose to retrieve a data block from either of the disks containing a replica of that data block. Thus, in our system, by (re)scheduling some data retrieval from disk D_i to disk D_{i+1}, the server can create available bandwidth space on disk D_i, which can be used for retrieval of other blocks corresponding to streams that were originally scheduled for service on either disk D_i or disk D_{i-1} and consequently can be transferred for service to disk D_i. The provision of a backup copy in a *lower* resolution can also be applied to the chained declustering scheme. Figure 6.2 illustrates an example system using chained declustering and partial replication.

6.2.4 Interval-based retrieval and admission control

Many different approaches to allocating (or reserving) bandwidth, at admission control time, for servicing video streams are possible. The scheduling framework de-

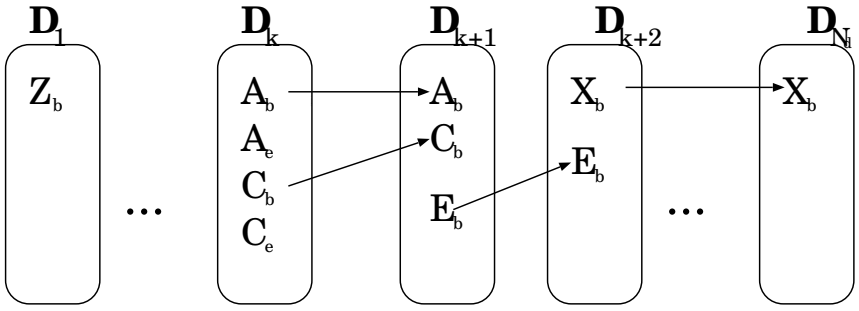

Figure 6.2: Chained declustering with partial replication. The segments of the form A_b correspond to the first b resolution layers of video object A, and have been replicated. For the A_e segments which compose the remaining or enchanced resolution layers, only a single copy is kept. The arrows indicate a potential shift of the retrieval of data blocks from one disk to the next.

scribed in Sections 6.3 and 6.4 are not limited to a specific approach; however, in order to focus the discussion better, we assume the following bandwidth reservation scheme, performed at admission control time. For each interval, the bandwidth reservation is performed by approximating the bandwidth requirements of the video during that interval with its *average* over that interval, i.e., $\mu_i^L(w)$. The average bit rate reservation can utilize resources more efficiently than the peak rate reservation, but leads to potential congestions due to the variability of the stream bandwidth demand. The significance of underutilization of resources due to peak rate reservation in a piecewise (i.e., interval-based) manner, as opposed to average rate reservation with possibility of overflow depends on the characteristics of the video trace and the size of the intervals. One of the foci of this work is a technique for alleviating these congestions without significantly affecting the quality of service provided to the users. In order to limit such congestion and be able to provide an "acceptable" QoS level, it is necessary to perform some form of admission control.

During *admission control*, it is determined whether there are sufficient available resources for retrieval of a newly requested video stream. This is done by considering $\mu_i^L(w) \quad \forall w$ and determining, on per interval basis, whether there is sufficient bandwidth available in the system for support of this new stream. The reservation of bandwidth resources made during the admission control phase, provides statistical guarantees for the retrieval of the streams, and only approximates the actual retrieval schedule, without preventing potential congestion. Hence, a bandwidth re-allocation mechanism that dynamically alleviates congestions is required. The scheduling techniques presented here are not limited to a specific admission control scheme. Thus, we only assume that the admission control policy results in statistical QoS guarantees as follows: A server can guarantee, to at most n_L users, a video quality of Q_L, with probability P_L, where various methods for determining n_L and P_L can be constructed, but are outside the scope of this work.

A priori knowledge of the video trace allows detection of congestion (before it occurs) due to the reservation performed at admission time, according to the stream's average bit rate requirement. An important issue, in this case, is the choice of interval sizes. The choice of the interval size depends on the guarantees that the server will provide and reflects the tradeoff between "stronger" server guarantees and a more efficient utilization of resources in a statistical service setting. As it was mentioned earlier, a peak rate reservation in a piecewise manner with larger intervals results in a less efficient bandwidth utilization than an average-based reservation. Although, the use of larger intervals may speed up the scheduling mechanism (i.e., fewer but larger intervals can be considered), in the case of the average bit rate reservation, it might sacrifice some amount of service guarantees, depending on the techniques used

to re-adjust data retrieval schedules to resolve congestion (i.e., the average may not be "representative" of the interval and thus result in some amount of congestion that will need to be resolved in order provide a needed QoS).

For the sake of simplicity, we assume equal-size intervals which are the same for all streams and are equal to an integral number of cycles. In general, we will consider intervals on the order of a few seconds. Of course, in order for continuous playback guarantees to be satisfied, we also need to assume that the user provides a buffer of size $2b_{max}\ \tau_w\ T_{cycle}$, at the user's site, where T_{cycle} is the duration of one cycle, b_{max} is based on the user's video quality requirement and is basically the maximum bandwidth requirement corresponding to the quality of service he/she requests. Here, we assume double buffering at the user's site, for ease of exposition. This buffer space will mask the jitter that might otherwise result from altering the data retrieval to be on a per-interval, rather than per-cycle, basis.

6.3 Scheduling of data retrieval

It is the main task of the scheduler to determine the amount of data to be retrieved per stream per cycle and schedule the retrieval for the appropriate cycle and disk. Due to the guarantees made during the admission control phase, the scheduler is ensured that there is available disk bandwidth somewhere in the system to serve all admitted streams with video quality Q_L with probability P_L (see Section 6.2.4). However, the scheduler might be required to re-allocate (or shift) part of the workload across the disks of the system, due to congestion. Recall that, due to replication, some of the data can be retrieved from either of two disks. Furthermore, transferring of some load from one disk to the next one might result in a number of shifts of load between

consecutive disks. That is, a video block of stream S_i, of size b_i, can be shifted from a congested disk D during some time interval T, to disk $\bar{D}$ if: a replica of the video block is also stored on disk $\bar{D}$ **and** the available bandwidth, that exists or could be made available on disk $\bar{D}$, during time T *at least* corresponds to the amount needed to retrieve a block of size b_i. Below, we present a general framework for re-allocating bandwidth to streams in cases of fluctuations in order to dynamically exploit the available bandwidth of the system.

6.3.1 Resolution adjustment on multiple disks

An optimal assignment of data block retrievals to disks minimizes the amount of data that can *not* be retrieved (due to overflow). It can be determined using a "max-flow" algorithm [20]. Papadimitriou [69] discusses a max-flow algorithm with lower bounds on edges.

Graph Generation

We begin with the needed generation of an appropriate graph. Construction of this graph and computation of the corresponding optimum schedule under given constraints requires knowledge of the amount of bandwidth that was *reserved* at admission control time for retrieval of data blocks corresponding to active streams as well as the amount of bandwidth that is *required* by these streams during the time interval in question. The max-flow algorithm is specified on a per-interval basis. We can determine, on a per-interval basis, whether overflow will occur in that interval and run the max-flow algorithm in order to determine a new assignment of streams to disks and the corresponding retrieval schedule which alleviates overflow in an optimal

manner given the optimality criteria stated above, where τ_w is a parameter of the scheduling algorithm. The max-flow algorithm will be applied on a graph $G^L(w) = (V, E)$ (e.g., see Figure 6.3), which can be constructed as follows:

$$\begin{aligned} G^L(w) = (V, E), \;\; V &= \{D_1, D_2, \ldots, D_{N_d}\} \cup \{V_1, \ldots, V_n\} \cup \{s\}, \\ E &= \{(s, V_j), \;\; j = 1, .., n\} \cup \{(V_j, D_i), j \in L_{D_i}, \; i = 1, .., N_d\}. \end{aligned}$$

L_{D_i} is the set of streams whose data blocks are stored on disk D_i and where each node D_i has capacity B, which is the effective bandwidth capacity of disk D_i (see Section 6.2.1). There are n number of nodes V_j, each corresponding to a currently active stream S_j. Each is connected to the start node s (an artificial node) with an edge $(s, V_j) \in E$ whose capacity is bound by l_j and u_j, i.e., the flow on this edge, (s, V_j), is $l_j \leq f_j \leq u_j$. If the required bandwidth of stream S_j during interval w, $fb_j(w)$, is less than $X^L_{j,d}(w)$, then $l_j = u_j = fb_j(w)$. Therefore, if the adjusted sustained bandwidth of a user is less than or equal to the mean bandwidth that the server has reserved for that user, then the retrieval corresponding to S_j will be at least of size $fb_j(w)$. On the other hand, if the user's required bandwidth is higher than $X^L_{j,d}(w)$, then $l_j = \mu^L_j(w)$ (refer to Section 6.2.1) and $u_j = fb_j(w)$. Furthermore, if the first k layers of a stream S_i are stored on disk D_j, then there is an edge (V_i, D_j), with its lower and upper capacity equal to 0 and $\mu^k_i(w)$, respectively. The max-flow algorithm will assign a flow to each edge (s, V_i) which represents the bit rate that the server will deliver to S_i during interval w. Note that, a flow > 0 on edges (V_j, D_i) indicates that a fraction of stream S_j will be retrieved from disk D_i.

The max-flow algorithm produces one bandwidth allocation from all possible ones that retrieves the maximum total amount of data for the active streams with respect to the guarantees of certain quality of service and constraints on the playback

rates of the users. We should emphasize that the feasibility of a flow on the above

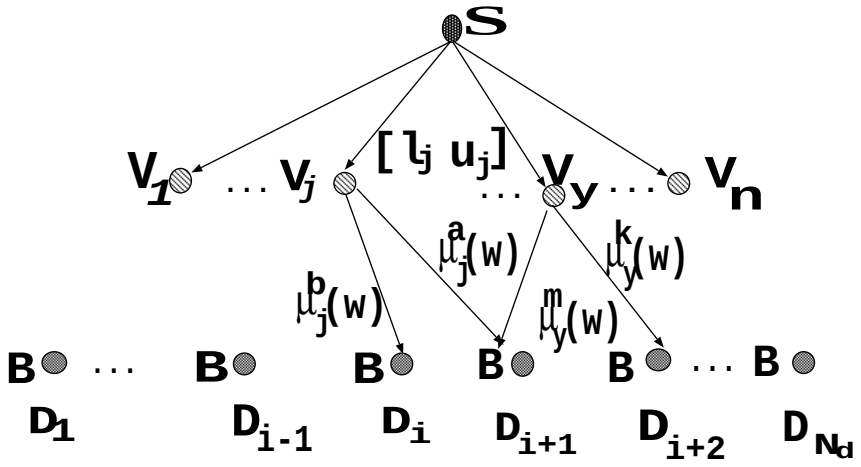

Figure 6.3: Per-interval basis retrieval for each stream by using the max-flow algorithm. The max-flow is applied on a graph generated based on the current streams, their requirements during that interval, and data layout.

graph will depend on the statistical guarantees the server makes at admission control time.

Lastly, the time complexity of the max-flow algorithm is $O(|V||E|log|V|)$, if Sleator and Tarjan's algorithm [95] is used. Then, given that the maximum total number of streams in the system is $n_f N_d$, the time complexity becomes

$$O((N_d + N_d n_f + 1)(3 n_f N_d)(log(N_d n_f + N_d + 1)) \tag{6.5}$$

or $O((n_f N_d)^2(\log(\max(N_d, n_f))))$. The max-flow algorithm can run at the beginning of an interval to compute the optimal bandwidth allocation for the next interval. Of course, the tradeoff here is between having sufficient time to run the max-flow algorithm (i.e., making the intervals longer) and the amount of buffer space needed

at the user site, which grows with the interval length.

6.3.2 Resolution adjustment on per-disk basis

In this section we consider the problem of per-disk re-adjustment of the data retrieval schedule. As previously mentioned in Section 6.1, the motivation here is to be able to resolve short-term overflow problems relatively quickly and on a per-disk basis.

One of the important requirements of bit rate re-adjustment is to minimize the distortion that the streams will experience as a result of the re-adjustment process. We define the distortion of a stream S_i during time interval t, due to a decrease of information retrieved from $b_i(t)$ to $b_i(t) - \beta_i$ as a non-decreasing (cost) function R_i: $R_i(\beta_i, t) = \tau_i \, \Delta_i(\beta_i, t)$, where $b_i(t)$ corresponds to a feasible retrieval schedule of the requested resolution, τ_i is a distortion tolerance factor, and $\Delta_i(\beta_i, t)$ is the distortion due to retrieving only $b_i(t) - \beta_i$ instead of $b_i(t)$ amount of data. A retrieval schedule is feasible when it does not violate the disk bandwidth constraints while guaranteeing normal playback at the receiver without discontinuities. The distortion tolerance factor, τ_i, is a function of the QoS requirements of the user, the feedback mechanism, and the subjective distortion measures on specific video segments, e.g., the service of streams in fast forward or rewind mode, which can tolerate larger distortions. Each layer k is associated with a distortion measure, $\delta_i(b_i^k(t)) = r_k$, where $b_i^k(t)$ corresponds to the amount of data that needs to be retrieved for the k^{th} layer of stream i at time interval t. Distortion measures may vary from a simple mean squared error (MSE) estimate to more complicated perception-based functions. Therefore, r_k indicates the resolution improvement, if in addition to the first $k-1$ layers of video retrieval, we also retrieve the k-th layer. When β_i amount of data is not retrieved, the decoder

will be able to decode only the first λ_i layers (instead of all λ_{max} layers that compose the requested resolution data blocks), and therefore, the distortion that the user will experience during interval t will be as follows:

$$\Delta_i(\beta_i, t) = \sum_{k=\lambda_i+1}^{\lambda_{max}} \delta_i(b_i^k(t)), \quad where\ \lambda_i = \max_k(\beta_i \leq \sum_{l=k}^{max} b_i^l(t)). \tag{6.6}$$

Here, the cost function R_i quantifies the QoS of a stream S_i.

The server runs the Resolution Adjustment (RA) algorithm (given below) on a future time interval t after detecting a congestion in that interval in order to determine the proper bandwidth re-allocation. Let us assume that for that interval t, B_{of} is the disk bandwidth overflow on the disk in question, b_i^{max} is the retrieval unit corresponding to requested resolution of the video, and b_i^{base} is the retrieval unit corresponding to some minimum acceptable resolution, as specified by the user. For instance, in the notation of the previous section, if the required bandwidth of stream S_i during interval t $fb_i(t)$, is higher than $X_{i,d}^L(t)$, then $b_i^{base} = \mu_i^L(t)$ and $b_i^{max} = fb_i(t)$; otherwise $b_i^{base} = b_i^{max} = fb_i(t)$. Then, RA can be formulated as follows:

- Find β_i $i = 1, \ldots, n$
- Minimize $max_i\{R_i(\beta_i)\} - min_i\{R_i(\beta_i)\}$
- such that:
  1. $\sum_i \beta_i = B_{of}$
  2. $0 \leq \beta_i \leq b_i^{max} - b_i^{base}$, where β_i is a non-negative integer, $i = 1, 2 \ldots, n$, where n is the number of active streams on the disk.

In the above formulation, we dropped the time dependence in the notation in order to simplify it.

The RA problem is a fair resource allocation problem that can be solved using the FAIR algorithm, as given in [49], of complexity $O(n \log(\max(n, B_{of})))$, where n is the number of active streams on a disk and B_{of} is the disk bandwidth overflow. If RA does not have a feasible solution, we can attempt to shift the overflow to the remaining disks in the system as described above.

6.4 Application of max-flow

In this section we consider sources of congestion due only to changes in sustained bandwidth of users, and then evaluate the performance of the resulting server, termed $S_{scalable}$. This evaluation focuses on the retrieval scheduling/overflow management policy formulated in Section 6.3 as a max-flow problem. For the purposes of comparison we use a baseline server, termed $S_{independent}$, which does not take advantage of replication (and here specifically chained declustering with partial replication). In $S_{independent}$ all disks are independent, i.e., a retrieval for a particular stream is always scheduled on a specific disk without the flexibility of replication that exists in $S_{scalable}$, which allows shifting of the load across disks. The metric used in evaluating the overflow management policy is the percentage of newly available bandwidth that we are able to utilize. We explain this in more detail below. The results of this evaluation have been obtained through simulation, where we use the disk model given in Section 6.2.1. The parameters of that model used in this section appear in Table 6.2 [52, 86].

Below we illustrate that, through the use of chained declustering with partial replication, the server is not only capable of resolving overflow due to fluctuations, for instance, of VBR video compression, but is also sufficiently flexible and can take

τ_{seek}	30 msec
τ_{rot}	10 msec
T_{cycle}	530 msec
B_{track}	100 KB

Table 6.2: Parameters of the disk model.

advantage of the available bandwidth that results from reductions in the sustained bandwidth of some users. For the purposes of the following discussion, we term the users with decreases in sustained bandwidth requirements in an interval w as "degraded users"; similarly, we term the users with increases in sustained bandwidth requirements in an interval w as "upgraded users". The simulations described below aim to evaluate the bandwidth re-allocation process of $S_{scalable}$ and $S_{independent}$ by computing the portion of bandwidth (able to be scheduled) that was freed by the degraded users. This can potentially be re-assigned to the upgraded users in order to satisfy their requests for increases in bandwidth, in an interval w.

Let us consider some such interval w. In order to focus on the comparison between overflow management of the two servers, we fix the load on each disk of both server to be the same. Specifically, for our simulation we consider a cluster of 45 disks. The load on each disk, i.e., the total amount of bandwidth that is needed during interval w, is determined based on a uniform distribution with values ranging in 38 Mb/s to 60.8 Mb/s.

As previously mentioned, $S_{scalable}$ uses the max-flow algorithm; Figure 6.4 illustrates the graph for this max-flow algorithm that can be created as follows: Each disk corresponds to a node, and each stream to an edge. There is a start node as well as a sink node. For each upgraded stream, there is an edge that connects a

node corresponding to a disk on which the stream is at least partially scheduled with an artificial start node. That start node has capacity equal to the increment in the bandwidth requirement of that stream. For each degraded stream, there is an edge that connects a node corresponding to a disk on which the stream is at least partially scheduled with a sink node, with capacity equal to the decrement in the bandwidth requirement of that stream. For each node that corresponds to a disk, there is an edge that connects it to the node that corresponds to the disk on its right. For the last disk, its node is connected with the node of the first disk. The capacity of this edge is equal to the amount of bandwidth that can be transferred during this interval w from the disk in question to the next disk on its right. This depends on the degree of replication used in the system, as well as the bandwidth requirement of streams accessing the disks in that interval. Recall that in chained declustering, the disk logically to the right of disk i contains copies of data stored on disk i.

This graph is somewhat different from the one described in Section 6.3.1, due to the source of change in workload, which in this case is due *only* to the changes in sustained bandwidth of degraded and upgraded users. In addition, in this case we are considering a specific form of replication, namely that of chained declustering. Moreover, this simplification in the formulation of the max-flow algorithm reduces the complexity of the solution to $O((n_f(N_d)^2)(\log N_d))$.

The max-flow algorithm returns the maximum amount of bandwidth, as a fraction of the total amount of bandwidth that has been released by the degraded users, which can be re-allocated to the upgraded users. Note that, $S_{independent}$ is able to assign additional bandwidth to the upgraded users only if the disk on which they are served has some available bandwidth due (in this case *only*) to the reduction in the

bandwidth requirements of the degraded users. Both servers assign to the degraded users bandwidth equal to their new reduced bandwidth requirements.

We illustrate in Figure 6.5 the amount of bandwidth that was released by the degraded users, which can be re-allocated to the upgraded users, as a function of the skew of the changes in users' required bandwidth on the disks. The increment/reduction in sustained bandwidth corresponds to the difference between the current sustained bandwidth requirements and the amount of bandwidth reserved at admission control time. Specifically, the total increment (reduction), $load^d_{increment}$ ($load^d_{reduction}$) in the bandwidth requirements on disk d is given by :

$$load^d_{increment} = \frac{C_{increment}}{\phi(d)^\gamma}, \quad load^d_{reduction} = \frac{C_{reduction}}{f(d)^\gamma} \tag{6.7}$$

where $C_{increment}$ and $C_{reduction}$ are constants, and $\phi()$ and $f()$ provide a "one-to-one" mapping of the (N_d) disks to the (N_d) different loads (i.e., in Eq. 6.7 d is in the range 1 to N_d), where N_d is the disk cluster size. As γ increases, the access pattern becomes increasingly skewed. In our simulations we consider the following values of γ : 0.0, 0.25, 0.5, 0.75, and 1.0. Note that the *maximum* total increment (reduction) in bandwidth requirements on a disk is $C_{increment}$ ($C_{reduction}$). In this simulation we consider $C_{increment}$ and $C_{reduction}$ to be 35 Mb/s.

In Figure 6.5, we illustrate the performance of the two servers, $S_{independent}$ and $S_{scalable}$, with full (100%) and partial replication, where, for simplicity of illustration, we have assumed that the degree of partial replication corresponds to the amount of bandwidth reserved for a stream at admission control time.

For the case of partial replication, we assume the degree of replication (i.e., percentage of the retrieval unit that is replicated) to be equal with the percentage of bit rate that is reserved during the admission control per stream. As previously

mentioned, the performance metric is the percentage of released bandwidth (by the degraded users) which can be re-scheduled for use by upgraded users.

The performance of both servers degrades as the skew increases. For example, in the case of $S_{scalable}$ and $\gamma = 0.25$ with 100% replication, 99% of the released bandwidth is re-allocated to upgraded users, whereas with partial replication 97% of the released bandwidth is re-allocated. However, in the case of $\gamma = 1.0$ with 100% replication, 91.6% of the released bandwidth is re-allocated, whereas in the case of partial replication 89.8% is re-allocated. Therefore, as expected, the less uniform the change in requested bandwidth, the more difficult it is to re-allocate the bandwidth.

Furthermore, the gap in performance between $S_{scalable}$ and $S_{independent}$ increases as the skew increases. Even under high skews in changes in bandwidth requirements, $S_{scalable}$ is able to re-allocated a large percentage of the released bandwidth by exploiting replication (or specifically chained declustering in this case) and shifting the increase in load (due to upgraded users) from one disk to another, i.e., one that has the newly available bandwidth due to degraded users. For instance, under full replication both servers have the same performance when $\gamma = 0.0$, whereas when $\gamma = 1.0$, the gap in performance between $S_{scalable}$ and $S_{independent}$ is more than 48.5%. Note also that the performance of $S_{independent}$ decreases almost linearly from 100% to 43% as the skew increases. This is due to the fact that as the skew increases fewer good matches will occur on each disk. By a good (or perfect) match, we mean the case where the total amount of additional bandwidth that is requested by the upgraded users, to be retrieved from a single disk, is approximately or exactly the same as the total amount of the reduction in requested bandwidth (due to the degraded users) on the same disk. In the case of $S_{scalable}$ the higher the degree of replication, the

lower the probability of having a "good match" that affects the performance of the server. For example, under full replication, the performance of $S_{scalable}$ is 100% (for $\gamma = 0.0$) and is reduced to 91.6% (for $\gamma = 1.0$), whereas under partial replication the performance of $S_{scalable}$ is reduced from 100% (for $\gamma = 0.0$) to 89.8% (for $\gamma = 1.0$). Thus, based on the results depicted in Figure 6.5, we can conclude that skew has a greater effect on the performance of $S_{independent}$ than on the performance of $S_{scalable}$.

In summary, due to proper utilization of replicated data, $S_{scalable}$ is more effective at taking advantage of the bandwidth that has been freed in an interval w due to degraded users and assigning it to the upgraded users. Therefore it adapts more effectively to changes in bandwidth requirements or availability of resources. This becomes more critical under higher degrees of skew, that is, there is a larger improvement in the percentage bandwidth that can be "re-allocated" by $S_{scalable}$ as compared to $S_{independent}$. Finally, the higher the degree of replication the less skew affects the performance of $S_{scalable}$. This reduced sensitivity to skew, of course, is achieved at the cost of additional storage space.

6.5 Conclusions

In summary, we consider the problem of delivery of VBR video streams in a VOD server under provisions of statistical quality-of-service guarantees. We present several techniques for re-scheduling the video streams and adjusting bandwidth allocations for streams in progress when fluctuations in workload as well as changes in availability of resources occur while satisfying QoS constraints and utilizing system resources efficiently.

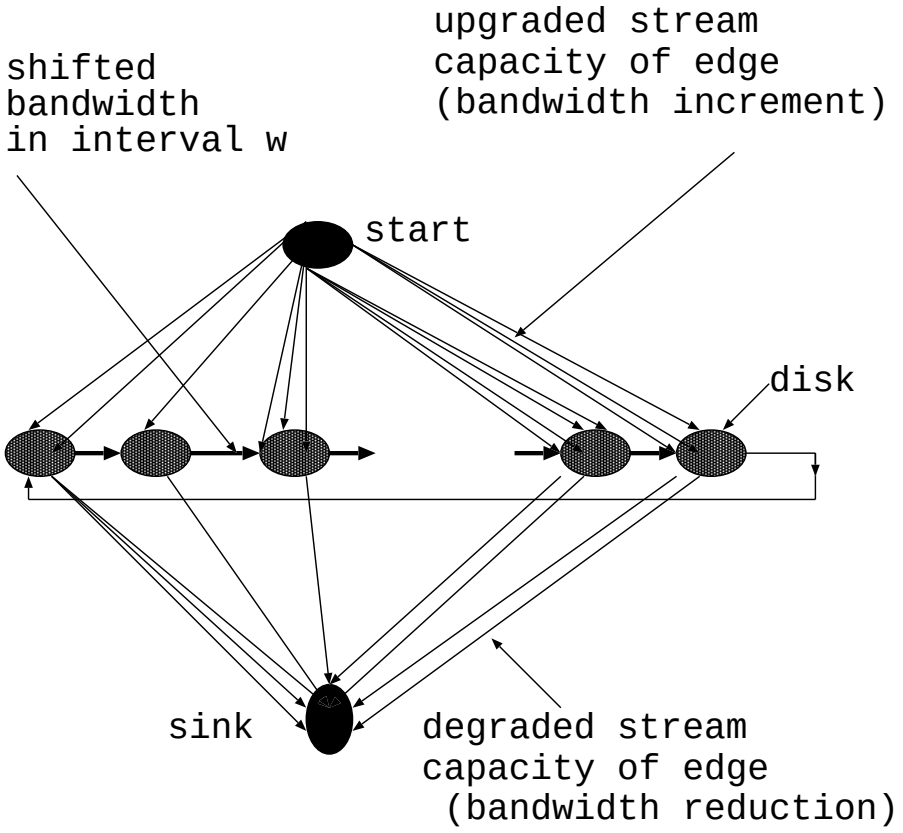

Figure 6.4: By taking advantage of replication in a chained declustering layout, the server shifts the retrieval from one disk to its consecutive to alleviate disk overflow. This graph is generated based on the current streams, their requirements, and data layout. The server applies a max-flow algorithm on that graph to determine the retrieval of each stream during each interval.

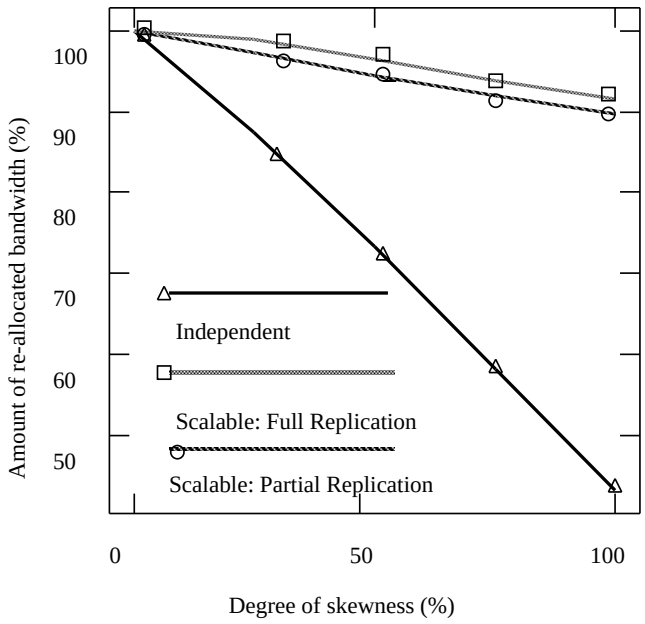

Figure 6.5: Effects of replication on the disk bandwidth utilization: the amount of bandwidth re-allocated to upgraded users as a function of skewness. The Independent scheme corresponds to a baseline server that does not take advantage of replication. For the case of partial replication, the degree of replication is equal with the percentage of bit rate that is reserved during the admission control per stream.

Chapter 7

Conclusions and future work

We conclude this dissertation with a summary of our contributions and directions for future work.

7.1 Summary

The main challenges for this thesis is to accelerate the data availability and enhance the dissemination and discovery of information when hosts face changes in the bandwidth availability and loss of connectivity to the Internet due to host mobility. We propose 7DS that addresses this challenge by providing a novel mechanism that enables wireless devices to share resources in a self-organizing manner, without the need of an infrastructure. 7DS is an architecture, a set of protocols, and an implementation enabling resource sharing among peers that are not necessarily connected to the Internet. Peers can be either mobile or stationary. The focus is on three facets of cooperation, namely information sharing, bandwidth sharing, and message relaying. In the information sharing facet, peers query, discover, and disseminate information.

For message relaying, hosts forward messages to the Internet when they gain Internet access on behalf of other hosts. The system adapts its communication behavior, such as query mechanism, frequency, type of cooperation, based on the availability of power, and bandwidth.

7.1.1 Information sharing and message relaying

For the information sharing and message relaying, we introduce a general framework for the mobile wireless data access. We model several schemes depending on the type of cooperation among nodes, querying mechanism, their energy conservation, host density, and transmission power and evaluated them via simulations. The emphasis is the *transient aspect of information dissemination.*

In our simulations, we considered variations of the P-P and S-C schemes as well as some hybrid ones. We measure the percentage of hosts that acquire the data item as a function of time, and their average delay. For simplicity we fix the data object and assume that at the beginning of each experiment, only one $7DS$ host has the data item. All the remaining hosts are interested in this data item. We also evaluate the message relaying by computing the number of messages that will finally reach the Internet. We found that the density of the cooperative hosts, their mobility, and the transmission power have the most pronounced impact on data dissemination. For a region with the same density of hosts, P-P outperforms S-C with no cooperation among the mobile devices. The simulations indicate that the probability a host querying a data object will acquire it by time t follows the function $1 - e^{-a\sqrt{t}}$ when using FIS. In case of high density of cooperative hosts, the data dissemination using P-P grows even faster. For example, in a P-P setting of 15 hosts with wireless range

of 230 m, after 25 minutes, 99% of the users will acquire the data, compared to just 42% of the users in the FIS. For the same average delay of 6 minutes, a host using FIS will get the data with a 42% probability, whereas using synchronous P, even in a setting of only five hosts per km^2, this probability is double. For lower transmission power, P-P outperforms FIS by 20% to 70%. In the case of only five hosts, the two approaches differ by 3% to 43%, depending on the transmission power.

We can use the simulation results on the impact of the querying mechanism, the energy conservation, host density, type of cooperation, and power transmission to tune 7DS. For example, we showed that the synchronous energy conservation is beneficial. Also, in FIS when the host density is low, the query frequency can be set as large as three minutes without hurting the speed of data dissemination. This is also true in the case of P-P with data sharing and low host density. The scaling properties of data dissemination can give us insight for the design of a wireless information infrastructure. We found that performance remains the same when we scale the area but keep the density of the cooperative hosts and transmission power fixed. Also, for a fixed wireless coverage density, the larger the density of cooperative hosts, the better the performance. In S-C, this implies that for the same wireless coverage density, it is more efficient to have a larger number of cooperative hosts with lower transmission power than fewer with higher transmission power. We showed that message relaying increases the data access by exploiting the host mobility. Moreover, the simulations indicate that it is sufficient to relay a message to one more relay host; the gains from relaying it to more hosts are very low.

We develop an analytical model for FIS using theory from random walks and environments and the kinetics of diffusion-controlled processes. The analytical results

on data dissemination are consistent with the simulation results for FIS. These details of the performance analysis are discussed in Chapters 3 and 4.

7.1.2 Bandwidth sharing in wireless LANs

We investigate the bandwidth sharing mechanism in a wireless LAN. When the bandwidth sharing is enabled in a wireless LAN, the system allows a host to temporarily act as an application-based gateway and share its connection to the Internet. We design a lightweight protocol that discovers a gateway in the wireless LAN and enables hosts to share their connections to the Internet and increase their quality of data. We present the benefits of this system via simulation results. We show that for Pareto flows, the system results in very high gateway connection utilization (around 80%) at the price of a high packet loss rate (around $5\% - 10\%$). Exponential flows exhibit more conservative bandwidth utilization (around 60%) and a very low packet loss rate. The system can load-balance the traffic of the WAN connection across gateways. These details of the network connection sharing protocol and performance results are discussed in Chapter 5.

7.1.3 Bandwidth sharing in multimedia servers

For the bandwidth sharing in a multimedia server, we design a novel server that exploits the multiresolution property of compressed video and replication. We present a novel retrieval technique that takes advantage of the layered information and replication to dynamically reallocate the disk bandwidth. We model the multi-disk environment for different degrees of replication and measure the disk bandwidth utilization. We showed how the system performs in the case of no replication, partial replication,

and full replication as a function of skewness of the user access pattern. In the partial replication only a part of the multimedia objects are replicated. Our retrieval algorithm on full replication results up to double improvement in the disk bandwidth utilization compared to the case of no replication. These details for the retrieval algorithm can be found in Chapter 6.

7.2 Future directions

There are several interesting directions for future research based on the work described in this dissertation. Some of these are extensions of our work, while others are motivated by the more general problem of mobile wireless access in pervasive computing.

7.2.1 Location-dependent applications and services

A fruitful approach to this research would be to develop a general infrastructure for applications, build some of the applications, and then extract a toolkit that other new applications could use. This could accelerate the deployment and use of wireless technologies, and also provide insight to many important issues. We need to build a testbed with a large spectrum of applications to better understand their delay and privacy requirements, as well as information and query locality. Currently, we deploy 7DS on the campus and plan to integrate it with a tour guide, an academic news notification system, and some augmented and virtual reality applications. This testbed can also assist in investigating the challenges we describe next.

7.2.2 Actual traces and models for user mobility, data locality and access patterns

As we mentioned in Chapter 3.2, for user mobility, most of the studies on ad hoc routing protocols use random-walk-based models. The analytical results in Section 4.3 use random walk that differs from the randway model in the time scale of the displacement step. However, it is difficult to speculate the performance of the system for other mobility patterns. Unfortunately, there are not many traces available of actual data access patterns of mobile wireless users or realistic models for different scenarios. In earlier work [73], we simulated a baseline scenario of information sharing in a subway, where users enter the platform, ride a subway car, stop at their destination, and leave. However, the mobility model was oversimplified and the performance of data dissemination tied to that setting. This is a common characteristic of most of the mobility patterns used in simulation studies for mobile ad hoc networks. The development of realistic and general models is imperative. We need to use the extended 7DS testbed to generate real traces about data access, impact of caching, spatial locality of user queries and information, hosts coresidency time (i.e., time two hosts are within wireless range), disconnection from the Internet, and user mobility patterns. We plan to collect these traces in various settings, such as, in a campus, during a seminar, in a conference setting, and in a main street of a town. Then, based on those actual traces derive realistic general models for each setting.

7.2.3 Enhanced energy conservation mechanism

An attractive feature for $7DS$ is a mechanism that would indicate and tune the appropriate interaction (P-P or S-C with active or passive querying) based on several parameters, such as data availability prediction, cooperative or malicious users in close proximity, and battery level. Advertisement messages from servers or other cooperative hosts, location information of gateways or servers, and traffic measurements can provide hints. We plan to investigate how they can be used to improve the energy utilization and the performance of data dissemination and message relaying. We also intend to evaluate how the on interval of the rendezvous-based energy conservation impacts the performance considering packet losses and retransmissions.

7.2.4 Security and micropayment issues

It is crucial for the deployment of cooperative wireless devices to design mechanisms that

- allow them to share resources without compromising user privacy,
- detect malicious attacks, isolate them, and enable the devices to adapt when an attack is detected, and
- stimulate cooperation through micropayments (as discussed in Chapter 2.2).

These issues are challenging not only due to energy constraints, but also due to the lack of continuous access to the Internet or an infrastructure. We need to evaluate the approaches discussed in Chapter 2.1. Also, assuming a richer set of peer-to-peer applications for mobile users, we intend to provide a systematic methodology

for selecting the appropriate micropayment mechanism for a given application. The selection can be based on the degree and frequency of cooperation, the access to the Internet, the guarantees that need to be provided to the users, the amount of losses that can be tolerated, the cost of hardware, and the energy expenditure.

7.2.5 Extending the network connection protocol

As we mentioned in Chapter 5.5, we intend to evaluate the network connection discovery protocol considering fault tolerance issues, micropayment mechanisms, and device mobility.

7.2.6 Generalization of diffusion models for peer-to-peer schemes

In Chapter 4.3, we used the trapping model from the diffusion-controlled processes to model the FIS scheme. We plan to explore diffusion models for other types of interaction (e.g., P-P schemes) among mobile devices and incorporating parameters such as the expiration of data objects. For the performance analysis of data dissemination in Chapter 3.3, we focused on a relatively small time window (e.g., 25 to 50 minutes), and we did not consider data expiration issues. It would be interesting to extend the study for a longer time scale and consider data expiration, data popularity, and their interplay.

7.2.7 Adaptive scalable algorithms and protocols for information discovery and dissemination

Typical examples with respect to adaptation and configuration can be found in Internet protocols. For example, the congestion adaptation in TCP and the network self-configuration in IP multicast. However, they did not have to cope with tight energy and bandwidth constraints, nor user mobility and privacy issues as primary design issues. In several pervasive computing environments, the participants (sensors, mobile, wireless devices, and cameras) can be part of data-centric, mobile, ad hoc networks; they collect, measure, process, query, and relay information. The devices need to have local autonomy and the system self-organizes to minimize the administration overhead. The hosts with partial knowledge of the environment need to make local decisions to achieve a global effect. A fundamental challenge is the design of adaptive, scalable algorithms for information discovery protocols that maximize the system performance (e.g., quality of information, number of requests they serve) while minimizing the constrained resources. The modeling and abstraction of the quality of information in this highly dynamic environment impose complex problems. Determining the relevant data (to query or transmit) depends on a number of factors including which data have been previously transmitted, protocol dynamics, user and device interaction with the environment, user location and information preference, and network conditions (e.g., device failures, constraints, host mobility, error in measurements, and packet losses). Furthermore, the problem becomes more complicated considering the timeliness and redundancy of information.

Bibliography

[1] Wireless and mobility extensions to ns-2. http://www.monarch.cs.cmu.edu/cmu-ns.html.

[2] 7DS. http://www.cs.columbia.edu/~maria/7ds.

[3] Airflash. http://www.airflash.com.

[4] Avantgo. http://www.avantgo.com/.

[5] Adam Back. Hashcash - a denial of service counter-measure. Technical report, August 2002.

[6] Hari Balakrishnan, Srinivasan Seshan, Mark Stemm, and Randy H. Katz. Analyzing stability in wide-area network performance. In *SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, Seattle, Washington, June 1997.

[7] Daniel Barbara and Tomasz Imielinski. Sleepers and workaholics: Caching strategies in mobile environments. In *ACM SIGMOD International Conference on Management of Data*, pages 1–12, 1994.

[8] Dina Bitton and Jim Gray. Disk shadowing. *Proc. of the 14th International Conference on Very Large Data Bases*, pages 331–338, August 1988.

[9] Matt Blaze, John Ioannidis, and Angelos D. Keromytis. Offline micropayments without trusted hardware. In *Proceedings of Financial Cryptography*, Cayman Islands, February 2001.

[10] Josh Broch, David Maltz, and David Johnson. Supporting hierarchy and heterogeneous interfaces in multi-hop wireless ad hoc networks. In *IEEE Proceedings of the Workshop on Mobile Computing (I-SPAN)*, 1999.

[11] Josh Broch, David Maltz, David Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, October 1998.

[12] L. Buttyan and J. P. Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. Technical Report DSC/2001/001, Swiss Federal Institute of Technology, Lausanne, January 2001.

[13] B. Cain, S. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan. Internet group management protocol, version 3. Internet Draft, Internet Engineering Task Force, May 2002. Work in progress.

[14] Paul Castro, Benjamin Greenstein, Richard Muntz, Parviz Kermani, Chatschik Bisdikian, and Maria Papadopouli. Locating application data across service discovery domains. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 28–42, Rome, Italy, July 2001.

[15] Cell-Loc. http://www.cell-loc.com.

[16] M. Chen, D. Kandlur, and P. Yu. Optimization of the Grouped Sweeping Scheduling (GSS) with Heterogeneous Multimedia Streams. *ACM Multimedia '93*, pages 235–242, 1993.

[17] Stuart Cheshire and Mary Baker. Experiences with a wireless network in MosquitoNet. In *IEEE Micro*, pages 44–52, February 1996.

[18] T.C. Chiueh and R. Katz. Multi-resolution video representation for parallel disk arrays. In *Proceedings of ACM Multimedia*, pages 401–409, 1993.

[19] World Wide Web Consortium. The w3c ecommerce/micropayment activity. http://www.w3.org/ECommerce/Micropayments/.

[20] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to algorithms*. The MIT Press, 1989.

[21] James Davis, Andy Fagg, and Brian Neil Levine. Wearable computers as packet transport mechanisms in highly partitioned ad-hoc networks. In *Proc. International Symposium on Wearable Computers*, October 2001.

[22] Whitfield Diffie, Paul van Oorschot, and Michael Wiener. Authentication and authenticated key exchanges. In *Design, Codes and Cryptography*, 1992.

[23] H.-P. Dommel and J.J. Garcia-Luna-Aceves. Network support for turn-taking in multimedia collaboration. In *Proc. of IS&T/SPIE Symposium on Electronic Imaging: Multimedia Computing and Networking*, San Jose, California, February 1997.

[24] Dianne E. Duffy, Allen A. McIntosh, Mark Rosenstein, and Walter Willinger. Statistical analysis of CCSN/SS7 traffic data from working CCS subnetworks. *IEEE Journal on Selected Areas in Communications*, 12(3):544–551, April 1994.

[25] Richard Durrett. *Lecture notes on particle systems and percolation.* Pacific Grove, CA, 1988.

[26] Alexandros Elefteriadis. *Dynamic rate shaping of compressed digital video.* Ph.D. dissertation, Columbia University, 1995. Technical Report CU/CTR/TR 419-95-25.

[27] Chip Elliott. Building the wireless internet. *IEEE Spectrum*, 38(1), January 2001.

[28] A. Elwalid, D. Heyman, T.V. Lakshman, and D. Mitra. Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing. *IEEE Journal on Selected Areas in Communications*, 13(6), August 1995.

[29] Mike Esler, Jeffrey Hightower, Tom Anderson, and Gaetano Borriello. Next century challenges: data-centric networking for invisible computing – the Portolano project at the University of Washington. In *Proceedings of the Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 256–262, Seattle, Washington, August 1999.

[30] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In ACM, editor, *Proceedings*

of the Annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 263–270, Seattle, Washington, August 1999.

[31] eTForecasts: Internet Users vs. Wireless Users (millions).

[32] eTForecasts: PDA. http://www.etforecasts.com/pr/pr0102.htm.

[33] Kevin Fall and Kannan Varadhan. ns: Notes and documentation. Technical report, University of California at Berkeley, LBL, USC/ISI, and Xerox PARC. Technical Report.

[34] W. Fenner. Internet group management protocol, version 2. Internet Draft, Internet Engineering Task Force, November 1997. Work in progress.

[35] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. Reliable multicast framework for light-weight sessions and application level framing. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 784–803, Cambridge, Massachusetts, September 1995.

[36] Lixin Gao, Jim Kurose, and Don Towsley. Efficient schemes for broadcasting popular videos. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pages 317–329, Cambridge, England, July 1998.

[37] J. Gemmell, H.M. Vin, D.D Kandlur, and P.V. Rangan. Multimedia storage servers: a tutorial. In *IEEE Computer*, pages 40–49, May 1995.

[38] Glimpse. http://www.webglimpse.org.

[39] Savo G. Glisic and Pentti A. Leppanen. *Wireless Communications TDMA versus CDMA*. Kluwer Academic Publishers, 1997.

[40] Gnutella. http://gnutella.wego.com.

[41] L. Golubchik, J.C.S. Lui, and R.R. Muntz. Chained declustering: load balancing and robustness to skew and failure. *RIDE-TQP Workshop*, February 1992.

[42] Leana Golubchik, John C. S. Lui, and Richard R. Muntz. Adaptive piggybacking: a novel technique for data sharing in video-on-demand storage servers. *Multimedia Systems*, 4(3):140–155, 1996.

[43] Boris Grndahl. Wireless world meets to lick its wounds. http://www.thestandard.com/article/display/0,1151,22322,00.html?nl=dnt.

[44] Bjorn Gronvall, Assar Westerlund, and Stephen Pink. The design of a multicast-based distributed file system. In *Operating Systems Design and Implementation*, pages 251–264, 1999.

[45] Matthias Grossglauser and David Tse. Mobility increases the capacity of mobile ad-hoc wireless networks. In *IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[46] Shlomo Havlin, Menachem Dishon, James E. Kiefer, and George H. Weiss. Trapping of random walk in two and three dimensions. In *Physical Review Letter*, page 407, 1984.

[47] H. Hsiao and D. J. DeWitt. Chained declustering: a new availability strategy for multiprocessor database machines. In *Proceedings of Data Engineering*, pages 456–465, 1990.

[48] Barry D. Hughes. *Random Walks and Random Environments*. Oxford Science Publications, 1995.

[49] T. Ibaraki and N. Katoh. *Resource Allocation Problems*. The MIT Press, 1988.

[50] Tomasz Imielinski, S. Viswanathan, and B. R. Badrinath. Energy efficient indexing on air. In *International conference on Management of Data, ACM SIGMOD*, Minneapolis, 1994.

[51] Jon Inouye, Jim Binkley, and Jonathan Walpole. Dynamic network reconfiguration support for mobile computers. In ACM, editor, *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 13–22, 1997.

[52] David M. Jacobson and John Wilkes. Disk scheduling algorithms based on rotational position. Technical report, HP Laboratories technical report HPL-CSP-91-7rev1, March 1991.

[53] Van Jacobson. pathchar – a tool to infer characteristics of Internet paths. In *Mathematical Sciences Research Institute (MSRI) Math Awareness Week (Mathematics and the Internet)*, April 1997. software at ftp://ftp.ee.lbl.gov/pathchar/.

[54] S. Jamin, S. J. Shenker, and P. B. Danzig. Comparison of measurement-based admission control algorithms for controlled-load service. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Kobe, Japan, April 1997.

[55] Wenyu Jiang. Detecting and measuring asymmetric links in an IP network. Technical Report CUCS-009-99, Columbia University, New York, New York, January 1999.

[56] James J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. *Proc. ACM Symposium on Operating Systems Principles*, 8(2):213–225, October 1991.

[57] R. Kravets and P. Krishnan. Power management techniques for mobile communication. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 157–168, Dallas, Texas, October 1998.

[58] Joanna Kulik, Wendi Rabiner, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In ACM, editor, *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 24–35, Seattle, Washington, August 1999.

[59] Kevin Lai, Mema Roussopoulos, Diane Tang, Xinhua Zhao, and Mary Baker. Experiences with a mobile testbed. In *Proceedings of the Second International Conference on Worldwide Computing and its Applications*, March 1998.

[60] Hui Lei and Dan Duchamp. An analytical approach to file prefetching. In *USENIX 1997 Annual Technical Conference*, Monterey, CA, January 1997.

[61] The Jakarta Project Lucene. http://jakarta.apache.org/lucene/docs/index.html.

[62] William Mendenhall, Dennis Wackerly, and Richard Scheaffer. *Mathematical Statistics with Applications*. Duxbury Press, 1994.

[63] Metricom. The Ricochet wireless network overview, 1999.

[64] Napster. http://www.napster.com.

[65] Inc NTTDoCoMo. http://www.nttdocomo.co.jp/.

[66] The New York Times on the Web. http://www.nytimes.com/adinfo/wireless_audience.html.

[67] A. Ovchinnikov, S. Timashev, and A. Belyy. *Kinetics of Diffusion controlled chemical processes*. Nova Science Publishers, 1989.

[68] T. W. Page, R. G. Guy, J. S. Heidemann, D. Ratner, P. Reiher, A. Goel, G. H. Kuenning, and G. J. Popek. Perspectives on optimistically replicated peer-to-peer filing. *Software—Practice and Experience*, 28(2):155–180, February 1998.

[69] C. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., 1982.

[70] Maria Papadopouli and Leana Golubchik. Support of VBR video streams under disk bandwidth limitations. *SIGMETRICS Performance Evaluation Review*, 25(3):13–20, December 1997.

[71] Maria Papadopouli and Leana Golubchik. A scalable video on demand server for a dynamic heterogeneous environment. In *Proceedings, Lecture Notes in Computer Science*, volume 1508. Springer, September 1998.

[72] Maria Papadopouli and Henning Schulzrinne. Connection sharing in an ad hoc wireless network among collaborating hosts. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSS-DAV)*, pages 169–185, Basking Ridge, New Jersey, June 1999.

[73] Maria Papadopouli and Henning Schulzrinne. Seven degrees of separation in mobile ad hoc networks. In *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, San Francisco, California, November 2000.

[74] Maria Papadopouli and Henning Schulzrinne. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *Proc. of Mobihoc*, Long Beach, California, October 2001.

[75] Maria Papadopouli and Henning Schulzrinne. Performance of data dissemination and message relaying in mobile ad hoc networks. Technical Report CUCS-004-02, Dept. of Computer Science, Columbia University, New York, New York, February 2002.

[76] Vern Paxson and Sally Floyd. Wide-area traffic: the failure of Poisson modeling. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 257–268, London, United Kingdom, August 1994. ACM.

[77] Roman Pichna, Tero Ojanpera, Harri Posti, and Jouni Karppinen. Wireless internet - IMT-2000/wireless LAN interworking. *Journal of Communications and Networking*, 2(1):46–57, March 2000.

[78] DATAMAN project. http://www.cs.rutgers.edu/dataman/, 2001.

[79] RadioLAN. http://www.radiolan.com/, 1998.

[80] T. S. Rappaport. *Wireless communications, principles and practice.* Prentice Hall, 1996.

[81] Krishnamurthi Ravishankar and Suresh Singh. Broadcasting on [0,L], 1994.

[82] Sidney Resnick. Heavy tail modeling and teletraffic data. Technical Report TR 1134, Cornell University, 1995.

[83] RIM. http://www.goamerica.net/coverage/cingular.html.

[84] Sheldon M. Ross. *Stochastic Processes*. John Wiley and Sons, New York, New York, 1983.

[85] Elizabeth M. Royer and Charles E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In ACM, editor, *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 207–218, Seattle, Washington, August 1999.

[86] Chris Ruemmler and John Wilkes. An introduction to disk drive modelling. *IEEE Computer*, pages 17–28, March 1994.

[87] James D. Salehi, Zhi-Li Zhang, Jim Kurose, and Don Towsley. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. *IEEE/ACM Transactions on Networking*, 6(4):397–410, August 1998.

[88] Subhabrata Sen, Jennifer Rexford, Jayanta Dey, James F. Kurose, and Don Towsley. Online smoothing of variable-bit-rate streaming video. Technical Report 98-75, Department of Computer Science, University of Massachusetts, Amherst, Massachusetts, 1998.

[89] S. Seshan, M. Stemm, and R.H. Katz. Spand: Shared passive network performance discovery. *Proc 1st Usenix Symposium on Internet Technologies and Systems*, December 1997.

[90] Dorgham Sisalem, Henning Schulzrinne, and Christian Sieckmeyer. The network video terminal. In *HPDC Focus Workshop on Multimedia and Collaborative Environments (Fifth IEEE International Symposium on High Performance Distributed Computing)*, Syracuse, New York, August 1996. IEEE Computer Society.

[91] M. Spreitzer, M. Theimer, K. Petersen, A. Demers, and D. Terry. Dealing with server corruption in weakly consistent, replicated data systems. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 234–240, Budapest, Hungary, September 1997.

[92] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications, vol.E80-B, no.8, p. 1125-31*, E80-B(8):1125–31, 1997.

[93] Mark Stemm and Randy H. Katz. Vertical handoffs in wireless overlay networks. *ACM Mobile Networking (MONET)*, 1998. Special Issue on Mobile Networking in the Internet.

[94] Carl Tait, Hui Lei, Swarup Acharya, and Henry Chang. Intelligent file hoarding for mobile computers. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 1995.

[95] R.E. Tarjan. *Data Structures and Network Algorithms*. CBMS-NSF Regional Conference Series in Applied Mathematics, 1983.

[96] D. Taubman and A. Zakhor. A common framework for rate and distortion based scaling of highly scalable compressed video. *IEEE Transaction on Circuits and Systems for Video Technology*, pages 329–354, 1996.

[97] Vindigo. http://www.vindigo.com.

[98] Vindigo. http://www.vindigo.com/learn_more.html.

[99] WaveLAN. WaveLAN library. http://www.wavelan.com/support/library.html, 1998. Lucent Technologies.

[100] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. *ACM Computer Communication Review*, 25(4):100–113, October 1995.

[101] NYC wireless. http://www.nycwireless.net.

[102] Linda Wu, Rosen Sharma, and Brian Smith. Thin streams: An architecture for multicasting layered video. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, St. Louis, Missouri, May 1997.

[103] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad-hoc routing. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, August 2001.

[104] Tao Ye, H.-Arno Jacobsen, and Randy Katz. Mobile awareness in a wide area wireless network of info-stations. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, October 1998.

[105] Xinhua Zhao and Mary Baker. Flexible connectivity management for mobile hosts. Technical report, Stanford University, September 1997.