

DATA DISSEMINATION STRATEGIES FOR MOBILE PEER-TO-PEER INFORMATION SYSTEMS WITH APPLICATIONS TO HEALTHCARE

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Fatih Melih Özbekoğlu

August, 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Özgür Ulusoy (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İbrahim Körpeoğlu (Co-supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Uğur Güdükbay

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Ali Aydın Selçuk

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ezhan Karaşan

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

DATA DISSEMINATION STRATEGIES FOR MOBILE PEER-TO-PEER INFORMATION SYSTEMS WITH APPLICATIONS TO HEALTHCARE

Fatih Melih Özbekoğlu

M.S. in Computer Engineering

Supervisors: Prof. Dr. Özgür Ulusoy and

Asst. Prof. Dr. İbrahim Körpeoğlu

August, 2009

Peer-to-peer (P2P) architecture is becoming increasingly popular for various applications, replacing the classical Client-Server architecture. With the enhanced capabilities of mobile devices (PDAs, mobile phones, etc.) wireless networks started to take advantage of P2P paradigm along with its properties like infrastructure-free operation, scalability, balanced and distributed workload. Mobile peer-to-peer (MP2P) networks refer to the application of P2P architecture over wireless networks. Problems about dissemination of data in both P2P and MP2P networks are widely studied, and there are many proposed solutions.

Healthcare information systems are helping clinicians to hold the information belonging to patients and diseases, and to communicate with each other since early 1950s. Today, they are widely used in hospitals, being constructed using Client-Server network architecture. Wireless technologies are also applied to medical domain, especially for monitoring purposes.

In this thesis, we present and evaluate various data dissemination strategies to work on a mobile peer-to-peer (MP2P) network designed for a medical healthcare environment. First, the designed network system is presented along with the network topology. Then, the proposed data dissemination strategies are described. And finally, these strategies are evaluated according to the properties and needs of a medical system.

Keywords: Peer-to-peer architecture, Mobile Peer-to-peer networks, Data dissemination, Healthcare information systems.

ÖZET

MOBİL EŞLER ARASI BİLGİ SİSTEMLERİNDE VERİ DAĞITIM STRATEJİLERİ VE SAĞLIK ALANINDAKİ UYGULAMALARI

Fatih Melih Özbekoğlu

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Yöneticileri: Prof. Dr. Özgür Ulusoy ve

Yard. Doç. Dr. İbrahim Körpeoğlu

Ağustos, 2009

Eşler Arası (Peer-to-peer) mimarisi, klasik istemci/sunucu mimarisinin yerini alarak çeşitli uygulamalarda giderek daha çok kullanılmaya başlamıştır. Mobil cihazların gelişen özellikleriyle beraber, kablosuz ağlar eşler arası modelini kullanıp, bu ağların altyapı gerektirmeme, ölçeklenebilirlik ve dengeli yük dağılımı gibi özelliklerinden faydalanmaya başlamıştır. Mobil eşler arası (Mobile Peer-to-peer) ağlar terimi, eşler arası mimari ile kablosuz ağların birleşiminden ortaya çıkmıştır.

Sağlık bilgi sistemleri 1950'li yıllardan beri sağlık görevlilerinin hasta ve hastalık bilgilerini tutmasını ve birbirleriyle iletişimini sağlamıştır. Günümüzde istemci/sunucu mimarisiyle geliştirilmiş olarak yaygın bir şekilde hastanelerde kullanılmaktadır. Kablosuz teknolojiler de tıp alanında özellikle hasta takip amacıyla kullanılmaktadır.

Bu tezde, tıbbi bir ortam için tasarlanmış olan bir mobil eşler arası ağ üzerinde çalışan çeşitli veri dağılım stratejileri sunulmakta ve değerlendirilmektedir. İlk olarak, tasarlanan ağ sistemi, ağ topolojisiyle birlikte tanıtılmaktadır. Daha sonra, önerilen veri dağıtım stratejileri tanımlanmaktadır. En son olarak da, bu stratejiler, tıbbi bir sistemin nitelik ve ihtiyaçlarına göre değerlendirilmektedir.

Anahtar sözcükler: Eşler Arası mimarisi, Mobil Eşler Arası ağlar, Veri dağıtımı, Sağlık bilgi sistemleri.

To my family...

Acknowledgement

I would like to thank my supervisors Prof. Dr. Özgür Ulusoy and Assist. Prof. Dr. İbrahim Körpeoğlu for their endless support, guidance and continuous encouragement. It was an honour to work with them.

I would like to thank to the jury members Assoc. Prof. Dr. Uğur Güdükbay, Asst. Prof. Dr. Ali Aydın Selçuk and Assoc. Prof. Dr. Ezhan Kardeşan for kindly accepting to spend their valuable time and evaluating my thesis.

I would like to express my appreciation to the Scientific and Technical Research Council of Turkey (TÜBİTAK) for its support to my master's study.

I want to express my thanks to my father, my mother and my sister for their endless love and support.

Starting with Hilal and Barış, I want to thank all of my friends for making my master's study such enjoyable. I am so grateful for their accompaniment and support.

Contents

- 1 Introduction** **1**

- 2 Background and Related Work** **5**
 - 2.1 Peer-to-peer Networks 5
 - 2.1.1 Centralized Peer-to-peer Networks 6
 - 2.1.2 Decentralized Peer-to-peer Networks 7
 - 2.1.3 Semi-Centralized Peer-to-peer Networks 9
 - 2.2 Mobile Technologies 10
 - 2.3 Mobile Peer-to-peer Networks 11
 - 2.4 Healthcare Information Systems 12

- 3 Data Dissemination Strategies for Medical Mobile Peer-to-Peer System** **14**
 - 3.1 Proactive Data Dissemination 21
 - 3.2 Reactive Data Dissemination 24
 - 3.2.1 Query Caching & Resending 24

<i>CONTENTS</i>	ix
3.2.2 Replicating Patient Data	25
3.3 Classification of Patient Records	27
3.4 Summary	28
4 Evaluation and Simulation Results	36
4.1 Evaluation Criteria and Simulation Environment	36
4.2 Simulation Results	39
4.2.1 Comparison of Client-Server and P2P paradigms	39
4.2.2 Evaluating Methods for Proactive Data Dissemination	42
4.2.3 Evaluating Methods for Reactive Data Dissemination	43
4.3 Summary of Results	57
5 Conclusion and Future Work	62
Bibliography	64

List of Figures

2.1	Centralized P2P Architecture.	7
2.2	Decentralized P2P Architecture.	8
2.3	Semi-Centralized P2P Architecture.	9
3.1	System overview.	16
3.2	Push operation with subscription.	23
3.3	Caching and resending queries	30
3.4	Replicating patient data	33
4.1	The square topology with 9 superpeers each of them having 3 peers.	39
4.2	Difference in average delivery times of messages for push operations with P2P and Client-Server architectures.	40
4.3	Difference in average delivery times with various update periods. .	41
4.4	Comparison of push operation with flooding and push operation with subscriptions in terms of total number of created messages. .	43
4.5	Comparison of push operation by assigning more subscriptions in terms of total number of created messages.	44

4.6 Comparison of push operation by assigning more subscriptions in terms of average delivery time for updates. 45

4.7 Comparison of resend operation for different wait times in terms of QSR. 46

4.8 Comparison of resend operation for different wait-times in terms of Total Number of Messages. 47

4.9 Comparison of resend operation for different wait-times in terms of ADT. 48

4.10 The number of query hits caused by resend operation and other hits for different wait-times. 49

4.11 The ratio of query hits caused by resend operation over total number of hits for different wait times. 50

4.12 Change in Query Success Ratio (QSR) for different values of query threshold when different replication limits are applied. 51

4.13 Change in Average Delivery Time (ADT) for different values of query threshold when different replication limits are applied. 52

4.14 Change in Total Number of Messages for different values of query threshold when different replication limits are applied. 53

4.15 Change in Total Record Storage Volume (TRSV) for different values of query threshold when replication limits are 5 and 10. 54

4.16 Change in Quality Success Ratio (QSR) for different values of replication period when different replication limits are applied. 55

4.17 Change in Average Delivery Time (ADT) for different values of replication period when different replication limits are applied. 56

4.18 Change in Total Number of Messages for different values of replication periods when different replication limits are applied. 57

4.19 Change in Total Record Storage Volume (TRSV) for different values of replication periods when replication limits are 5 and 10. 58

4.20 QSR values for four different categories in a network of 90 peers when classification is applied and not applied. 58

4.21 QSR values for four different categories in a network of 250 peers when classification is applied and not applied. 59

4.22 CQSR values for four different categories in networks of 90, 250 and 640 peers when classification is applied and not applied. 59

4.23 CADT values for four different categories in networks of 90, 250 and 640 peers when classification is applied and not applied. 60

4.24 QSR values for four different categories in a network of 250 peers when classification is applied and not applied. Here, replication methodology is applied. 60

4.25 ADT values for four different categories in a network of 250 peers when classification is applied and not applied. Here, replication methodology is applied. 61

4.26 CADT values for four different categories in a network of 250 peers when classification is applied and not applied. 61

List of Tables

3.1	Information about identity of patient.	17
3.2	Diagnostic information about the patient.	18
3.3	Textual test & examination results.	18
3.4	Graphical or animated results extracted from biomedical imaging or video recording devices.	19

Chapter 1

Introduction

Peer-to-peer (P2P) networks are decentralized, self-organizing, distributed, application-level computer networks. P2P networks take advantage of cumulative bandwidth and connectivity between peers while proposing alternative solutions to the traditional client-server architecture. They are overlay networks implemented on application layer regardless of underlying physical connection layers. With its success in information sharing, P2P networking gained popularity through years especially with file sharing systems like Kazaa, Napster and Gnutella. Not only file sharing applications make use of this paradigm. Applications that perform multimedia streaming, communication [45], or excessive processing [44] are also designed in a P2P manner and the trend is to apply P2P networking in areas other than file sharing systems. There are many reasons behind this popularity which can be listed as follows:

- P2P networks do not require a central infrastructure (server), avoiding the overhead of its administration. This feature also makes these networks more reliable, since there is no single point of failure.
- They balance the workload by spreading computational overhead over peers. For example, the SETI@home Project introduced in [44] makes use of this advantage. Lots of people help the project by donating some of their computers' processing power.

- P2P networks are scalable. That means you can directly add new nodes to the network without making upgrades to the system. Only deploying the client software is sufficient. However, in the client server architecture, upgrades like adding new servers must be made in order to increase capacity.
- In P2P networks, the clients can share their resources, including bandwidth, storage space, and computing power. They are allowed to use all the resources available in the system.
- Database systems deployed in P2P networks can withstand high update rates. It is an advantage in continuous movement of data. Servers can not meet demands for some cases in the client-server architecture.

With the increased capacity of mobile devices (e.g., Personal Digital Assistants-PDAs, mobile phones, laptops, etc.) and underlying wireless network technologies (e.g., IEEE 802.11, Bluetooth, GSM, etc.) it has been possible to adapt P2P paradigm to mobile wireless systems. Mobile P2P (MP2P) networks take advantage of P2P networks in a wireless environment. These networks have some distinct properties and constraints making them different from classical Internet P2P networks:

- MP2P solutions are generally applied on relatively smaller networks compared to Internet P2P applications.
- The network topology is highly dynamic and unpredictable. Nodes can move frequently and independently.
- Links between nodes are fragile and bandwidth is limited compared to wired links.

In the light of these properties, current research in MP2P networks tries to make full use of the network resources by decreasing the message overhead in the network, and finds effective methods.

Application of wireless mobile technologies to Medicine is becoming increasingly popular and there is a significant amount of work in that area. Medical

personnel may require to use mobile devices like PDAs in performing their jobs. In order to respond to such requirements, principles of mobile P2P networking can be evaluated according to needs of medicine, and a P2P model can be constructed for a medical environment like a hospital.

Networks of different domains and different sizes may require unique approaches to handle specific issues of that domain. A medical network is different from the others in terms of network size and types of flowing data. In this thesis, we aim to present various data dissemination strategies for a mobile P2P network in medical domain. Our main contributions are listed as follows:

- Information sharing on healthcare is generally implemented through the client-server model. We investigate what are the benefits of using P2P paradigm in such type of networks and what are the trade-offs to be considered.
- We investigate the possibility of providing fast dissemination of data by making full use of the collection of computational power and use of the bandwidth formed by peers in the network. In our application domain, doctors may need to get information about their patients' data. The underlying network requires to provide data quickly by applying efficient routing strategies which will reduce message overhead and effectively use bandwidth. We present and evaluate a variety of data dissemination schemes aiming to satisfy this requirement.

As it is stated while describing the advantages of P2P networks, databases in P2P systems can withstand high update rates that may be generated by continuous data. Servers have certain limitations and can answer to a certain number of requests. But, since P2P systems do not rely on servers, large number of updates can be handled. For example, a large number of doctors can issue continuous queries at about the same time in the network. This would normally cause a big problem at a central server. But, the system we propose to use can support that kind of requests without posing a threat for the network.

One of the main advantages of P2P networks is scalability. We aim to provide a mobile P2P system that can be extended easily. New peers and peer groups can be integrated to the system easily. For our healthcare domain, the scalability property of P2P networks can be used in interconnecting several hospitals or constructing a bigger medical information system. Such a system may provide a facility that will connect many doctors, clinics, hospitals and pharmacies countrywide. Currently, works on these types of systems are carried on using the classical client-server architecture and this approach brings some significant problems due to its nature like massive load for servers, and denial of service as a consequence.

Healthcare information systems can benefit from the advantages of P2P architecture. The medical systems built using the P2P model can either be complementary for existing systems or they can serve as the main system for various situations. For example, when a disaster like an earthquake happens, a large area (e.g., stadium, race circuit, park) can be facilitated as a hospital and a P2P healthcare information system can be set up easily using mobile devices. Information sharing between clinicians can be provided by such a mobile P2P system without a need for extra infrastructure. Family doctor system can be considered as another application which can benefit from the P2P model. Recently, each family has its own doctor and the family doctor is primarily responsible for the health related issues of that family. As a consequence, medical data about family members are found at family doctors. That distributed information structure can be managed using a P2P network.

Inspired from the reasons mentioned above, in this thesis, the motivation is to show that it is possible and beneficial to apply P2P network architecture in healthcare information systems. The rest of this thesis is organized as follows. In the next chapter, background information about P2P networks, mobile technologies and healthcare information systems are provided, and the related research in these areas is introduced. Chapter 3 presents the proposed system and describes the methods provided for proactive and reactive data dissemination. Evaluation results of these methods are provided in Chapter 4. Finally, the last chapter concludes the thesis.

Chapter 2

Background and Related Work

In this chapter, information about the related technologies for our work is provided by describing the underlying structure and documenting former research. First, peer-to-peer networks are discussed in detail. Then, underlying mobile technologies are introduced. Former efforts on mobile peer-to-peer systems are expressed in Section 2.3. In the last section, we discuss the recent research and related technologies in healthcare information systems.

2.1 Peer-to-peer Networks

Peer-to-peer (P2P) networks emerged as an alternative to networks structured in a client-server manner. The need for extra infrastructures on the server side and single point of failure were some of the leading handicaps of client-server model, and P2P model provided efficient and effective solutions replacing the former model in certain areas. First of all, P2P networks are highly scalable, enabling large numbers of people to be a part of a collaborative environment. Especially file sharing systems like Kazaa, Napster and Gnutella made use of this model and gained great popularity.

We have introduced the advantages of P2P networks in the first chapter. In

this chapter, we focus on how these networks are designed, how they have evolved, and what research on P2P networks is performed. From the first day, different implementations of P2P networks have been attempted in order to resolve the issues like to maintain connectivity and resource sharing between peers. Harjula et al. [14] and Kant et al. [24] worked on how different P2P networks can be classified. In terms of topology formation, these networks can be differentiated as structured and unstructured. In unstructured P2P networks, there is no control over the network topology, whereas in structured networks peers are placed in some kind of regulation. Locating peers and resources is the main problem in P2P networks, and three different architectures exist for this problem. In chronological order, the first one is the centralized (also called hybrid) architecture, the second one is the decentralized (also called pure) architecture, and the last one is the mixture of these two called semi-centralized architecture. In our work we use the third approach and therefore, we discuss why we decided to use this architecture by introducing properties of all three architectures. We provide detailed explanation about these approaches in the following subsections.

Peer-to-peer networks are well studied throughout years. Especially, there are many efforts on search methods and lookup services. Tsoumakos and Rousopoulos compared some of the most popular methods in [49]. Yang and Garcia-Molina presented three new techniques and compared them to existing ones in [53]. Garcia-Molina also introduced the concept of routing indices with Crespo in [8]. Flooding is a commonly used lookup method in P2P networks. Jiang et al. came up with an improvement in this method which they call LightFlood in [22]. Doulkeridis et al. [10] explored the role and benefits of context-awareness and caching query results in a P2P context.

2.1.1 Centralized Peer-to-peer Networks

Centralized architecture also called hybrid because although peers create peer-to-peer connections between each other, locating the resources and finding the addresses of other peers is performed in a client-server fashion. A central server holds the index of peers and files. Searches are made through that index server.

Figure 2.1 illustrates how this structure works in a file sharing system. In this figure, peer A searches for a file that is present at peer C. In order to locate the file, it sends a query to the central server (CS) and gets a response containing the address of peer C (white arrow). Then A sets up a direct connection to C using the information gathered from the server (dark arrow). The process works the same when peers H and E search for the files which are present at peers E and D, respectively.

The centralized approach to P2P networks has some advantages like fast and efficient searching. But, it also has a lot of disadvantages. Although, the task of file transfer is taken from the server, it still has an indexing mission and that brings a load to the server causing it to become a bottleneck in terms of network traffic. Index server is a single point of failure vulnerable to attacks and also it is hard to achieve scalability.

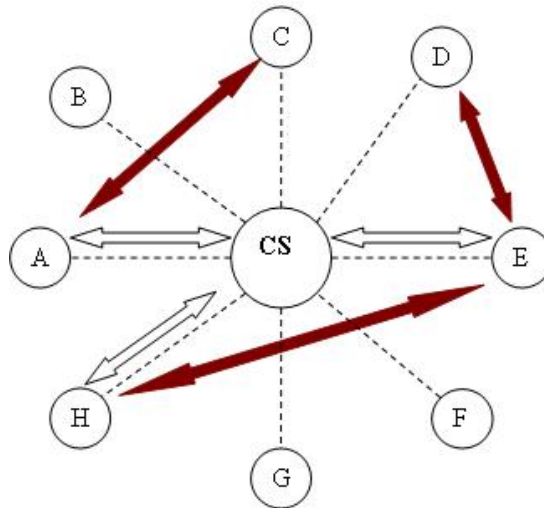


Figure 2.1: Centralized P2P Architecture.

2.1.2 Decentralized Peer-to-peer Networks

In decentralized peer-to-peer architecture, there is no central infrastructure to hold indices of content or addresses. Locating resources or finding other peers

is achieved through the collaboration of peers individually which are unaware of each other. Generally queries are flooded in the network to find the desired resources. Gnutella is a typical example which employs a decentralized approach to network design [49].

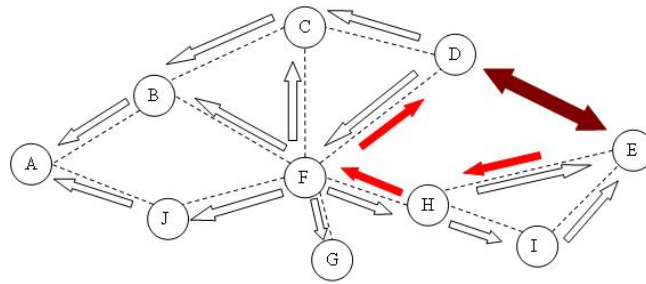


Figure 2.2: Decentralized P2P Architecture.

Figure 2.2 shows an example of how this structure works in a file sharing system. In this figure, peer D searches for a file that is present at peer E. In order to locate the file, it sends a query to its neighbors (C and F) and they search that query in themselves. Then, they forward that query to their own neighbors (white arrows). This goes on until the query reaches to peer E which sends a response to D for the query including its address (dark arrows). Then D sets up a direct connection to E using the information gathered from the response (two sided arrow).

Decentralized P2P structure eliminates some deficiencies of the centralized structure like single point of failure, performance bottleneck and unscalability. However, due to the nature of decentralized organization, some disadvantages may arise. The most noticeable one is the network overhead. Flooding queries throughout the network causes a big increase in the network traffic and consumes a lot of bandwidth. Also, it takes longer to search objects compared to the centralized approach.

2.1.3 Semi-Centralized Peer-to-peer Networks

Semi-centralized peer-to-peer network structure is a combination of both centralized and decentralized approaches. It aims to make use of the advantages of both techniques and achieves that to some extent. That structure has a hierarchical formation of network elements. There is a backbone of elements called superpeers which acts like a decentralized P2P network. Each superpeer has peers connected to itself and they all constitute a sub-network. The superpeer acts as an index server of a centralized P2P network.

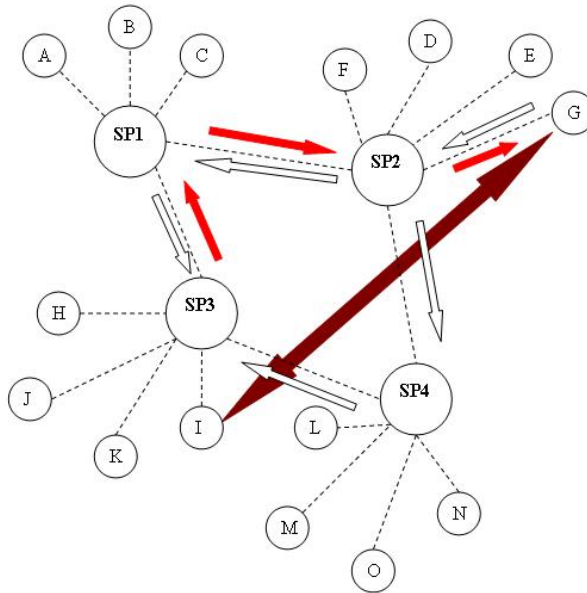


Figure 2.3: Semi-Centralized P2P Architecture.

Figure 2.3 illustrates a working example of this structure in a file sharing system. In this figure, peer G searches for a file that is present at peer I. In order to locate the file, it sends a query to its superpeer (SP2). SP2 holds an index of contents within its peers. Therefore, it looks for that file in the index it holds. Then, SP2 forwards that query to its neighbor superpeers SP1 and SP4 (white arrows). This goes on until the query reaches to the superpeer SP3 which sends a response to G for the query including the address of its peer I (dark arrows). Then G sets up a direct connection to I using the information gathered from the

response (two sided arrow).

As it is mentioned before, semi-centralized P2P networks try to bring together the advantages of centralized and decentralized P2P networks. It has the efficiency of centralized approach and load balancing of decentralized approach [4]. This architecture captures the advantages of both, but, of course, not as the same extent of the former ones. It also eliminates most of the disadvantages of them. Two different examples of this hybrid structure are given in [4] and [11] with various design details.

2.2 Mobile Technologies

There are numerous wireless communication technologies like GSM, Bluetooth [5], IEEE 802.11 (WirelessLAN) [18], IEEE 802.16 (WirelessMAN) [20] and IEEE 802.15 (WirelessPAN) [19] that enable interconnection of devices in a mobile and wireless fashion. While technologies like GSM, and WirelessMAN cover kilometer squares of areas, others (Bluetooth, WirelessLAN and WirelessPAN) serve locally with smaller ranges measured as meters.

For the proposed network architecture of this research, WirelessLAN (also called WiFi) technology which is standardized by IEEE 802.11 working groups can be considered as the most suitable one. We will explain the details of our system in the following chapters, but at least we can say for now that a collection of access points can construct an indoor network environment for a medical service like a hospital. WirelessLAN has different versions of standards which have evolved in time. IEEE802.11b, IEEE802.11g and IEEE802.11n are the most significant ones, and IEEE802.11n has not yet fully been specified. It is currently a draft standard, but many manufacturers have already started to sell products supporting that standard. IEEE802.11g is currently the most used WiFi technology. It provides a bandwidth of maximum 54 Mbit/s with a range of approximately 38 meters. This protocol operates at 2.4 GHz frequency band. Therefore, it sometimes may interfere with cordless phones or microwave ovens, but in general it is a quite

stable and commonly used technology. Today, we can barely find mobile devices (even mobile phones) without WiFi access.

2.3 Mobile Peer-to-peer Networks

The technologies described in the previous sections made it possible to apply peer-to-peer paradigm in mobile environments. Mobile ad hoc networks (MANETs) can be facilitated to work like wired, Internet P2P networks. However, the devices that P2P system works on are mobile and they have limitations like low power, limited memory and limited connectivity. Therefore, the solutions presented for classical P2P networks are not fully applicable to mobile peer-to-peer networks. Research on mobile P2P networks not only resulted in modification of some methods of classical P2P but also led the way to some novel approaches.

The challenges of applying P2P paradigm in mobile environments are discussed by Kellerer et al. along with the requirements and solutions in [26]. Ahmed and Shirmohammadi [1] addressed several design issues and presented a guideline to design of mobile peer-to-peer systems. Mondal et al. also presented a summary of design issues in mobile P2P networks and a way of handling these issues using economic models in [34]. In the light of those guidelines, systems like [16], [41] and [27] have been presented. JXTA is one of the most important technologies that allowed researchers to develop such P2P systems [29].

Improvements on mobile P2P networks continued with the researchers focusing on different aspects. Wolfson et al. [52] proposed a rank-based dissemination algorithm to overcome energy, bandwidth and storage constraints. In [12], Tornado coding is proposed as a solution to data dissemination. Peng et al. presented dynamic indices in [40], and Ahmed et al. took advantage of multi-level hashing in [2]. Repantis and Kalogeraki [43] used Bloom filters in order to achieve content-driven routing, where Joseph et al. [23] introduced a scheme to support scalable data retrieval in large-scale MP2P networks.

Proactive dissemination of data and data replication issues were also studied

by the researchers. Ratner et al. [42] addressed the replication requirements in mobile environments and described a system to meet those requirements. Mondal et al. [32] proposed a dynamic replication scheme (CADRE) for improving low data availability. They also came up with an economic model for efficient dynamic replication in [33]. Maintaining the consistency of data is an important issue in replication schemes. [48] and [35] proposed two different mechanisms to overcome this important issue.

Proactive dissemination of data is one of the important issues that we focus on in this thesis. Although, most of the research on that topic are for classical P2P networks, they are also helpful for mobile P2P networks. In [7], Chirita et al. proposed a publish/subscribe system. Also, a publish/subscribe mechanism is used in [38] for a P2P system. Kassinen et al. presented a group based content push scheme with an intelligent mobile middleware for mobile P2P networks in [25].

2.4 Healthcare Information Systems

Since early 1950s, information technologies have been widely used for medical purposes. Especially in the last two decades, medical world increasingly took advantage of information technologies from hospital information systems to digital libraries. The term “health informatics” is used for the discipline resulted as the mixture of information science, computer science, and health care. Organizations like IMIA [21] and HIMSS [15] provide expertise and leadership for development in health informatics.

Healthcare information systems are used in almost every hospital nowadays. These systems provide storage and processing of administrative and medical information using a classical client-server model. Using P2P paradigm and wireless technologies (wireless telemedicine systems) is not that much popular in healthcare information systems. Hereby, we list some efforts making use of these technologies. Pattichis et al. [39] investigated existing applications of wireless

telemedicine systems. In [37], Ng et al. highlighted current uses and future trends of various wireless communications in the healthcare domain. Cypher et al. [9] addressed the benefits and challenges occurred as a result of operating wireless communication in healthcare networks. Using wireless technologies for patient monitoring is one of the most popular trends in healthcare information systems. Varshney [50] showed how patient monitoring can be achieved using infrastructure-oriented wirelessLANs. He also addressed reliability and power management issues in using ad hoc networks on patient monitoring with Sneha in [51]. Lin et al. presented a wireless PDA-based physiological monitoring system in [28], whereas Milenkovic, Otto and Jovanov listed the issues in wireless personal health monitoring and proposed an architecture in [31]. Al-Leddawi and Kunwar [36] stated security and privacy threats, vulnerabilities in mobile information systems. The WARD-IN-HAND project [3] presents a scheme where existing information systems can be accessed with a wireless connection through PDAs. Clinical Grade [13] also is an example of healthcare communication networks.

In this thesis, we want to find solutions, taking the needs of a medical environment into account and take the advantage of diversity in medical information. Some research on those issues helped us in achieving our goal. Brassey et al. [6] presented the results of a questionnaire in order to measure the habits of clinicians using a healthcare information system. Smith listed what kind of information doctors use [46]. In [17], Hudson and Cohen analyzed different data types in medical records.

As it can be seen above, there are many solutions designed for P2P networks, mobile P2P networks and healthcare information systems. In this thesis, we propose a design for a mobile P2P network in medical domain by taking advantage of all the stated efforts in those different research areas.

Chapter 3

Data Dissemination Strategies for Medical Mobile Peer-to-Peer System

In this thesis, we aim to present and evaluate various data dissemination strategies in order to build an effective mobile peer-to-peer (P2P) network which is intended to be used in a medical healthcare system. We perform evaluations on these methods considering the needs of medical information systems. The system we assume in our work is intended to model the data types belonging to medical domain.

The system used in the evaluation of various techniques can be considered as a modified realization of hybrid peer-to-peer (P2P) networks on medical domain. Due to the needs and nature of that domain, there exists some constraints affecting the design of the system as follows:

- Doctors in the same department tend to be physically together.
- More than one doctor of a department may take care of a patient. Information about that patient is shared among the doctors of that department. As a result of that, information exchange inside a department happens more

frequently than information exchange between departments.

- If some information is present in the system, it must be accessible by the authorized personnel most of the time. Success of the queries about medical records must be provided at the highest level possible.
- Quick access to data is very important. If a doctor needs data about a patient (perhaps in case of emergency) then data belonging to that patient must be provided as soon as possible.
- A hospital is a very dynamic environment in terms of information. Every second, some new data about a patient may be collected or a new habit can be observed. These changes must be reflected to the system as soon as possible to provide the most up-to-date data to users.

In the light of the constraints stated above, we propose a specific version of hybrid P2P networks for such an environment. In our system, the whole network consists of n interconnected peer groups. Since doctors in the same department are more likely to communicate between each other and exchange patient information, peer groups are selected as the departments in the hospital. Each peer group owns a stationary superpeer and this superpeer has wireless links to every peer in that group. Connection between super peers are always maintained and peers use super peers as gateways. Peers are the mobile devices held by doctors in the hospital. The super peers have more computational power and energy than the peers, so that most of the complex work like caching and routing messages are performed by them. Electronic Health Records (EHRs) are held by peers and records belonging to all of the patients are available throughout the whole network. Figure 3.1 shows an example of our network with four departments each having two peers.

In the proposed system, information about the patients are stored in data structures, what we call patient records. Patient record is an electronic health record (EHR) which includes different types of information for a specific patient. A patient record can contain:

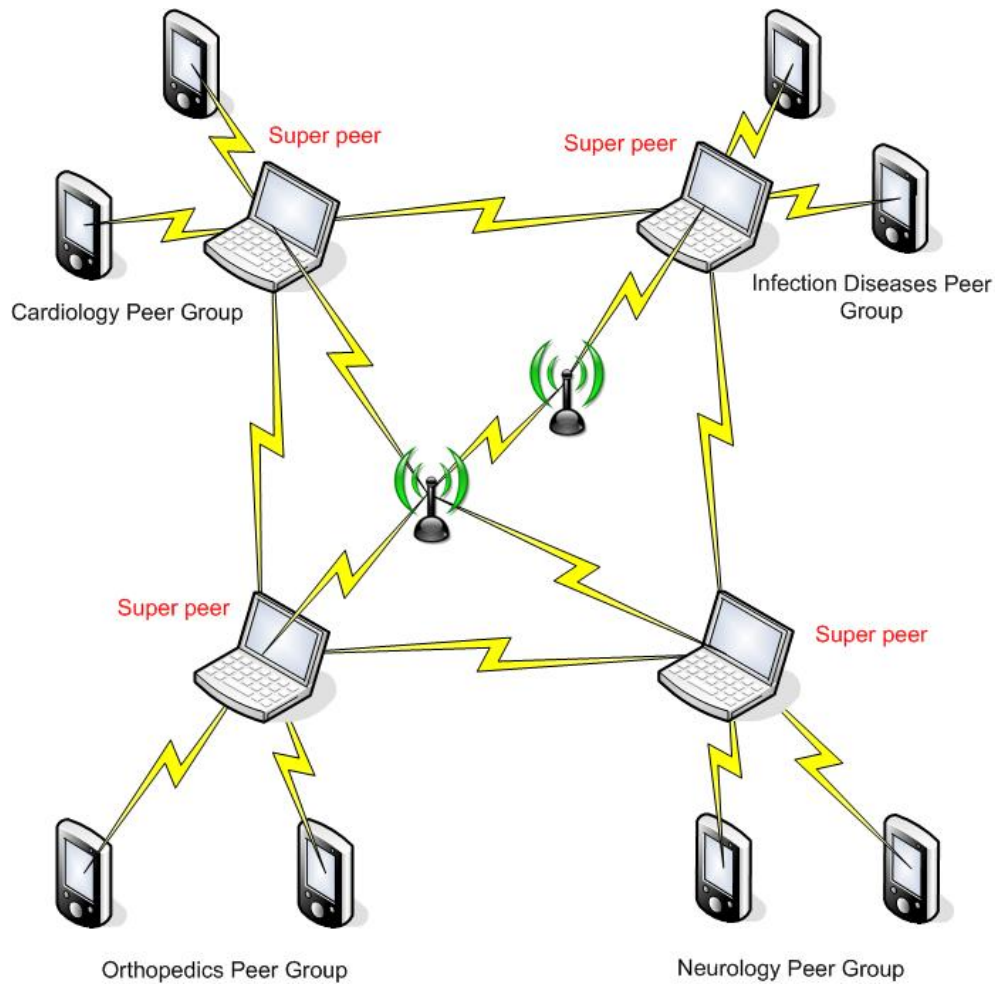


Figure 3.1: System overview.

- Information defining who the patient is. Identity information about a specific patient is maintained as shown in Table 3.1.
- Diagnostic information providing details about the current or past situation of data. As shown in Table 3.2, it may include past diseases, allergies, current diagnosis, recommendations, etc.
- Textual test examination results which are easier to maintain and deliver. They are generally numeric files stored as texts. Since file sizes are small, they are easier to deliver. They can include blood pressure, sugar level, heart rate, etc. Table 3.3 depicts some of such information.

- Graphical or animated results extracted from biomedical imaging or video recording devices. They are bigger and harder to handle in terms of storage and transfer speed. MRI and X-Ray images, ultrasound results, camera records belonging to endoscopy operations can be listed as examples of that type (see Table 3.4).

Different types of information belonging to the same patient record may be located at different superpeers. For example, MRI image of a patient may be stored in the radiology department whereas ECG data of the patient may be stored in the cardiology department. Patient record is a composite structure made up of different types of data. The types described here are not definitely the parts of a record. These are given to picture the general ingredients of a patient record. Division and categorization of parts of patient records are discussed in the section “Categorization of patient records”.

Table 3.1: Information about identity of patient.

ID
Name
Age
Adress
Nationality
Social security status
Occupation
...

Designing a mobile peer-to-peer medical healthcare system brings a set of problems with itself. For example, in classical P2P systems, exchange of large number of messages in the network can cause a high volume of traffic and decrease efficient use of bandwidth. However, in mobile P2P networks, in addition to massive traffic, another considerable overhead is the energy spent by mobile agents. Also there are some medical issues related to the constraints stated above. This arises a requirement for a different approach to P2P networking problems. Although some of these issues are considered as classical P2P network problems, our special design and needs forces to a new consideration. As it was stated in

Table 3.2: Diagnostic information about the patient.

Diagnosis
Allergies
Recommendations
Cross & match (Blood type)
Past diseases
Genetic diseases
Past cures and treatments
...

Table 3.3: Textual test & examination results.

Full blood count
ECG values (heart rate)
Blood pressure
Blood sugar level
Fever (Temperature)

Chapter 2, there are different solutions proposed for the networking problems in both peer-to-peer(P2P) and mobile peer-to-peer(MP2P) networks. But using these paradigms in medical healthcare systems is another issue. Such a system brings its own requirements like highest query success, fast delivery of information, handling special types of data, etc. In our research, we aim to consider those aspects and propose appropriate methods to be used in that specific domain. We attack the problems in three different dimensions as described below:

- The first issue we deal with is the proactive dissemination of data inside the network. That involves the push operation in which data of the patient is measured (collected) and sent to nodes in the network by a peer called the source node. Update messages about a patient record are created and sent to relevant or all of the peers in the network. Generally, this operation generates a huge number of messages causing too much traffic in the network.

Table 3.4: Graphical or animated results extracted from biomedical imaging or video recording devices.

X-Ray image
MRI image
Camera records
Ultrasound results
...

For this problem, we propose a subscription based strategy and compare it to the one where updates are sent to all peers in the network. The aim here is to reduce the overhead cost created by messages flooded to all nodes.

- The second issue is the reactive dissemination which is performed on demand by peers upon incoming queries or requests. Data carrying patient information is delivered as response to queries issued by the peers in the network. For this type of data dissemination, queries are propagated inside the network and if a peer satisfies the conditions in the query, then connections between requesting and replying peers are established. Reducing the overhead cost created by the propagated query messages is again an important issue as well as reducing the response time for queries and reducing the number of update messages used for maintenance of indices if there are any. But, success of queries is more important than any of these concerns. As we mentioned before, accessibility of data is very important in a medical environment. There are several methods in order to overcome these issues by making use of indices, routing algorithms, etc., but we focus on two different approaches which are listed below:

- Caching and resending queries is the first method that we deal with. In a mobile environment, peers can frequently be unavailable and data they possess become unreachable when they are gone. If a query, which can normally be answered by an unavailable peer, is issued when that peer is connected, the query will not be successful. But, such peers may reconnect after a short period of disconnection. In that case, reissuing the same query will provide the temporarily unavailable

peers to answer that query. Instead of reissuing the same query by the user, we propose a mechanism to cache the queries on superpeers for a while and resend them to the regular peers in the same group. Details of that mechanism are explained in the following section.

- Replicating popular records is another approach that is intended to bring some advantages in our case. The reasoning behind this approach is that a record that is queried frequently is popular and the probability that it will be queried in the future is high. In the light of that reasoning, we propose a mechanism where patient records are replicated to some extent, if they are queried for a certain number (query threshold value) of times (i.e., their popularities reach a certain level). The aim here is to increase the possibility of query success and reduce the turnaround time (delivery time) for the issued query. When a frequently queried record is replicated, its abundance throughout the network increases. Therefore, even some peers containing pieces of that record go offline, data will still be available. We provide further explanation about this method in Section 3.2.2.
- As it was declared before, patient records may contain different types of information. Each type has different characteristics. Some type of information change frequently, some of them do not. Or, some data should be accessed as fast as possible where some can wait for a while. If all these types are treated equally, the chance of focusing on needs of different types is missed. Satisfying the needs of queries accessing special data becomes impossible. Therefore, classification of data belonging to patient records is performed and the performance impact of this classification is evaluated in this thesis with the aim of taking the advantage of diversity in data characteristics. Information carried on patient records is classified and assigned different privileges on query processing and replication processes. Detailed information about this approach is provided in Section 3.3.

In our work, we try to come up with effective solutions for the problems listed above. During this process there are some constraints which must be taken into

account like the issues related to mobility (churn, dynamic topology, etc.), and the nature of Electronic Health Records (EHRs) and medical information systems. In the following sections, the details of our approaches are explained.

3.1 Proactive Data Dissemination

Push operation leads to the proactive dissemination of patient data. That means peers in the network do not have to issue queries to retrieve data. Data are directly delivered to those peers. In a hospital environment, records that contain information about patients are changed frequently with new examinations, tests, etc. Therefore, those changes should be delivered to medical personnel after each change. In our system, update messages are responsible for carrying newly updated data. Update messages about a patient record are created and sent to relevant or all of the nodes in the network.

There are two different ways to deliver update messages to relevant peers in the network. One of them is to allow update messages to reach all of the nodes they can. This guarantees that the update will reach peers which possess that record. The other delivery method is to maintain a list for each patient record what we call a “subscriber list” and send the update messages to the peers in that list.

Implementation of push operation through flooding involves direct relaying of update messages to neighbor superpeers and peers. In this method, a peer either creates or receives an update message as an end user. The method is described in Algorithm 1. When a superpeer receives an update message, it first sends this message to all of its peers and then it delivers this message to all of its neighbor superpeers. The algorithm implemented at superpeers is presented in Algorithm 2.

In our design, each part of a patient record contains a “subscribers” field in which the peers and peer groups that are interested in that patient are listed. Then, the peer groups are subscribed to related patients data. A patient record

Algorithm 1 Push operation with flooding - Peers

```

1: if an update occurs then
2:   create the update message
3:   send the update message to own superpeer
4: end if
5: if an update is received then
6:   for each patient record  $p$  held do
7:     if update is about record  $p$  then
8:       save the update
9:     else
10:      drop the update
11:    end if
12:  end for
13: end if

```

Algorithm 2 Push operation with flooding - Superpeers

```

1: an update message is received
2: for each peer belonging to that peer group do
3:   send update message to that peer
4: end for
5: for each neighbor superpeer do
6:   send update message to that superpeer
7: end for

```

can have more than one subscriber. Also, doctors can individually subscribe to a patient's data as a peer. In implementing the push operation, updates about a patient are delivered to the subscriber departments and peers. Data of the patient are collected and sent to subscribers by a peer (doctor) which is called the source node in the network. While sending data, the source node looks up the subscriber list of the patient record and sends the message containing patient data to the peers and superpeers in the list in a unicast fashion. The important part of this method is the maintenance of subscriber lists held on peers. When a doctor wants to subscribe to a patient's data, it first queries using the reactive dissemination methods (which will be introduced in section 3.2) and gets that record. Then updates the record by adding himself on the subscriber list. Then this update is delivered to all peers in the subscriber list. This process is illustrated in Figure 3.2. When the peers in the subscriber list receive that update, they change the record they possess. Whenever an update occurs, which could be a subscriber change

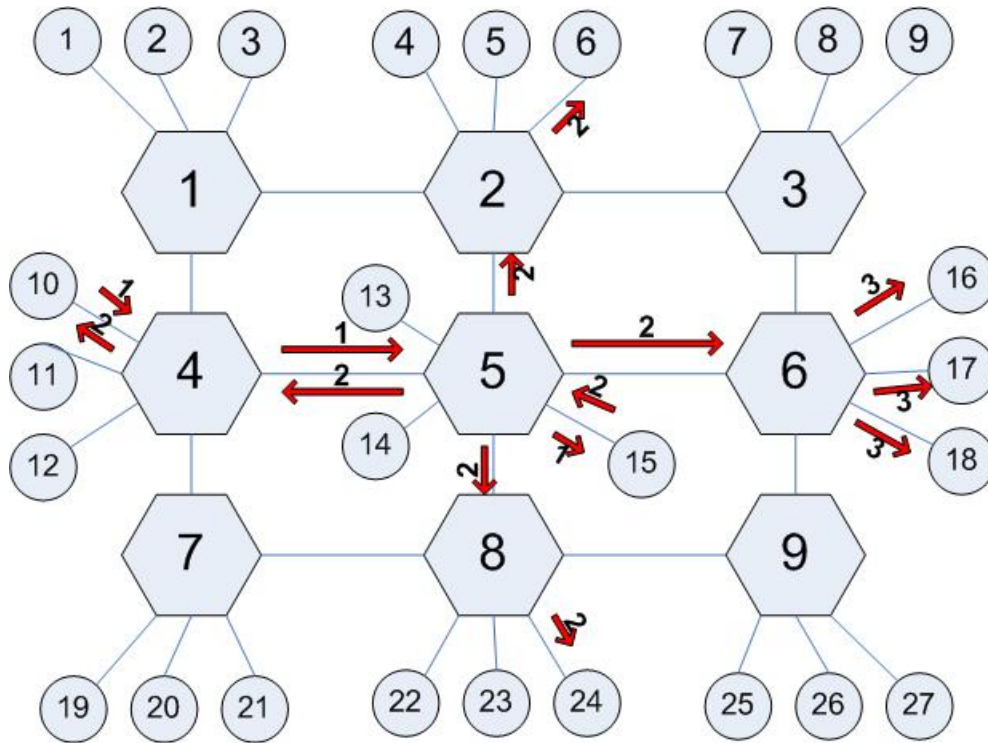


Figure 3.2: Push operation with subscription. (1) Peer 10 wants to subscribe to a patient record stored in peer 15. So, it finds that peer and sends a message to update the subscribers field in the record. (2) This update is delivered to subscriber peers 6, 10, 24 and peer group 6. (3) Superpeer of peer group 6 delivers that update message to its peers 16, 17 and 18.

or a medical information change, an update message is issued to subscribers. The algorithms implemented at superpeers and peers are shown in Algorithms 3 and 4, respectively. Maintenance of subscriber lists seems to be the drawback of this method, but the test results we observed show that the impact of this issue is not quite significant. This subscriber based routing reduces the message overhead and the average delivery time by eliminating uncontrolled flooding of update messages. We present the performance results of this method in Chapter 4.

3.2 Reactive Data Dissemination

Reactive data dissemination is the delivery of information throughout the network upon requests made by peers. Those requests, generally named as queries, search for data, and information is delivered to the sources of requests as responses. In a P2P environment all nodes in the network can both issue queries and supply data. For our medical environment, medical personnel (doctors) can search for information of a specific patient, or they can collect statistical data from patient records for their research. The requested data is supplied by other members of the network which can perform the same operations as others. How queries are directed and how contents of records are indexed are well-studied problems and there are several methods used for them. In our research we try to concentrate on different aspects of query processing other than maintaining indices, hash tables, routing tables, etc. Due to the nature of mobile P2P networks we primarily aim to increase abundance of data (increase success of queries) by taking advantage of long running queries and creating more copies of records. Our proposed solutions are based on the following two methods:

- query caching & resending
- replicating patient data

3.2.1 Query Caching & Resending

Due to the nature of mobile devices and connections between them, peers in a mobile wireless network often experience disconnections. Those disconnections may cause a decrease in the amount of available data at certain times. When a peer is offline, queries made for records which are located at that peer will not be successful. However, those disconnections do not have to last for a long time. Many of them occur frequently but for a small duration. Therefore, a methodology which eliminates this disadvantage can increase the success of queries. Out of two different methods presented in this work, caching and resending queries is the first and the most significant one. Storing queries for offline peers on superpeers

may tolerate disconnections and help peers provide data when they reconnect to the network.

The system basically works as follows: when a superpeer receives a query, it looks up the regular peers in its group and forwards the query to them. If there are any offline peers, it stores the query message in its cache. Then it periodically tries to send this message to the offline peers for a certain amount of time. If a disconnected peer becomes online during that time, it can receive the query and respond to that. The amount of time to resend queries is called *wait-time* and it should be determined carefully. If it is too long, too many messages will be issued. If it is too small, it will expire before the reconnection of peers. Determining an appropriate value is important. This value can change according to the design and characteristic of the network. Also, the period that superpeer resends queries in its cache is important for the performance of this method. Therefore, in some performance experiments, we tried different *wait-time* and *resend period* values. Examples describing our query caching and resending method are presented in Figure 3.3, and the algorithms used for the implementation of this method for peers and superpeers are described in Algorithms 5 and 6, respectively.

The primary goal for implementing this method is to increase the query success ratio in the medical environment. We aim to make sure that doctors reach patient information with the highest possibility when they need. In order to see if it works or not, some performance experiments have been implemented. Results of these experiments are shown and discussed in Chapter 4.

3.2.2 Replicating Patient Data

The second method to provide an improvement in query performance in the presence of disconnections is to replicate patient data which are located on peers. Creating more copies in an environment is likely to decrease the risk to miss a data when its source node is offline, since there will be more providers in the network. But this doesn't mean that we should replicate all the data in the network as much as we can. High levels of replication can lead to some other

problems. Therefore, the degree of replication should be determined carefully.

Our approach for data replication works as follows: when a peer receives a query for a data record it possesses, it responds to the query and increments the counter which keeps track of how many times that record has been requested recently (in the last 100 seconds for our algorithm). Peers periodically check their counters and replicate a record if its counter is above some threshold value. For each execution of replication process, peers create and send 4 copies of that data record to peers in their own peer group and 3 more copies to peers which reside in other peer groups. These peers are determined in a randomized fashion. For this thesis, the number of copies created in each iteration is decided empirically. But, further research can help to determine optimized values for this parameter. An important parameter used for this method is the maximum number of replicas that can be created for a particular record. Also, another parameter to consider is the *query threshold* value that defines when a peer will start to replicate a data record. If this value is too low, then there will be too much replication. Otherwise (if the value is too high), there may not be a significant increase in query success since there will not be enough replicas. In order to guide how the values of these parameters should be selected, we performed tests with different *query threshold* and replication limit values. An example to show how the system works is depicted in Figure 3.4, and the algorithms implemented at peers and superpeers are described in Algorithms 7 and 8, respectively.

The goal for implementing this method is to further improve the performance of the first method (query caching & resending) in a medical environment described in previous sections. We aim to make sure that doctors reach patient information with the highest possibility when they need. In order to see if it works or not, some performance tests have been implemented. Results of those tests are presented and discussed in Chapter 4.

3.3 Classification of Patient Records

Patient records stored in a medical information system can be divided into sub-parts according to their usage. Each part can possess different types of information. This modularity allows software systems to distribute contents and treat them differently in arranging network traffic and regulating allocation of resources throughout the network. Hudson and Cohen [17] show an example of structuring medical information and classifies the information in several ways. Smith [46] describes which type of information the doctors need during the decision making process. According to different needs of doctors, patient data are classified and parts of a patient record can be determined. This allows us to control replication of records and manage update operation of those replicas. According to the importance and expiration time of different kinds of data, we can make some of the data more abundant and more up-to-date.

In the light of the types described in [46] the following classification is provided. This classification is made according to the frequency of changes occurred for different patient data types:

- The first data type is the identity information of patients. This information is generally administrative and includes some attributes like age, place they live, etc. This type of information changes very rarely.
- The second data type is the medical history of a patient. This includes diseases, cures, diagnoses, allergies, genetic background, etc. This type of data is important for doctors to make the right decision during diagnosis or cure. These attributes change more frequently than the first type of data.
- The third data type is the analysis and laboratory results. This type includes one-on-one examinations, MRI images, X-Rays, ECG results, etc. This type of data tends to change frequently and plays very important role in daily activities performed in a hospital. Doctors really require this type of data a few times a day.
- The last data type is the temporary medical situation of the patient. This

type of data includes the most recent information received. Generally, these data are collected from the patients who are in intensive care and the information belonging to a particular patient changes very rapidly.

Data in patient records are classified into four categories as described above. The important question here is what benefits we are going to obtain if we perform this classification. In our model, we are trying to increase the success of queries through the methods of caching & resending queries, and by replicating data records. Different wait-times can be assigned for queries issued in order to retrieve records of different types. With that approach, we can categorize queries and cache them for different periods of time according to their types. Queries for more critical data can be cached longer. As we have mentioned before, replication of data should be controlled to prevent overloading of messages and storage, and to reduce the complexity of managing update operations. With the classified patient data, we can reduce the message traffic by limiting the replication of data which are updated frequently. For the data types that are important updated quite rarely, we can have more replication.

3.4 Summary

In this chapter, the data dissemination strategies and the network structure on which these strategies are used have been presented. An overview of the system was provided along with the patient records used. Then, the proactive data dissemination strategy (i.e., subscription method) was presented. Reactive data dissemination strategies (query caching & resending and replicating patient data) were explained in detail. These two strategies aim to improve the success ratio of queries while reducing the time to get replies. Finally, a classification of patient records was presented. In this method, patient records are classified into four categories for assigning different privileges to data records. In the next chapter, we provide the evaluation results of the proposed dissemination strategies.

Algorithm 3 Push operation with subscription - Peers

```

1: if an update occurs then
2:   create the update message
3:   for each subscriber peer group  $pg$  in the subscriber list do
4:     send the update message to superpeer of  $pg$ 
5:   end for
6:   for each subscriber peer  $p$  in the subscriber list do
7:     send the update message to  $p$ 
8:   end for
9: end if
10: if user wants to subscribe to a patient's record then
11:   query the record with the reactive data dissemination methods
12:   if queried record is found then
13:     create a subscription update
14:   end if
15: end if
16: if a subscription update occurs then
17:   create the update message
18:   for each subscriber peer group  $pg$  in the subscriber list do
19:     send the update message to superpeer of  $pg$ 
20:   end for
21:   for each subscriber peer  $p$  in the subscriber list do
22:     send the update message to  $p$ 
23:   end for
24: end if
25: if an update is received (data update or subscription update) then
26:   if peer owns a record for that update then
27:     save the update
28:   end if
29: end if
30: if peer moves to another peer group then
31:   send the old superpeer a message indicating the new superpeer
32: end if

```

Algorithm 4 Push operation with subscription - Superpeers

```

1: an update message is received
2: for each peer  $p$  belonging to that peer group do
3:   if  $p$  has moved to another group then
4:     send update to new superpeer of  $p$ 
5:   else
6:     send update message to  $p$ 
7:   end if
8: end for

```

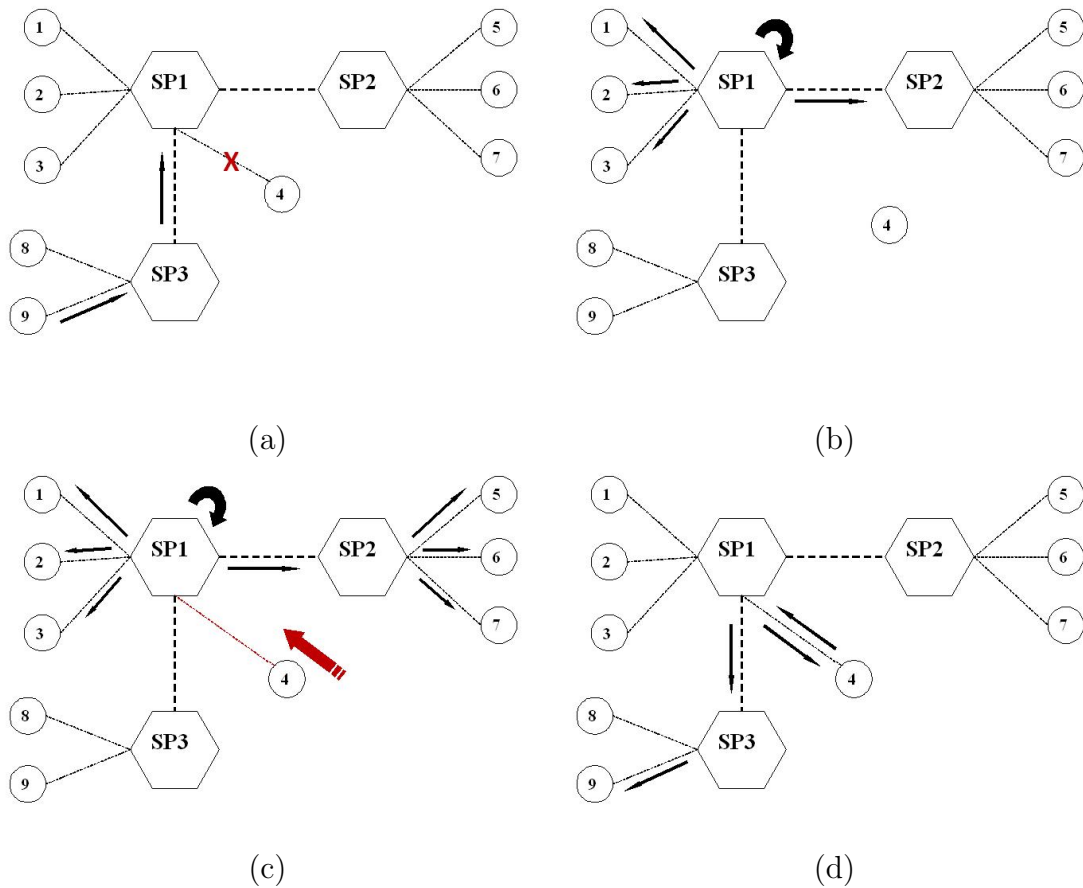


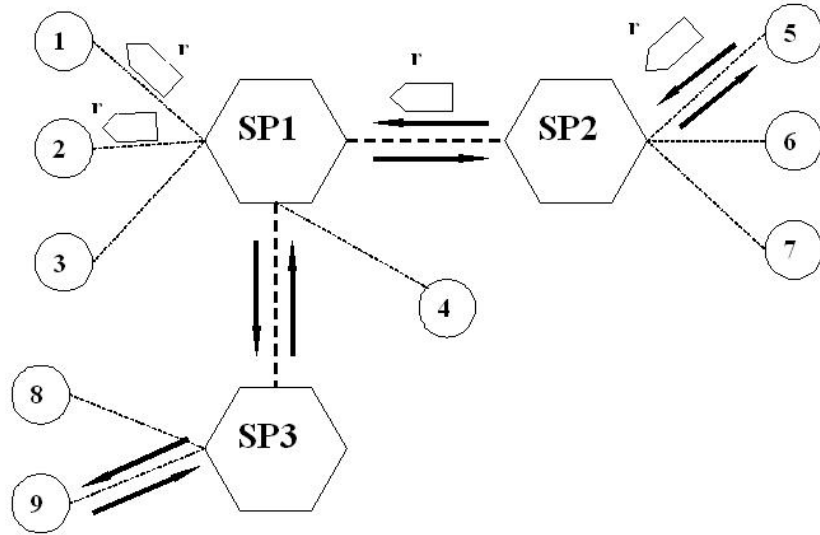
Figure 3.3: Caching and resending queries (a) Peer 4 goes offline as peer 9 issues a query which searches for a data at peer 4 (b) Superpeer 1 receives query, sends it to its peers and caches as it realizes peer 4 is offline (c) Peer 4 goes online as superpeer 1 periodically sends the cached query which searches for the data at peer 4 (d) Peer 4 receives query, sends the response and data transfer process begins.

Algorithm 5 Query caching & resending - Peers

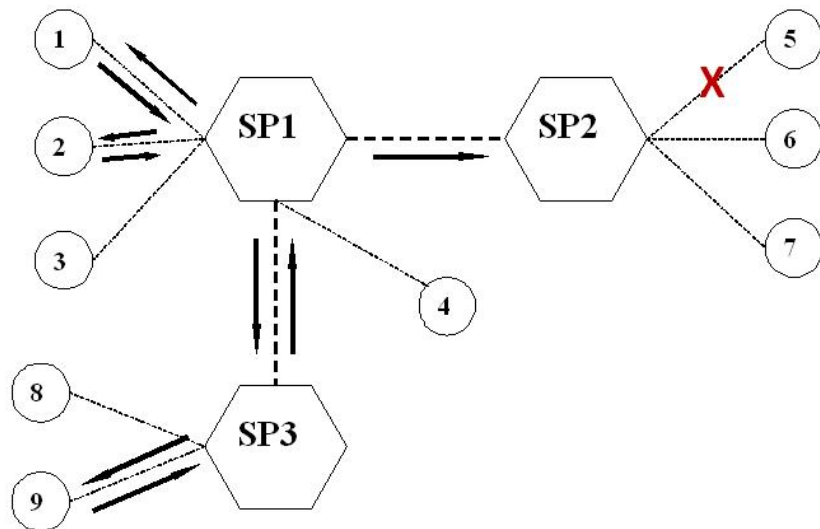
```
1: if a query is issued then
2:   create the query message
3:   send the query message to superpeer
4: end if
5: if a query is received then
6:   if peer possesses data for that query then
7:     create the response message
8:     send the response message to source of query
9:   end if
10: end if
11: if a response is received then
12:   if a response for that query has not been received before then
13:     create the data request message
14:     send the data request message to source of response
15:   end if
16: end if
17: if a data request message is received then
18:   create the data message
19:   send the data message to source of data request
20: end if
21: if a data message is received then
22:   save the data
23: end if
24: if peer moves to another peer group then
25:   send the old superpeer a message indicating the new superpeer
26: end if
```

Algorithm 6 Query caching & resending - Superpeers

```
1: for each stored message on superpeer cache do
2:   if wait-time for that cache entry has not expired then
3:     if target peer of cached message is online then
4:       send cached message to target peer
5:       remove message from cache
6:     end if
7:   end if
8: end for
9: if a response is received then
10:  if a response for that query has not been received before then
11:    if target of response is a peer of that superpeer then
12:      send the response message to target peer
13:    else
14:      for each neighbor superpeer do
15:        send the response message to neighbor superpeer
16:      end for
17:    end if
18:  end if
19: end if
20: if a query is received then
21:  if a message for that query has not been received then
22:    for each peer in the peer group do
23:      if peer is online then
24:        send the query message to the peer
25:      else
26:        save the message to the cache
27:      end if
28:    end for
29:    for each neighbor superpeer do
30:      send the query message to neighbor superpeer
31:    end for
32:  end if
33: end if
```



(a)



(b)

Figure 3.4: Replicating patient data (a) When peer 5 receives a query from peer 9 for a patient record, its counter reaches the threshold value and that peer sends replicas to peers 1 and 2 (b) After some time, peer 5 goes offline and at that time peer 9 issues another query for the same record. Peers 1 and 2 which hold replicas created before can now respond to that query and start transfer.

Algorithm 7 Replication of Patient Data - Peers

```
1: for each record on the peer do
2:   if number of queries received in last 100 time units for that record is greater
   than threshold then
3:     if number of replications for that record is smaller than replication limit
       then
4:       create replica of that record
5:       send the replica to 4 peers in the same peer group
6:       send the replica to 3 peers in other peer groups
7:     end if
8:   end if
9: end for
10: if a query is issued then
11:   create the query message
12:   send the query message to superpeer
13: end if
14: if a query is received then
15:   if peer possesses data for that query then
16:     create the response message
17:     send the response message to source of query
18:   end if
19: end if
20: if a response is received then
21:   if a response for that query has not been received before then
22:     create the data request message
23:     send the data request message to source of response
24:   end if
25: end if
26: if a data request message is received then
27:   create the data message
28:   send the data message to source of data request
29: end if
30: if a data message is received then
31:   save the data
32: end if
33: if peer moves to another peer group then
34:   send the old superpeer a message indicating the new superpeer
35: end if
```

Algorithm 8 Replication of Patient Data - Superpeers

```
1: if a response is received then
2:   if a response for that query has not been received before then
3:     if target of response is a peer of that superpeer then
4:       send the response message to target peer
5:     else
6:       for each neighbor superpeer do
7:         send the response message to neighbor superpeer
8:       end for
9:     end if
10:  end if
11: end if
12: if a query is received then
13:   if a message for that query has not been received then
14:     for each peer in the peer group do
15:       send the query message to the peer
16:     end for
17:     for each neighbor superpeer do
18:       send the query message to neighbor superpeer
19:     end for
20:   end if
21: end if
```

Chapter 4

Evaluation and Simulation Results

In the first section of this chapter, the criteria used in the evaluation process are explained and the simulation environment prepared for measuring the performance of the proposed methods is presented. In the second section, simulation results are presented.

4.1 Evaluation Criteria and Simulation Environment

In the evaluation process, we focus on three aspects: comparison of P2P architecture to Client-Server architecture for the medical system we work on, evaluating the effects of using subscription method in proactive data dissemination, and evaluating the performance of three methods (caching queries for a while, replicating records, and classifying patient data) proposed to improve the reactive data dissemination (query processing).

A crucial issue about medical information systems is that the records must be reached whenever it is necessary. Maximum query satisfaction should be

provided. Because of this fact, we evaluate the methods for query processing by measuring the rate of successful query responses through a metric called Query Success Ratio (QSR).

$$QSR = \frac{\text{Number of queries which are replied}}{\text{Number of total queries issued in the network}} \quad (4.1)$$

Rapid delivery of data is a significant issue in network design, and it is especially crucial in medical information systems. Medical personnel should reach patient information as soon as possible. Therefore, the average delivery time of records throughout the network is a leading evaluation criterion. Average Delivery Time (ADT) is a means to evaluate how fast the data is delivered. In proactive data dissemination, it is the average of time spent from the release of an update to the time the update reaches its destination for each data update. For reactive data dissemination, ADT is measured as the average of time spent from the release of a query until a reply is received.

For proactive data dissemination:

$$ADT = \sum_1^{\#ofupdates} \frac{(Time_{update_reached_destination} - Time_{update_issued})}{\text{Total number of data updates}} \quad (4.2)$$

For reactive data dissemination:

$$ADT = \sum_1^{\#ofqueries} \frac{(Time_{query_replied} - Time_{query_issued})}{\text{Total number of queries}} \quad (4.3)$$

Classification of medical records can provide some benefits by assigning privileges to important types of data. To evaluate the effect of classification of records we use a metric that we call Combined Query Success Ratio (CQSR):

$$CQSR = \sum_1^{\#ofqueries} \text{weight}(i) * QSR(i) \quad (4.4)$$

where i specifies the type of data.

Another metric that we use to capture the effect of data type in determining average delivery time is called Combined Average Delivery Time (CADT):

$$CADT = \sum_1^{\#ofqueries} weight(i) * ADT(i) \quad (4.5)$$

where i specifies the type of data.

In both CQSR and CADT, $weight(i)$ specifies the weight indicating the significance of that type of data, taking a value in between 0 and 1. The sum of all weights is 1. $ADT(i)$ and $QSR(i)$ are measured ADT and QSR values for a specific type of data or query.

In P2P networks, heavy network traffic is one of the main problems. There are many works which focus on reducing the message overhead of P2P networks. We will keep an eye on that issue in most of the tests.

Since, storage capacity of mobile devices is limited, total storage volume required becomes a significant criterion. Especially, when replication of data is applied, performance of the system can be affected dramatically. We use a metric called Total Record Storage Volume (TRSV) in order to measure the data storage requirement. It is the total size of all data records stored in all elements of the network.

In our simulation of a P2P system, peers are allowed to move between peer groups and they may become unavailable for specific durations. Data updates and queries are generated periodically. In our simulated topology, superpeers are aligned according to the square topology and peers are connected to superpeers. Figure 4.1 shows an example of a square topology with 9 superpeers.

In order to perform necessary tests, CSIMforJAVA [30] discrete event simulator is used. CSIMforJAVA is a tool that is used in many studies to simulate

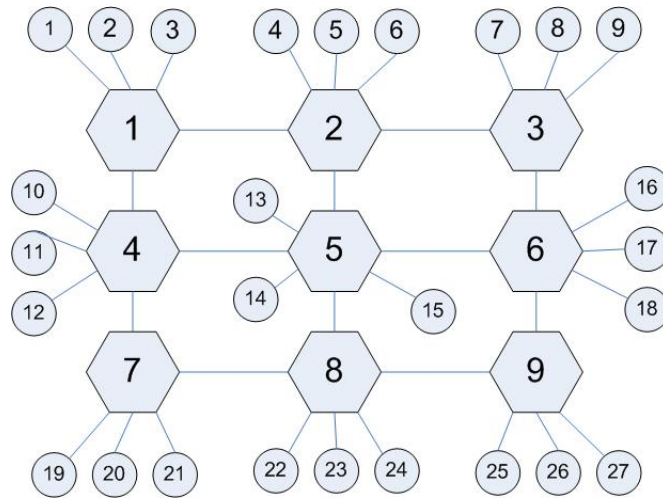


Figure 4.1: The square topology with 9 superpeers each of them having 3 peers.

network protocols, telecom communication systems, software and hardware applications, etc. It provides a JAVA library which includes classes for multi-threaded simulation. Therefore, JAVA [47] has been used as the programming language with the Eclipse IDE.

4.2 Simulation Results

4.2.1 Comparison of Client-Server and P2P paradigms

The architecture of a healthcare information system can follow the client-server or peer-to-peer paradigm. As stated before, client-server architecture has some benefits and drawbacks. Although this architecture is advantageous in terms of search efficiency, increased workload on the servers or failure of central facilities may cause significant performance problems in the network. Therefore, we choose P2P approach. To compare these architectures also numerically, we evaluated them on two basic operations of our system: proactive (push operation) and reactive (query processing) data dissemination.

Figure 4.2 shows ADTs (i.e., how long it takes to deliver update messages for the push operation) in P2P and Client-Server architectures. We used the subscription method described in Chapter 3 for P2P data dissemination. We tested both networks with varying numbers of peers, which are organized into a square topology. Number of peer groups varies from 9 to 100, each of them having 10 peers inside. In Figure 4.2, two curves correspond to two models (triangles for P2P and squares for Client-Server). The x -axis of the graph shows the network size in terms of peers and y -axis shows the ADTs of messages in simulation time units (1 simulation time unit is 10 seconds). The figure shows that when the number of peers is larger than a certain number (approximately 160 peers), using Client-Server architecture causes messages to take more time to reach the relevant server and peers. In smaller networks, using P2P architecture causes longer average delivery time. However, as the networks get bigger, more peers use the server at the same time. Since the server has limited capacity, after a certain value, it cannot meet the demands of peers and it takes more time to process the requests.

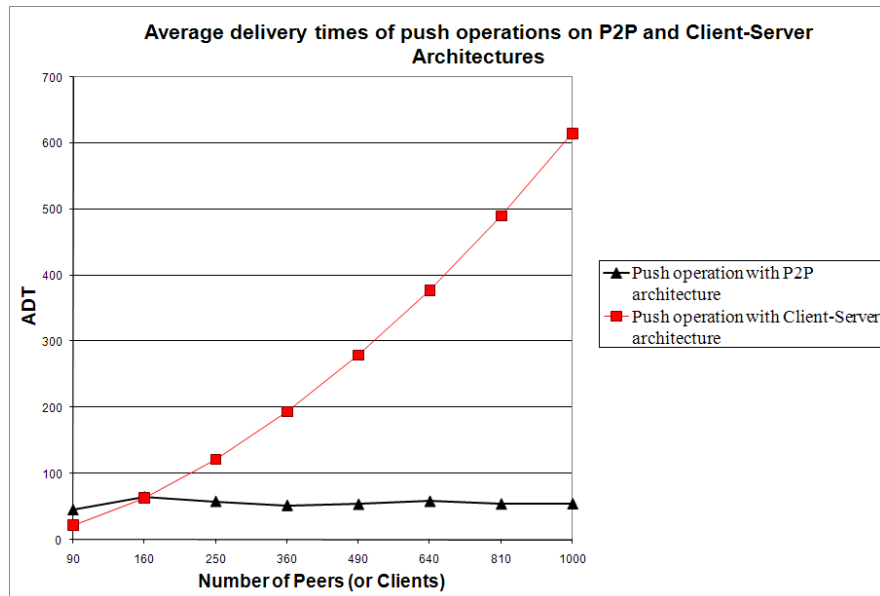


Figure 4.2: Difference in average delivery times of messages for push operations with P2P and Client-Server architectures.

As we mention in Chapter 1, P2P networks can withstand higher update

rates. In order to validate this claim, we performed tests on both architectures with different update rates. Figure 4.3 shows the results of these tests in a column chart. In this test, there are 25 peer groups, each of them having 10 peers, and 2 peers in each peer group issues updates periodically. The update period is varied from 1 second to 10 seconds and they are placed in the x -axis of the chart. The y -axis shows the ADT. Black columns show delivery times for P2P architecture where grey ones show delivery times for Client-Server architecture. It can be seen that when update frequency gets higher (i.e., update period gets lower), Client-Server architecture delivers messages with much higher delay. When update frequency gets lower (i.e., update period gets longer), both architectures perform similar. This is the result that we had foreseen. When updates arrive frequently, the server cannot forward all of them at the same time, after some extent. It starts to buffer incoming messages and relays them when it is available.

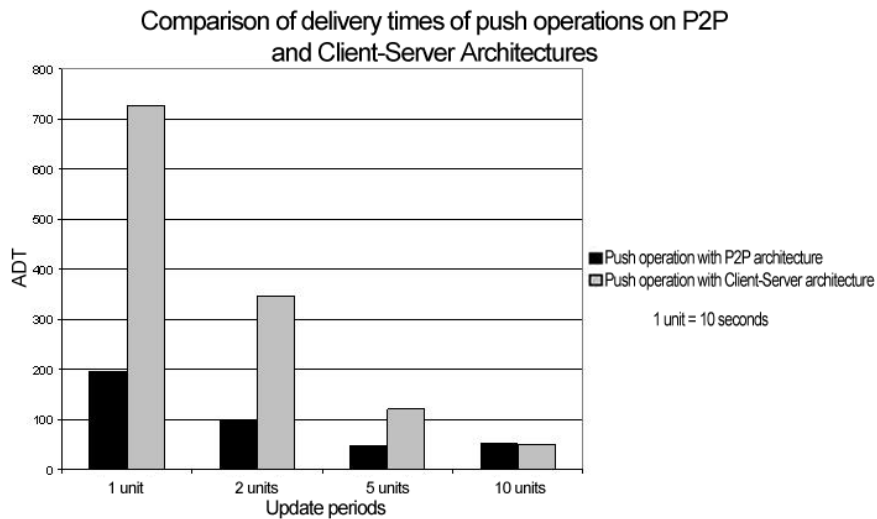


Figure 4.3: Difference in average delivery times with various update periods.

4.2.2 Evaluating Methods for Proactive Data Dissemination

In this section, we evaluate benefits of our subscription method compared to the flooding approach which is based on sending updates to all members in the network. The subscription technique was introduced earlier. Patient records all have their subscriber lists and updates are sent according to those lists.

In Figure 4.4, the number of messages created when these two methods are applied in P2P networks of different sizes, can be seen. The curve with square markers shows the total number of messages when the subscription method is used. The curve with triangle markers shows the same metric without subscriptions. As the figure shows, the number of messages created when subscription method is used is much smaller than the flooding method. Since, with subscription, a peer issuing an update message does not send it to peers which are not interested in that update, the number of created messages becomes much less. The gap between the two methods increases with the network size. Messages created in order to maintain subscription lists do not cause the number of total messages exceed the one with the flooding method. Storage cost of subscription lists is ignorable because they are textual data and do not require that much space as the common medical data like image or video files.

Figure 4.5 compares again the subscription method with the flooding method, but also shows what happens when the subscribers per patient record is increased. The curve with square markers shows the total number of messages when the subscription method is used. The curve with triangle markers shows the same metric when the number of subscriptions per record is doubled. The dashed curve with diamond markers shows the message count for the flooding method. The figure shows that even the number of subscribers per patient record is increased; the total traffic produced by subscription method is still less than the flooding method.

Figure 4.6 compares the methods in terms of ADT. The curve with square markers shows the average delivery time when the subscription method is used.

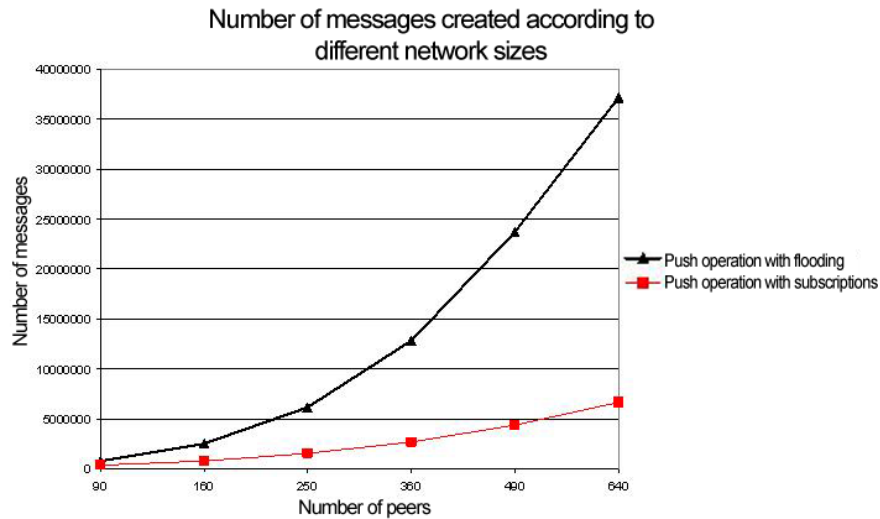


Figure 4.4: Comparison of push operation with flooding and push operation with subscriptions in terms of total number of created messages.

The curve with triangle markers shows the same metric when the number of subscriptions is doubled. The dashed curve with diamond markers corresponds to the flooding method. Results show that using subscriptions causes smaller average delivery time than directly sending the updates to all peers, regardless of network size. As it was seen on previous figures, when update messages are sent to all peers, the network traffic increases dramatically. That also causes the forwarding peers to buffer messages and deliver them later. This increases the ADT. As Figure 4.6 shows, adding more subscribers to records does not cause a significant increase in ADT. Messages created to update subscription lists themselves do not have a major impact on delay.

4.2.3 Evaluating Methods for Reactive Data Dissemination

In this section, we evaluate the three techniques we propose to increase the performance of reactive data dissemination (query processing) in the proposed P2P medical system architecture. The proposed techniques are:

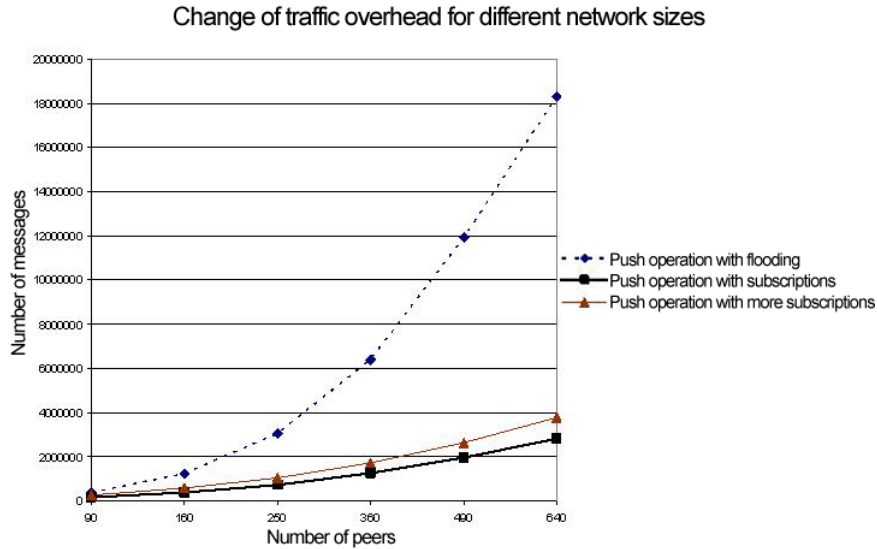


Figure 4.5: Comparison of push operation by assigning more subscriptions in terms of total number of created messages.

- Caching incoming queries on superpeers and resending them periodically for the duration of some certain time interval.
- Creating replicas of patient records depending on the frequency of queries made for those records.
- Classifying patient data into different types and treating them differently depending on their update frequency and importance.

4.2.3.1 Caching and resending queries

The charts presented in Figures 4.7, 4.8, 4.9, 4.10 and 4.11 show the results of experiments performed in a network of 250 peers to evaluate how caching and resending queries affects the performance. The number of peers (i.e., doctors) employed in the experiments is quite realistic to be seen in a typical medical environment. A total number of 1000 queries are issued in each experiment. 2 queries are issued at each peer group in every 5 units of time. This is supposed to be equal to 50 seconds (1 time unit = 10 seconds). Peers in the network

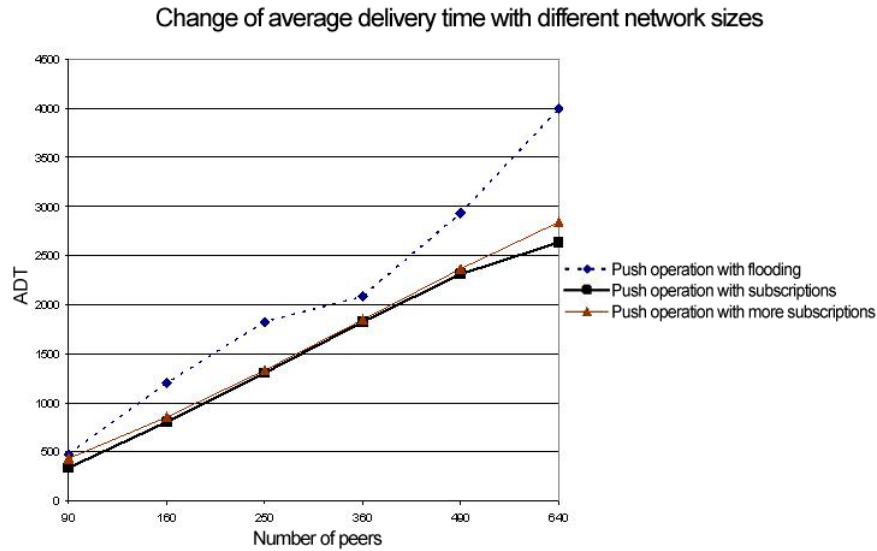


Figure 4.6: Comparison of push operation by assigning more subscriptions in terms of average delivery time for updates.

may become unavailable at some time intervals of a simulation run. They can disconnect from their former superpeers, move to other peer groups and connect to other superpeers.

For our simulation experiments, we implemented the query caching mechanism described in Section 3.2.1. In that mechanism, queries are cached and resent to target peers periodically for some specific duration called “Wait-time”. Every resend period (which is set to 1 simulation time unit), cached queries are resent if the time passed from their arrival to that peer did not exceed the wait-time. A peer which can answer the query may be unavailable at the moment the query reaches the peer. But, if the method we proposed is used, that peer may still receive the query when it becomes available. Therefore, as we cache the queries for some time and resend them periodically to formerly unavailable peers, we increase the number and ratio of queries that are answered successfully. This ratio is represented by our metric, Query Success Ratio (QSR). As it can be seen in Figure 4.7, as we increase the wait-time (i.e., caching time), QSR increases significantly.

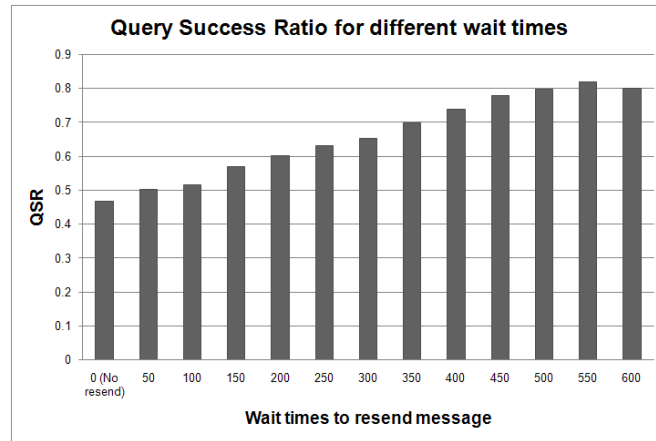


Figure 4.7: Comparison of resend operation for different wait times in terms of QSR.

Caching and resending queries is expected to cause more traffic and increase delivery times for the queries and replies. But, in our tests, we observed that the changes in the number of messages created in the network and delivery times for the queries are not that significant compared to the increase in QSR. In a medical environment, when a medical personnel wants to retrieve a patient's data, the data must be retrieved sooner or later. Being able to receive a reply is more important than the time and network traffic constraints. Therefore, it is beneficial to use the caching and resending mechanism. Figures 4.8 and 4.9 illustrate the changes in total number of messages created in the network and average delivery times for queries issued in the simulation environment, respectively.

As it can be seen on Figure 4.8, when the caching and resending mechanism is not used, the total number of messages created is about 160000. As we use the mechanism and gradually increase the time to hold the queries in the cache (wait-time), the number increases to about 190000 at most. The change is about 18% and it is not a big change compared to the increase in QSR (77%).

Similarly, in Figure 4.9, it can be seen that there is no significant increase in the average delivery time of queries. ADT increases from 1700 time units to 2000-2100 units. The change is about 25% which is well below the increase in

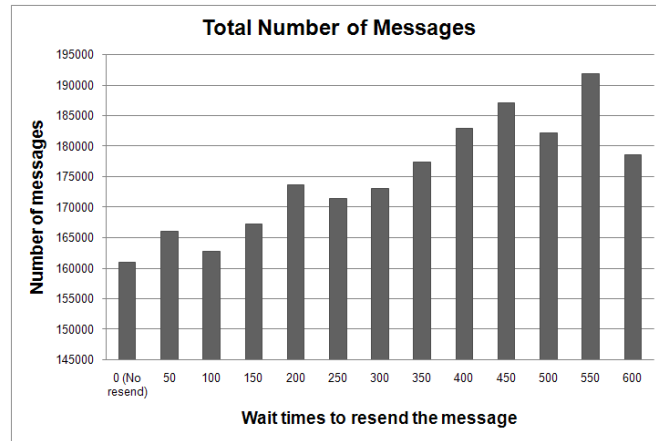


Figure 4.8: Comparison of resend operation for different wait-times in terms of Total Number of Messages.

QSR (77%).

Figures 4.10 and 4.11 show the effect of caching and resending queries on finding peers which include queried objects. Dark grey figures show the number of hits achieved by caching and resending. It can be seen that as we increase wait-times, total number of hits increases and hits caused by resending play a bigger role in that. Figure 4.10 shows the number of hits, while Figure 4.11 shows the hit ratio.

4.2.3.2 Creating replicas of patient records according to statistics

The idea of making more copies of popular data can increase the access rate and reliability of desired data. To see the effect of replication, in this section, we present the results of the experiments we performed by changing the values of three parameters that are used in our replication algorithm proposed in Section 3.2.2. These parameters are:

- *The threshold value for a patient record (Query threshold).* If the number of queries issued for a record within some specified time interval exceeds

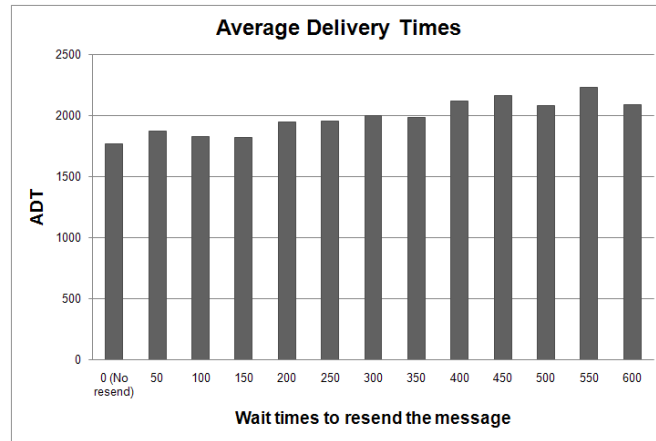


Figure 4.9: Comparison of resend operation for different wait-times in terms of ADT.

the threshold, that particular record is replicated in other peers within the peer group and in other peer groups.

- *Replication period.* It is the period of replication. In each time period, records are checked if they should be replicated. Hence, replication is performed periodically.
- *Replication limit.* Replication of records can reach an unacceptable level if it is not controlled. In such a case, benefits of replication may become useless due to the negative effects. Therefore, via experiments, we try to find an appropriate limit, while also showing that unlimited replication can be harmful.

The parameters above play an important role in the success of replication methodology. The values used during our tests for those parameters are selected as realistic as possible considering a real environment. We also identify the ranges of values that provide good performance. For example, replication limit values are determined as 5 and 10, since we observed that it is not advantageous to use a value bigger than 10. Likewise, query threshold value is varied as 1, 3 and 5. Also, the replication period value is set to 1, 3 and 5 time units (corresponding to 10, 30 and 50 seconds respectively) in order to see its effect.

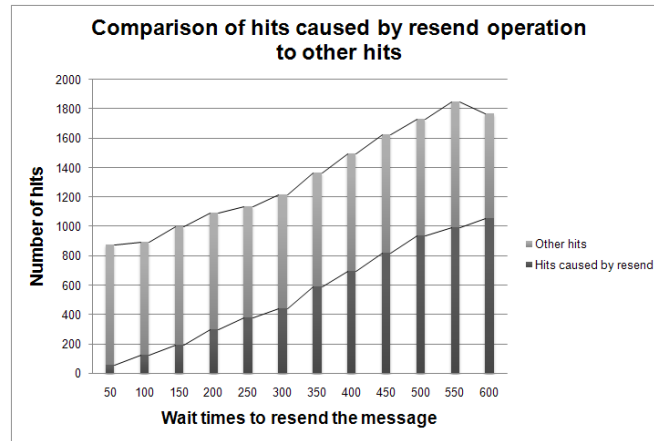


Figure 4.10: The number of query hits caused by resend operation and other hits for different wait-times.

In the first set of tests, we tried to observe the effect of different query threshold values on QSR, ADT, number of created messages, and TRSV, with different replication limits. These experiments are performed on a simulated network of 250 peers as in the previous section. A total of 1000 queries are issued by peers periodically; 2 queries in every 10 seconds. Data records are replicated if the query threshold value is reached in the last 100 time units (1000 seconds).

Figure 4.12 shows the effect of replicating patient data on the Query Success Ratio (QSR) for different query threshold values. This figure clearly shows that applying replication increases QSR. When more replicas of a record exist, even a peer which holds that record is unavailable, the possibility of finding another peer which holds that record is high. Therefore, QSR increases. The lower the query threshold is, the higher the QSR becomes. If the query threshold is 1, then there is an increase of about 50%. The increase in QSR is about 30%, if the query threshold is 3, and 10% if the threshold is 5. Using a lower threshold increases QSR, but it also increases the total number of messages generated to disseminate replicas (Figure 4.14). Another observation that we can make in this figure is that limiting the number of replicas does not have a significant effect on QSR.

Although replication increases the chance of finding a nearby peer holding the queried data, that may not decrease the Average Delivery Time (ADT), contrary

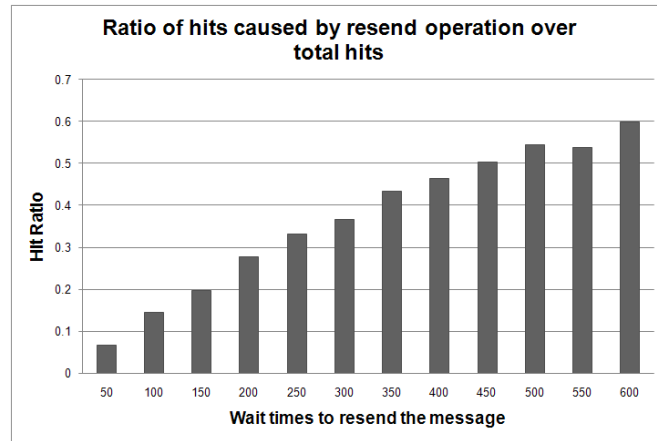


Figure 4.11: The ratio of query hits caused by resend operation over total number of hits for different wait times.

to the expectation. The extra traffic created due to messages for maintenance of the replicas increases the delay for forwarding the query and query hit messages and therefore causes an increase in ADT. The important question is how much is that increase compared to the improvement in QSR. Is it worth applying this method? In order to find an answer to that question, we should take a look at Figure 4.13 to compare the gain and the cost. The figure depicts that there is an increase of 3% - 4% in ADT on the average if replication is applied. On the other hand, in Figure 4.12, we see that there is an increase of 10%, 30% and 50% in QSR for different query threshold values. By looking at these two figures, we can conclude that making reasonable number of replicas is beneficial for the overall performance. Also, for higher threshold values, less replication is done and less messages are created for maintenance, causing less extra traffic (shown in Figure 4.14). Hence, ADT becomes lower. We can see this from Figure 4.12, as well.

Due to similar reasons for the change in ADT, there may be an increase in the total number of messages. It is also an overhead and we should see if it pays or not. Figure 4.14 shows the effect of replication on the total number of messages (total of messages for maintenance, queries and data) created in the network. If we examine the figure, we see that there is roughly 8% - 10% increase in the total number of messages created. Comparing that percent to the improvement in

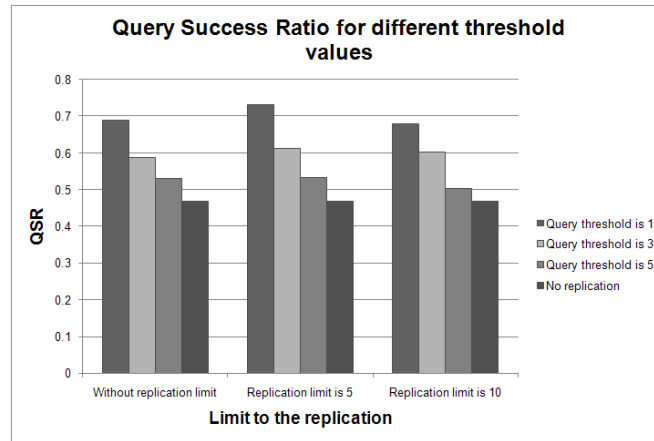


Figure 4.12: Change in Query Success Ratio (QSR) for different values of query threshold when different replication limits are applied.

QSR, we see that the advantage becomes the heavier side. The figure also shows that for higher query threshold values, less replication is done, and therefore, less messages are created for replica maintenance.

In our tests we observed that there is a huge difference in Total Record Storage Volume (TRSV) required throughout the network when no replication limit is applied, compared to situations where there is a limit. TRSV is about 2000 times bigger when replication is not limited. There is no doubt that limitation is required to avoid unnecessary storage usage. Figure 4.15 shows that when query threshold value gets higher, less storage is required, and higher replication limits mean more storage requirement. In order to find a reasonable replication limit, we should consider all these results together. Depending on the requirements of the system, more storage volume can be tolerated in order to obtain better QSR.

Figure 4.16 shows the effect of replicating patient data on the QSR metric for various replication periods. As explained in the replication algorithm in Section 3.2.2, records in a peer are replicated to other peers in every 1, 3 or 5 simulation time units. Figure 4.16 clearly shows that applying replication increases QSR. As the best replication period, a period of 3 time units (i.e., 30 seconds) seems to be the best choice. Approximately, replicating records every 30 seconds brings an

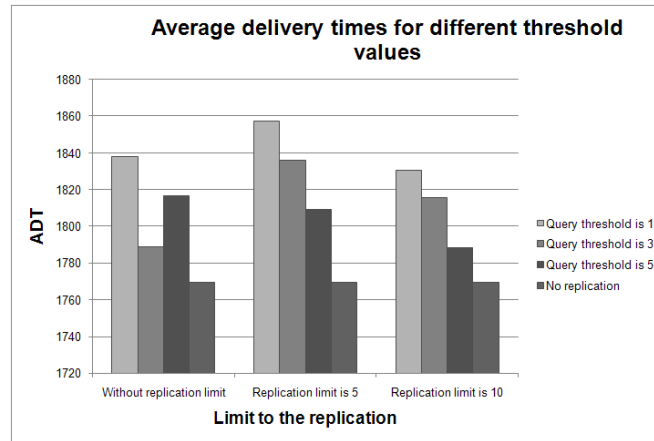


Figure 4.13: Change in Average Delivery Time (ADT) for different values of query threshold when different replication limits are applied.

outcome of about 0.03% - 0.04% increase in QSR compared to other replication periods. The increase in QSR compared to no replication is about 30%-40%. Using different replication periods is not directly related to using different query threshold values. Best values of those two parameters should be used for optimal result.

Figure 4.17 can give more information about which replication period is better. In the figure, if we look at the ADT values for different replication limits, we see that a period of 3 units (30 seconds) is the worst one among 1, 3 and 5. If the replication limit is applied as 5, then a period of 1 unit is better. For a replication limit of 10, 5 units of replication period gives lower ADT. 3 simulation time units of replication period was slightly better for the QSR value. But, for ADT, it is not the best fit. Here, the choice depends on the needs of the network. Since, in a medical environment, success of a query is more important than its turnaround time, and the ADT value does not increase drastically (3% - 4%), 3 simulation time units (i.e., 30 seconds) can be chosen as the replication period.

Figure 4.18 shows the number of created messages for different replication periods. As we see, as period becomes longer, less messages are created in the network. The number of messages created is larger than no replication, but the

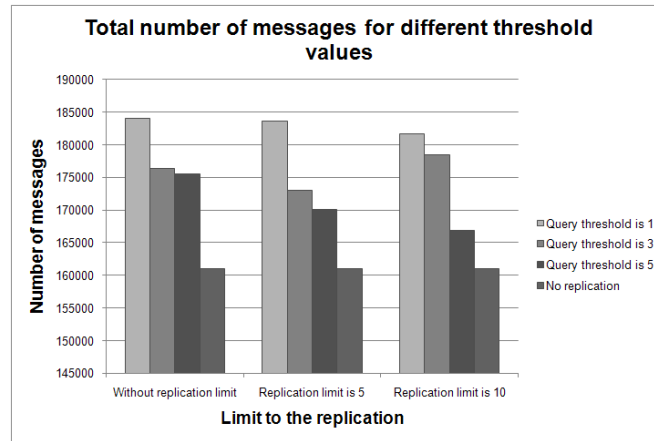


Figure 4.14: Change in Total Number of Messages for different values of query threshold when different replication limits are applied.

increase is about 10% at most. The overhead is not that big when we consider the gain in QSR.

Another important metric we use is TRSV. The TRSV values for replication limits of 5 and 10 are shown in Figure 4.19, and they are not much different than the ones we showed when investigating different query thresholds. Values created for different replication periods are not so much distinguishable from each other.

4.2.3.3 Classifying patient records according to their characteristics

In medical information systems, not all data have the same importance or characteristics. Some of them can change very frequently and some may be so crucial that queries for these data records must definitely be replied. In order to satisfy demands and increase efficiency, different types of data records may require special treatment. Distinguishing the data may enable us to take care of special data types with higher priority rather than dealing with all of them at the same level. Therefore, we performed experiments by assigning 4 different category levels to patient records. We gave some privileges to records depending on their classes.

Each class of patient record has a different weight value between 0 and 1. Sum

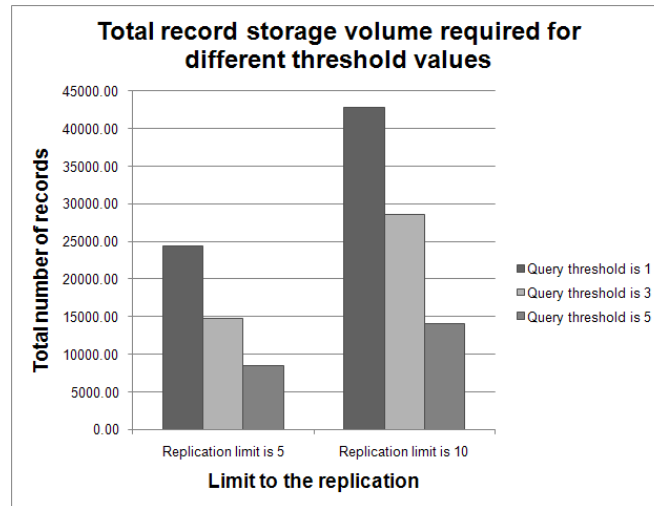


Figure 4.15: Change in Total Record Storage Volume (TRSV) for different values of query threshold when replication limits are 5 and 10.

of these weights add up to 1. In our experiments, weight values used for 4 different types of categories are 0.1, 0.2, 0.3 and 0.4. For the cache & resend method, the weight value of each category is used to determine the wait-time. Wait-times applied to patient records are directly proportional to their weights. Similarly, for the replication method, weight values were used as factors of number of replicas created in a replication period. Number of copies created in a replication action is directly proportional to the record's weight value.

First, we performed experiments to evaluate the resend method and collected results for each of the 4 categories. Figures 4.20 and 4.21 compare QSR values when classification is not applied and when classification is applied. On Figure 4.20, we see that for 3 of the categories, there is an increase in QSR about 8%, if classification is applied. Only for category 1 (the category with the lowest weight value) QSR gets worse with a change of 6%. However, this category is the least important one since it has the lowest weight value. Moreover, in Figure 4.21, QSR value is better for all categories when patient records are classified. This is a sign to indicate that classifying patient records is beneficial in terms of Query Success Ratio for medical records. This is an important issue because we provide the flexibility to system designers to assign privileges to some kind of data and

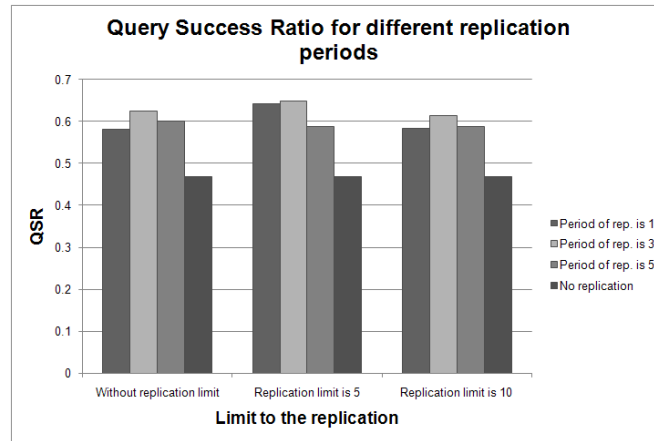


Figure 4.16: Change in Quality Success Ratio (QSR) for different values of replication period when different replication limits are applied.

we see that this effort pays off.

Combined QSR value (CQSR) shows the overall performance of the methodology. That metric was described in earlier sections. It gives a general understanding about the outcomes. Figures 4.22 and 4.23 help us to evaluate the effect of resend operation when patient records are classified. In Figure 4.22, we see that especially for smaller networks, resend operation increases the efficiency in terms of successful queries. However, caching and resending operation also has a negative impact on ADT. Since queries remain for a longer time in the network, time spent for locating data becomes higher. Figure 4.23 illustrates CADT results. In contrast to the improvement in QSR, ADT becomes worse. We need to overcome this flaw. Replication mechanism becomes important at this stage, because of its impact on ADT.

Using replication with classification may cause a decrease in ADT. In order to validate this, we have performed experiments on a network of 250 peers and obtained results for each of the categories. Unlike the experiment we performed with unclassified data, the replication method caused a decrease in ADT when the classification strategy is applied. Since relatively insignificant data is replicated

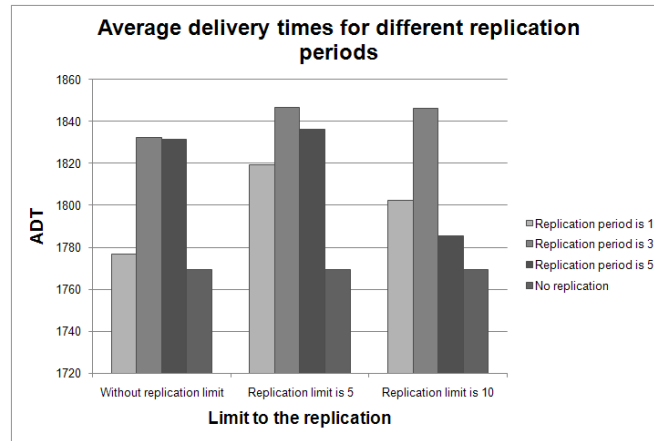


Figure 4.17: Change in Average Delivery Time (ADT) for different values of replication period when different replication limits are applied.

less, the traffic overhead has decreased, causing lower ADT. Figure 4.25 illustrates the effect of replication on ADT for different categories. The figure shows that (especially for category 1) using the replication technique decreases ADT significantly. Figure 4.26 also shows that the value of CADT becomes improved if classification is applied.

The experiments to see the effect of replication on QSR did not give consistent results to come to a decision. For two categories it was beneficial, but for the other ones it was not. But, at least, we know that it does not cause any harm on that criteria.

In the light of the results discussed above, we can claim that if we use replication along with caching & resending queries, we can eliminate, or at least reduce, the negative effect of resending on ADT metric without giving up the improvement achieved by that methodology in QSR metric.

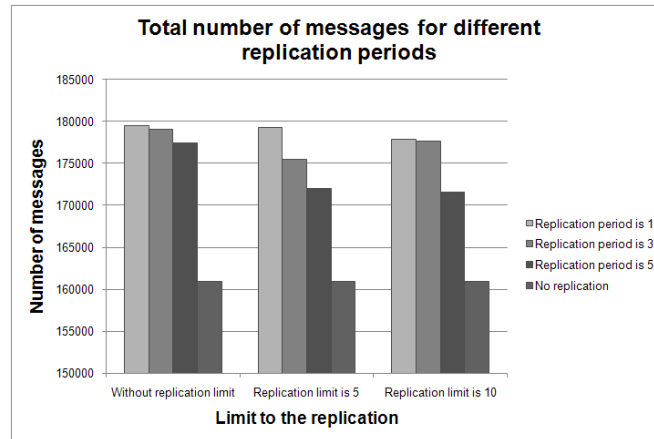


Figure 4.18: Change in Total Number of Messages for different values of replication periods when different replication limits are applied.

4.3 Summary of Results

In the performed experiments, we first observed that using the P2P architecture reduces the delivery time compared to the client-server architecture. Then, we observed that using the subscription method decreases the number of messages and the average delivery time (ADT). After that, we investigated methods for reactive data dissemination and observed that caching & resending queries increases Query Success Ratio (QSR) without a significant increase in ADT. Then, investigating the effect of applying replication, we observed that replication causes an increase in both QSR and ADT. Total number of messages created and the TRSV value when replication is applied, were investigated as well. In the last section, we showed that classifying data helps us to increase QSR for different network sizes with a slight increase in ADT.

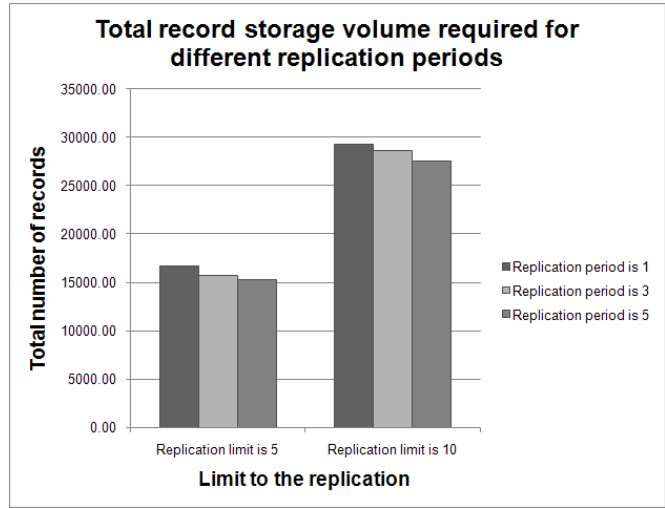


Figure 4.19: Change in Total Record Storage Volume (TRSV) for different values of replication periods when replication limits are 5 and 10.

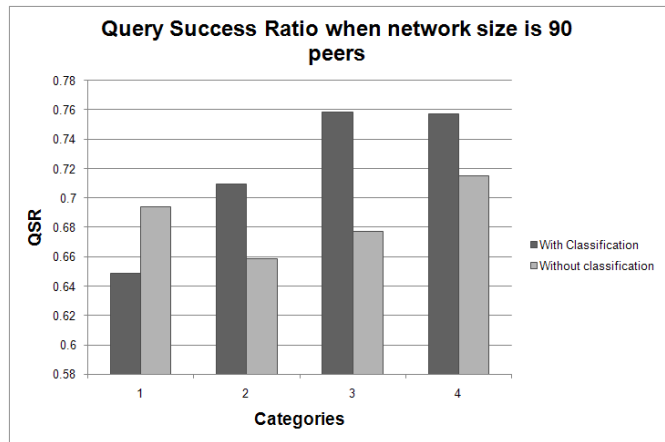


Figure 4.20: QSR values for four different categories in a network of 90 peers when classification is applied and not applied.

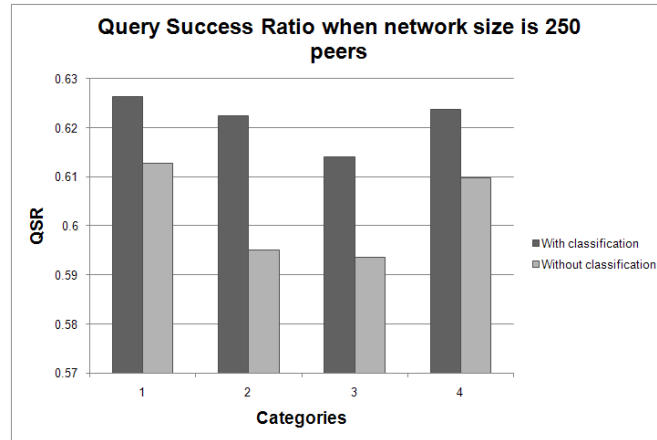


Figure 4.21: QSR values for four different categories in a network of 250 peers when classification is applied and not applied.

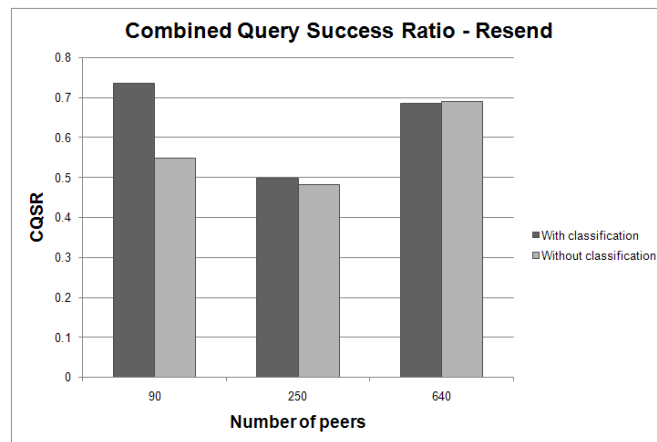


Figure 4.22: CQSR values for four different categories in networks of 90, 250 and 640 peers when classification is applied and not applied.

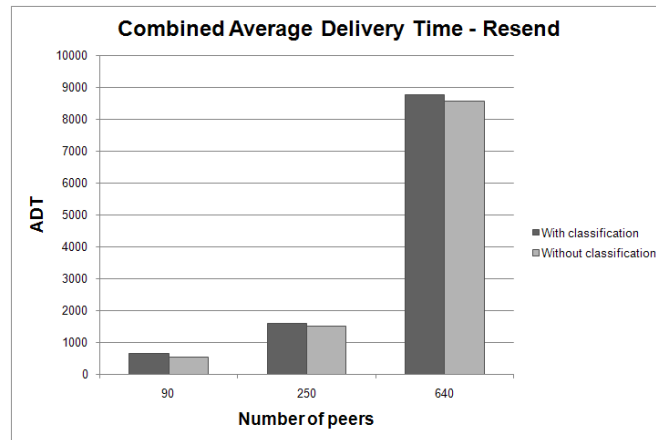


Figure 4.23: CADT values for four different categories in networks of 90, 250 and 640 peers when classification is applied and not applied.

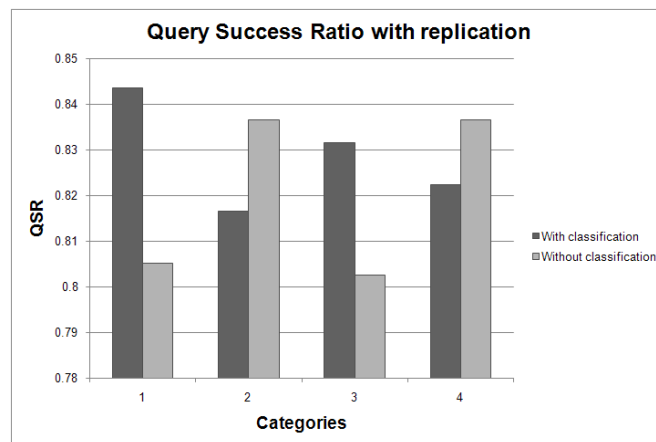


Figure 4.24: QSR values for four different categories in a network of 250 peers when classification is applied and not applied. Here, replication methodology is applied.

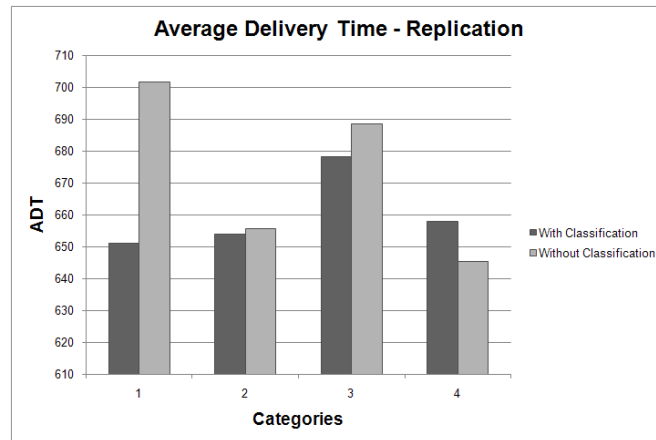


Figure 4.25: ADT values for four different categories in a network of 250 peers when classification is applied and not applied. Here, replication methodology is applied.

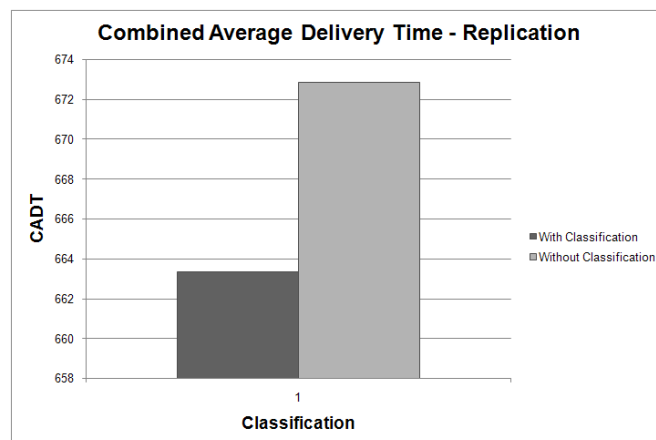


Figure 4.26: CADT values for four different categories in a network of 250 peers when classification is applied and not applied.

Chapter 5

Conclusion and Future Work

In this thesis, various data dissemination strategies to be used in a mobile peer-to-peer network designed for a healthcare information system are presented. The performance of these strategies are evaluated according to the properties of medical information systems.

We first describe the system model along with the hybrid peer-to-peer architecture used. Types of medical data assumed in our system are described, and the system overview is provided. Then, proactive and reactive data disseminations in this system are introduced. As a proactive dissemination strategy, using subscription lists to deliver updates is proposed. In that method, updates of a patient record are delivered to the peers and peer groups in the subscriber list of that record. It is shown through performance tests that this method reduces the network traffic and the time to receive update messages. For reactive data dissemination, three strategies are proposed. Caching and resending queries involve storing received query messages in a superpeer for some time, and periodically sending them to peers. It is observed that this strategy increases the rate of successful queries without a significant increase in the average delivery time. Another strategy adopted is replicating patient records. According to this method, patient records are replicated periodically, if they are frequently queried. In the evaluation of this method, it is shown that an increase is observed in both the successful query rate and the average delivery time. The last strategy used for

reactive data dissemination is classifying patient records. The idea here is to separate records according to their importance and to prevent relatively unimportant records from keeping relaying peers busy. It is observed that this strategy works well with an increase in the successful query rate for different network sizes and a slight increase in the average delivery time. The simulation results show that the proposed data dissemination strategies provide reasonable solutions to the requirements of a medical peer-to-peer system.

The performance tests were conducted for a network size of at most 1000 peers which can be reasonable for a hospital. It can be possible to expand this system and connect several hospitals to form a network of hospitals. Scalability property of peer-to-peer networks allows that. Further research can evaluate the strategies proposed in this thesis for a network of hospitals. Also for this thesis, one of the aims was to provide a guidance to people who will design such a system in the future. Realization of such a system can lead to an effective usage of mobile peer-to-peer networks in real life.

Bibliography

- [1] D. T. Ahmed and S. Shirmohammadi. Design issues of peer-to-peer systems for wireless ad hoc networks. In *Proceedings of the Sixth IEEE International Conference on Networking (ICN '07)*, pages 26–31, 2007.
- [2] D. T. Ahmed and S. Shirmohammadi. Multi-level hashing for peer-to-peer system in wireless ad hoc environment. In *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, pages 126–131, March 2007.
- [3] M. Ancona, G. Doderò, F. Minuto, M. Guida, and V. Gianuzzi. Mobile computing in a hospital: the ward-in-hand project. In *Proceedings of the ACM Symposium on Applied Computing (SAC '00)*, pages 554–556, 2000.
- [4] B. Beverly Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the 19th International Conference on Data Engineering*, pages 49–60, March 2003.
- [5] Bluetooth Special Interest Group, <http://www.bluetooth.com>.
- [6] J. Brassey, G. Elwyn, C. Price, and P. Kinnersley. Just in time information for clinicians: a questionnaire evaluation of the ATTRACT project. *British Medical Journal*, 322(7285):529–530, 2001.
- [7] P.-A. Chirita, P. Alex, R. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Designing semantic publish/subscribe networks using super-peers. *Semantic Web and Peer-To-Peer*, pages 159–179, 2005.

- [8] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 23–32, 2002.
- [9] D. Cypher, N. Chevrollier, N. Montavont, and N. Golmie. Prevailing over wires in healthcare environments: benefits and challenges. *IEEE Communications*, 44(4):56–63, April 2006.
- [10] C. Doulkeridis, V. Zafeiris, and M. Vazirgiannis. The role of caching and context-awareness in P2P service discovery. In *Proceedings of the 6th ACM International Conference on Mobile Data Management (MDM '05)*, pages 142–146, 2005.
- [11] L. Garcés-Erice, E. Biersack, P. A. Felber, K. W. Ross, and G. Urvoy-Keller. Hierarchical peer-to-peer systems. In *Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par)*, pages 643–657, 2003.
- [12] S. Goel, M. Singh, D. Xu, and B. Li. Efficient peer-to-peer data dissemination in mobile ad-hoc networks. In *Proceedings of Parallel Processing Workshops*, pages 152–158, 2002.
- [13] A. Graves, B. Wallace, S. Periyalwar, and C. Riccardi. Clinical grade - a foundation for healthcare communications networks. In *Proceedings of the 5th International Workshop on Design of Reliable Communication Networks (DRCN)*, pages 8–15, October 2005.
- [14] E. Harjula, M. Ylianttila, J. Ala-Kurikka, J. Riekkii, and J. Sauvola. Plug-and-play application platform: towards mobile peer-to-peer. In *Proceedings of the 3rd ACM International Conference on Mobile and Ubiquitous Multimedia (MUM '04)*, pages 63–69, 2004.
- [15] Healthcare Information and Management Systems Society, <http://www.himss.org>. *HIMSS*.
- [16] T. Horozov, A. Grama, V. Vasudevan, and S. Landis. Moby - a mobile peer-to-peer service and data network. In *Proceedings of the International Conference on Parallel Processing*, pages 437–444, 2002.

- [17] D. Hudson and M. Cohen. Structuring medical information for computer-assisted decision support. In *Proceedings of the 19th IEEE Annual International Conference of the Engineering in Medicine and Biology*, volume 3, pages 953–956, October–November 1997.
- [18] IEEE 802.11 Working Group, <http://grouper.ieee.org/groups/802/11/>. *IEEE 802.11 Wireless Local Area Networks*.
- [19] IEEE 802.15 Working Group, <http://grouper.ieee.org/groups/802/15/>. *IEEE 802.15 Wireless Personal Area Networks*.
- [20] IEEE 802.16 Working Group, <http://grouper.ieee.org/groups/802/16/>. *IEEE 802.16 Wireless Metropolitan Area Networks*.
- [21] International Medical Informatics Association, <http://www.imia.org>. *The IMIA vision*.
- [22] S. Jiang, L. Guo, and X. Zhang. Lightflood: an efficient flooding scheme for file search in unstructured peer-to-peer systems. In *Proceedings of the International Conference on Parallel Processing*, pages 627–635, October 2003.
- [23] M. Joseph, M. Kumar, H. Shen, and S. Das. Energy efficient data retrieval and caching in mobile peer-to-peer networks. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops*, pages 50–54, March 2005.
- [24] K. Kant, R. Iyer, and V. Tewari. A framework for classifying peer-to-peer technologies. In *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02)*, pages 368–375, May 2002.
- [25] O. Kassinen, T. Koskela, E. Harjula, J. Ala-Kurikka, P. Pohjanen, and M. Ylianttila. Group-based content push with dynamic session startup. In *Proceedings of the 4th ACM International Conference on Mobile and Ubiquitous Multimedia (MUM '05)*, pages 135–141, New York, NY, USA, 2005. ACM.

- [26] W. Kellerer, Z. Despotovic, M. Michel, Q. Hofstatter, and S. Zols. Towards a mobile peer-to-peer service platform. In *Proceedings of the International Symposium on Applications and the Internet Workshops (SAINT)*, pages 2–5, January 2007.
- [27] T. Kurt. Enabling mobile peer-to-peer networking. In *Proceedings of the Mobile and Wireless Systems, Lecture Notes in Computer Science*, volume 3427, pages 219–234, Dagstuhl, Germany, January 2005.
- [28] Y.-H. Lin, I.-C. Jan, P.-I. Ko, Y.-Y. Chen, J.-M. Wong, and G.-J. Jan. A wireless PDA-based physiological monitoring system for patient transport. *IEEE Transactions on Information Technology in Biomedicine*, 8(4):439–447, Dec. 2004.
- [29] N. Maibaum and T. Mundt. JXTA: a technology facilitating mobile peer-to-peer networks. In *Proceedings of the International Mobility and Wireless Access Workshop (MobiWac)*, pages 7–13, 2002.
- [30] Mesquite Software, <http://www.mesquite.com/csimforjava/>. *CSIMforJAVA*.
- [31] A. Milenkovic, C. Otto, and E. Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications*, 29(13-14):2521–2533, August 2006.
- [32] A. Mondal, S. K. Madria, and M. Kitsuregawa. CADRE: A collaborative replica allocation and deallocation approach for Mobile-P2P networks. *Proceedings of International Symposium on Database Engineering and Applications*, 0:21–28, 2006.
- [33] A. Mondal, S. K. Madria, and M. Kitsuregawa. EcoRep: An economic model for efficient dynamic replication in mobile-p2p networks. In *Proceedings of the International Conference on Management of Data (COMAD)*, December 2006.
- [34] A. Mondal, S. K. Madria, and M. Kitsuregawa. Research issues and overview of economic models in Mobile-P2P networks. In *Proceedings of the 18th International Conference on Database and Expert Systems Applications (DEXA '07)*, pages 809–813, September 2007.

- [35] J. M. Monteiro, A. Brayner, and S. Lifschitz. A mechanism for replicated data consistency in mobile computing environments. In *Proceedings of the ACM Symposium on Applied Computing (SAC '07)*, pages 914–919, 2007.
- [36] R. K. Mustafa Al-Leddawi. Reviewing security and privacy aspects in combined mobile information system (CMIS) for health care systems. Master's thesis, Blekinge Institute of Technology, 2007.
- [37] H. S. Ng, M. L. Sim, C. M. Tan, and C. C. Wong. Wireless technologies for telemedicine. *BT Technology Journal*, 24(2):130–137, 2006.
- [38] M. P. Papazoglou, B. J. Kramer, and J. Yang. Leveraging web-services and peer-to-peer networks. In *Proceedings of the 15th Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, pages 485–501, 2002.
- [39] C. Pattichis, E. Kyriacou, S. Voskarides, M. Pattichis, R. Istepanian, and C. Schizas. Wireless telemedicine systems: an overview. *IEEE Antennas and Propagation*, 44(2):143–153, Apr 2002.
- [40] G. Peng, S. Li, H. Jin, and T. Ma. Dynamic indices for mobile peer-to-peer networks. In *Proceedings of the Fourth International Conference on Computer and Information Technology (CIT '04)*, pages 410–415, September 2004.
- [41] G. Peng, S. Li, H. Jin, and T. Ma. M-CAN: a lookup protocol for mobile peer-to-peer environment. In *Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks*, pages 544–549, May 2004.
- [42] D. Ratner, P. Reiher, G. J. Popek, and G. H. Kuenning. Replication requirements in mobile environments. *Mobile Networks and Applications*, 6(6):525–533, 2001.
- [43] T. Repantis and V. Kalogeraki. Data dissemination in mobile peer-to-peer networks. In *Proceedings of the 6th ACM International Conference on Mobile Data Management (MDM '05)*, pages 211–219, 2005.
- [44] SETI@home Project, <http://setiathome.berkeley.edu>. *SETI@home*.

- [45] Skype Technologies S.A., <http://www.skype.com>.
- [46] R. Smith. What clinical information do doctors need? *British Medical Journal*, 313(7064):1062–1068, 1996.
- [47] Sun Microsystems, <http://java.sun.com>. *JAVA Homepage*.
- [48] H. Takahiro and S. K. Madria. Data replication for improving data accessibility in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(11):1515–1532, 2006.
- [49] D. Tsoumakos and N. Roussopoulos. A comparison of peer-to-peer search methods. In *Proceedings of the International Workshop on the Web and Databases (WebDB)*, pages 61–66, June 2003.
- [50] U. Varshney. Patient monitoring using infrastructure-oriented wireless LANs. *International Journal on Electronic Healthcare*, 2:149–163, 2006.
- [51] U. Varshney and S. Sneha. Patient monitoring using ad hoc wireless networks: reliability and power management. *IEEE Communications*, 44(4):49–55, April 2006.
- [52] O. Wolfson, B. Xu, and R. Tanner. Mobile peer-to-peer data dissemination with resource constraints. In *Proceedings of the International Conference on Mobile Data Management*, pages 16–23, May 2007.
- [53] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. Technical Report 2001-47, Stanford University, October 2001.