The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

DESIGN AND ANALYSIS OF WIRELESS AND MOBILE NETWORKS

A Thesis in

Computer Science and Engineering

by

Sunho Lim

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2005

The thesis of Sunho Lim has been reviewed and approved* by the following:

Chita R. Das
Professor of Computer Science and Engineering
Thesis Adviser
Chair of Committee

Thomas F. La Porta
Professor of Computer Science and Engineering

Wang-Chien Lee
Associate Professor of Computer Science and Engineering

Guohong Cao
Associate Professor of Computer Science and Engineering

Carleen Maitland
Assistant Professor of Information and Science Technology

Raj Acharya
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

* Signatures are on file in the Graduate School.

Abstract

With the increasing use of wireless networks as a ubiquitous communication infrastructure, design of efficient wireless networks has become a recent research focus. In addition, growing interest in accessing the wired network or Internet services anytime and anywhere has driven the developments of mobile ad hoc networks (MANETS), which can be used in many realistic applications. The critical design issues for wireless and mobile networks include provisioning of seamless communication with Quality-of-Service (QoS) guarantees, high service accessibility, reliable data transfer, low power consumption, and high communication performance. However, limited bandwidth and battery power, user mobility, and changing network topology make the design space much more complex and challenging.

The overall objective of this research is to design and analyze wireless and mobile networks that can provide QoS guarantees and high communication performance. In particular, we investigate four related research issues. First, a unified approach for QoS provisioning in cellular networks is proposed to provide improved and predictable performance. A sector-based differential bandwidth reservation policy and a QoS-aware admission control scheme are suggested. Second, we investigate the design of Internet-based mobile ad hoc networks (IMANETS) for providing universal data accessibility. In this context, an aggregate cache management scheme and an information search algorithm are proposed to improve the data accessibility and communication performance in IMANETS. Next, since updating a cached or replicated data item is a critical issue for supporting caching in mobile wireless networks, we investigate several push and pull-based cache invalidation strategies for Imanets. A global positioning system (GPS) based connectivity estimation scheme is suggested as a base algorithm to support any cache invalidation strategy. Finally, we investigate a randomized communication scheme, by integrating the IEEE 802.11 power saving mode (PSM) and dynamic source routing (DSR), to improve energy efficiency without compromising communication performance in Manets. The advantages of these techniques are demonstrated through extensive simulation.

Table of Contents

List of Tables

List of Figures

Acknowledgments

Over the past couple of years, I spent the most exciting, challenging, surprising, and enjoying times in my school life in the Pennsylvania State University, University Park. A young man, who came to State College alone with nothing but iron-will, has become a husband of a woman, a father of a daughter, and a professor of an academy that I really want to be. First of all, I would like to thank you to God who has been always with me and my family, blessed us, and kept us safe.

This thesis owes very much to my Ph.D. advisor, Dr. Chita Ranjan Das, and I would like to thank you from the bottom of my heart for his priceless guidance, advice, and encouragement during entire graduate school period. He not only trained me to be a professional but also took care of me like a family member. He led me to be open-minded and flexible for the way of thinking. I learn from him about how to become an excellent researcher as well as a thoughtful advisor.

I am also grateful to my Ph.D. committee members, Dr. Thomas F. La Porta, Dr. Wang-Chien Lee, Dr. Guohong Cao, and Dr. Carleen Maitland, for their valuable comments. When I was preparing a conference paper, Dr. Lee gave me three pages of review for an initial version that I will never forget. Based on his precious review, our paper was finally nominated as one of best paper candidates. Also, thankfully Dr. Chansu Yu is willing to co-work with me with his energetic research guide and insightful comment, and upgrades my research potential.

In addition, Dr. Soohoan Chae, my M.S. advisor, has been a great moral supporter for finishing Ph.D. degree. My M.S. committee members, Dr. Soochan Hwang, Dr. keunghae Lee, Dr. Sungdo Chi, Dr. Dongho Song, Dr. Jongsou Park, was also a great help for preparing study abroad.

The research projects which consist of this thesis have been supported in part by NSF grants MIPS-9634197, CCR-9900701, CCR-0098149, CCR-0208734, CCF-0429631, and EIA-0202007, and equipment grants from NSF and IBM.

It always pleasure to remind the colleagues who were in our research group (High Performance Computing Laboratory), Dr. Andy Yoo, Dr. Aniruddha S. Vaidya, Dr. Shailabh Nagar, Dr. Suneuy Kim, Dr. Mangesh Kasbekar, Dr. Kihwan Yum, Dr. Eunjung Kim, Dr. Nabil J. Sarhan, and Dr. Xidong deng for their helpful discussions. I enjoy spending time with current members Dr. Sungwon Syi, Gyusang Choi, Jinha Kim, Deniz Ersoz, Jongman Kim, Dongkook Park, and Younghyun Oh.

It also a good memory for me to study with Korean colleagues in Dept. of Computer Science and Engineering, Dr. Changsik Park, Hongseok Kim, Jonghwan Shin, Dr. Hwanho Kim, Dr. Hyunsuk Kim, Dr. Jincheol Yoo, Daegyu Lee, Ingyu Lee, Dr. Soontae Kim, Dr. Seungtaek Park, Dr. Junghye Min, Dr. Jinhyeong Park, Yoonchan Jhi, Byungwon On, Yoojin Hong, Heesook Choi, Jaesheung Shin, Seungwoo Son, Youngjae Kim, Seongah Choi, and Jieun Jeong.

I cannot forget to thank you to brothers and sisters in Penn State Korean Catholic Community, and people in Penn State Korean Student Association for their heart warming help. Also, I would like to thank you to visiting scholars, Dr. Kyungho Park, Dr. Wonyoung Yang, and Dr. Jaeok Jang, for their sincere advice.

I dedicate this thesis to my father, *Soonkwan Lim*, my mother, *Soonja Cho*, father-in-law, *Kwonsoo Park*, and mother-in-law, *Kijung Sung*, who always trust, support, and pray for us. I cannot thank you enough to my uncle, Jongseok Cho, and to my wife's uncles, Kichun Sung, Kijoong Sung, and Kibyung Sung for their encouragement and support. Also, I really appreciate to Sunbin Lim, Jooyoung Jung, Sunwha Lim, Jingone Seo, Sangjun Park, and Hyojin Park for their encouragement. In addition, I am grateful to all the other family members, who are not listed here, for their silence trust and help.

Last but not the least, I would like to express my most sincere gratitude to my lovely wife, *Sujin Park*, for her endless support, pray, and love, and my adorable daughter, *Seungwon Sophia Lim*, for her smiles which are always a source of energy. I love them so much and they are the reason why I live for.

Now, I am going to join with a University as a professor. Although I feel a little nervous and exciting for a real world, I believe that my past teaching and research experiences have prepared and trained me well not only as an instructor, but also as an academic consultant and advisor. Please keep warm eyes on us as you always have done so.

To my lovely wife and daughter

Chapter 1

Introduction

With the recent advent in wireless technologies and mobile devices, wireless networks have become a ubiquitous communication infrastructure. In addition, growing interest in accessing the wired network or Internet has fueled the development of mobile wireless networks, which can be used in many realistic applications. It is envisaged that in the near future, users will be able to access the Internet services and information anytime and anywhere. To realize this vision, however, provision of seamless communication with a certain level of Quality-of-Service (QoS) guarantee, reliable data transfer and service access, low power consumption, and high communication performance are some of the challenging issues that need to be addressed.

Before discussing the specific research issues, we start with a brief description of the motivation of this thesis. In the context of general wireless systems, QoS guarantees to on-going and new connections are becoming critical issues in designing such systems. Since users are expected to move around during communication sessions, one of the most important QoS factors is related to a handoff, which is a mechanism of transferring an on-going connection from the current cell to the next cell to which a mobile terminal[1] (MT) moves. With regard to the handoff, two supplementary techniques have been mainly used for controlling the connection dropping rate (CDR) and connection blocking rate

---

[1] In this thesis, we use the term mobile terminal (MT) to refer to a portable device (e.g. laptop, personal digital assistance (PDA), and mobile phone) or a person who carries it.

(CBR) - bandwidth reservation and admission control. In most prior studies, bandwidth reservation [18, 20, 25, 28, 43, 64, 77, 81, 94] and admission control [9, 49, 58] have been treated orthogonally. However, by integrating both these techniques, it may be possible to provide improved and predictable performance in cellular networks.

For a ubiquitous communication infrastructure, an MT may still have difficulty to connect to a wired network or Internet due to limited wireless bandwidth and accessibility. Under heavy traffic, an MT has to content for bandwidth and may get blocked from a wireless base station. Moreover, in some geographically remote areas, an infrastructure may not even be available. Thus, researchers in the academic and industry are exploring an alternative technology, called Mobile Ad Hoc Network (MANET), for its low cost and ease of deployment. However, research on MANET has primarily focused on developing routing protocols to increase connectivity among MTs in a constantly varying topology [30, 34, 38, 59, 60, 61, 62, 70]. Due to the users' interests in accessing the Internet, it is an important requirement to consider the integration of MANET with the Internet. Thus, to put the MANET technology into the context of real life, we consider an *Internet-based* MANET (IMANET), which is an evolving communication infrastructure that combines the wired Internet and wireless mobile ad hoc networks. IMANET is getting more attention and is applied to realistic Internet applications because of its flexible accessibility and information availability. For instance, in museums or shopping malls, users may need to access various types of information provided by a local service provider through an electronic guide such as an info-station [92]. Users can form an ad hoc network and share information directly or indirectly by accessing the info-station through selected access points to the Internet. In this research, we investigate mechanisms for efficient information search and date access in IMANETS by employing data caching as a viable alternative. To the best of our knowledge, none of the previous work has examined use of caching in the realm of IMANET.

Although caching or replicating frequently accessed data items is an effective technique to enhance the communication performance in a mobile environment, it also needs to consider a critical design issue, *cache invalidation*, for applications requiring data consistency with the sever. The cache invalidation problem has been well studied in mobile computing environments [11, 12, 32, 37, 40, 45, 68, 72, 90]. However, most prior cache invalidation schemes [11, 12, 32, 37, 90] are limited to cellular wireless networks, in which the MTs only communicate with the base stations (BSs) but not with each other. Since the cache invalidation problem in a mobile wireless network is unique from that in a cellular network due to multi-hop message relay, operation cost model, and uncertainty in message delivery, prior techniques can not be directly applied to an IMANET. In this thesis, we investigate and analyze several push and pull-based cache invalidation strategies for IMANETS.

In a wireless mobile environment, one of the most critical issues that affects the communication performance is energy. Since an MT is operated on a battery power, an efficient power saving mechanism is essential to guarantee a certain amount of device lifetime. It also directly affects the network lifetime in a MANET because MTs themselves collectively form a network infrastructure for routing. While there has been active study on multihop networks with respect to many aspects including energy conservation [7, 16, 27, 39, 41, 44, 76, 88, 89], there is little effort about how to integrate the well-known IEEE 802.11 power saving mode (PSM) with a multihop routing protocol such as dynamic source routing (DSR). In this thesis, we investigate the integrating of the IEEE 802.11 PSM with the popular DSR protocol, and propose a new communication mechanism, called *RandomCast* or *Rcast*, for MANETS.

The main motivation of the research is to design and analyze various issues in wireless and mobile networks. The research includes development of QoS provisioning in cellular networks, cache management in IMANETS, cache invalidation protocol in IMANETS, and randomized communication mechanism for MANETS. An overview of the thesis is given below.

1.0.1 Thesis Overview

The research investigates the following issues:

- *A unified approach for QoS provisioning in cellular networks*: To provide improved and predictable performance in cellular networks, we suggest an integrated approach that combines bandwidth reservation and admission control. In this study, we develop a *Differential Bandwidth Reservation (DBR)* scheme that uses a *sector-type* configuration to reserve/share bandwidth along the path of an MT, and a *QoS-Aware Admission Control* scheme using the differential bandwidth reservation policy.

- *A Cache management scheme for Internet based Mobile Ad Hoc Networks (*IMANETS*)*: To combine MANETS with the wired network or Internet, we consider the IMANET infrastructure and investigate the problem of information search and access under this environment. A broadcast-based *Simple Search (SS) algorithm* and an *aggregate caching* mechanism are developed for improving information accessibility and reducing communication latency in IMANETS. A cache admission control policy and a cache replacement policy, called *Time and Distance Sensitive (TDS) replacement*, are developed to reduce the cache miss ratio and improve overall information accessibility.

- *Cache invalidation Strategies in* IMANET: For providing data consistency of the aggregating caching scheme in a mobile environment, we investigate several push and pull-based cache invalidation schemes for IMANETS. In order to adapt and enhance these schemes for IMANETS, we develop a global positioning system (GPS) based connectivity estimation (GPSCE) scheme. In addition, we extend the simple search algorithm to facilitate the deployment of cache invalidation schemes. With GPSCE and the modified search protocol, we analyze four cache invalidation schemes for IMANETS: an aggregate cache-based on demand (ACOD) protocol, a modified timestamp (MTS) protocol, an MTS with update invalidation report (MTS+UIR) protocol, and an asynchronous (ASYNC) protocol.

- *A randomized communication mechanism for* IMANETS: To support energy-efficiency communication in MANETS, we integrate the IEEE 802.11 PSM, in which MTs consistently operate in the power saving (PS) mode, and the DSR routing protocol which is used as the network layer protocol, a novel communication scheme, called *RandomCast* or *Rcast*.

The rest of this thesis organized as follows. Chapter 2 discusses a unified approach for QoS provisioning in cellular networks. A cache management scheme for IMANETS is discussed in Chapter 3. This is followed by the discussion on the design of cache invalidation schemes, and the energy-aware communication scheme for Imanets in Chapter 4 and 5, respectively. Finally, the thesis is summarized in Chapter 6 along with a few research directions.

Chapter 2

A Unified Approach for QoS Provisioning in Cellular Networks

In this chapter, we develop a unified framework to provide QoS guarantees to on-going connections in cellular networks.

2.1 Introduction

As cellular networks are envisioned to be an ubiquitous communication infrastructure that will include a variety of mobile devices, provisioning of seamless communication as well as Quality-of-Service (QoS) guarantees to on-going and new connections are critical issues in designing such networks. Since a myriad of multimedia applications are expected to be serviced by cellular networks, managing the limited resources in a mobile environment brings in new design challenges. Specifically, due to the movement of users during communication sessions, one of the most important QoS factors is related to handoff, which is a mechanism of transferring an on-going connection from the current cell to the next cell to which a mobile terminal (MT) moves. A handoff, however, could fail due to unavailability of sufficient bandwidth in the destination cell. As the number of mobile terminals (MTs) increases, the probability of connection handoffs, and hence being dropped due to insufficient bandwidth during the lifetime of a connection could be high.

The connection dropping rate (CDR) can be reduced by reserving some bandwidth solely for handoffs. However, the connection blocking rate (CBR) of new connections may increase due to such bandwidth reservation. Hence, reduction of CBR and CDR are conflicting requirements, and optimization of both is admittedly extremely complex. Two supplementary techniques have been used for controlling the CDR and CBR in cellular networks - bandwidth reservation and admission control. While bandwidth reservation helps in minimizing the CDR of on-going connections, it cannot effectively guarantee a certain level of QoS without an admission control scheme when the traffic load is high. The difficulty in admission control in a mobile environment is that it is not adequate to admit a new call only based on the status of the current cell, where the call is generated. This is because when the MT attempts to move from the original cell to a next cell, there may not be sufficient bandwidth in the destination cell for accepting the handoff. This may result in dropping the call, and increasing the CDR. Therefore, an efficient admission control policy should check the bandwidth availability in the adjacent cells for a smooth handoff. Although dropping the handoff of an on-going connection is considered more objectionable than blocking a new connection, designing a system with zero CDR is practically impossible. Hence, most admission control policies attempt to provide an acceptable CDR, called target QoS for CDR or T_{QoS}, of 1% or 2%.

Most prior bandwidth reservation schemes [43, 64] were based on handoff prioritization, where each cell reserves a fixed bandwidth or dynamically adjustable amount of bandwidth exclusively for handoffs. Other prioritizing schemes [28, 81] either allow the handoffs or new connections to be queued until enough bandwidth is available in a cell. Also, several distributed channel allocation algorithms have been proposed to increase channel reuse to meet the increasing demand for wireless communication [14, 23]. In these studies, a new connection is blocked when a base station (BS) does not have available bandwidth or cannot borrow a channel from neighboring BSs. These schemes do not use any explicit admission control for satisfying a given QoS. Recently, a dynamic channel allocation scheme is integrated with an adaptive handoff mechanism [13] to improve the bandwidth utility and guarantee a certain level of QoS. However, the dynamic channel allocation schemes incur high message complexity when the traffic load is high.

Instead of only considering the current cell, where a new connection is initiated, admission control schemes [9, 49, 58] consider the status of a number of cells, which are located around or along the path in which the MT might move, and then make an admission decision based on the agreement among all cells or a subset of cells, respectively. For example, Naghshineh et al. [58] suggested a distributed admission control scheme, where the current cell and all its immediate adjacent cells are considered to decide whether a new connection should be accepted or not. The shadow cluster concept introduced in [49] estimates the future resource requirements and considers a set of cells located around an active MT. However, these schemes are based on strong assumptions such as precise knowledge about handoff and connection termination, as well as the MT's mobility pattern. Aljadhai et al. [9] defined a most likely cluster (MLC), which is a set of cells to which an MT is likely to move with a higher probability during its lifetime. The shape of the MLC and the number of cells in the MLC are determined based on the moving speed and direction of the MT. Since all cells or a fraction of the cells in the

MLC need to reserve bandwidth before admitting a new connection, the approach may either waste a large amount of bandwidth or may not well adapt to the T_{QoS}.

To keep the CDR below a predefined level, several bandwidth reservation schemes have been suggested using effective prediction mechanisms [18, 20, 25, 77, 94]. The global positioning system (GPS) technology [66] can be used to provide accurate position, velocity, and direction of an MT in real time. It is already commercialized and installed in a wide range of devices such as cellular phones, personal digital assistants (PDA), and wireless interface cards. Chiu et al. [18] and W. Sho et al. [77] proposed to predict an MT's moving direction and reserve bandwidth dynamically based on the accurate handoff prediction. To increase the accuracy of an MT's mobility, the size of GPS units is kept small. In addition, these techniques require frequent communication between an MT and the satellite. An aggregated moving history of an MT is used to estimate the MT's handoff behavior [20]. Based on the prediction, the number of reserved bandwidths is calculated. This scheme also incurs high communication overhead for providing accurate prediction of an MT's movement.

Similar resource reservation and admission control schemes [47, 54, 80] have been applied to packet-switched cellular networks to provide QoS support for high-speed multimedia applications such as video conferencing, digital library, and video-on-demand (VOD). Lu et al. [54] proposed an adaptive QoS design scheme based on a revenue-based resource adaptation and bandwidth reservation, where the next cell prediction for a handoff was deployed. Talukdar et al [80] suggested an admission control, derived from a measurement-based admission control scheme [36], to increase utilization of resources for real-time applications.

In most prior studies, bandwidth reservation and admission control have been treated orthogonally. However, it is essential to combine both these techniques to provide improved and predictable performance in cellular networks. In this chapter, we propose such an integrated approach. We propose a *Differential Bandwidth Reservation (DBR)* scheme that uses a *sector-type* configuration to reserve/share bandwidth along the path of an MT. The size of the sector and the number of cells in the sector can be dynamically configured for each MT using its mobility information. The sector of cells are further divided into two regions, called *inner* and *outer sectors* depending on whether they have an immediate effects on the handoff or not. While bandwidth reservation is used in the inner sector, bandwidth sharing is used in the outer sector to improve the effectiveness of resource sharing. We enhance the performance of the DBR algorithm with a *User Profile-based DBR (UPDBR)* policy that relies on the known mobility pattern of the MTs for better path prediction.

In addition, a *QoS-Aware Admission Control* scheme using the differential bandwidth reservation policy is proposed. The admission control algorithm first uses the bandwidth reservation algorithm to check if bandwidth reservation can be done in appropriate cells to admit the new call. In contrast to most prior schemes, not all the cells in the sector are required to reserve or share the bandwidth for satisfying the QoS requirement. The number of cells involved in admission control can be changed dynamically depending on the average CDR of the cells in the sector and that of the current cell, where a new connection is generated. The novelty of the proposed admission control mechanism is that it is adaptable to the mobility pattern of the MTs in terms of the sector size, number of cells in the sector, and specific QoS parameters.

We simulate a (6×6) wrap-around network of hexagonal cells to evaluate the effectiveness of the proposed DBR & admission control mechanisms. We compare our integrated technique with two related schemes, called STATIC [64] and Predictive Timed QoS guarantees (PT_QoS), which is based on the MLC concept [9]. Both voice and data traffics are used for performance evaluation. Simulation results indicate that our scheme is more adaptable to provide a certain level of CDR guarantee compared to the prior schemes. In particular, it guarantees the specified CDR over the entire workload, while maintaining a competitive CBR. The UPDBR scheme can exploit the path history for better bandwidth utilization as well as reduction in the number of communication messages compared to the DBR. In addition, we study the impact of the cell size, the number of cells involved in making admission decisions and other system parameters in satisfying the overall QoS parameters.

The rest of this chapter is organized as follows. We introduce the system model in Section 2.2 along with the proposed bandwidth reservation and admission control policy. Section 2.3 is devoted to performance evaluation and comparison of the algorithms. Finally, the last Section provides the concluding remarks.

2.2 A Unified Bandwidth Reservation and Admission Control Mechanism

In this section, first we present the system model, where a set of cells is divided into a couple of clusters in the form of a sector. Based on this model, we propose a differential bandwidth reservation (DBR) policy and then an admission control scheme that utilizes the DBR scheme.

In the proposed DBR approach, two different reservation policies are applied to two different regions of cells in the sector, $R(\theta)$. All cells in the sector participate in bandwidth reservation, and a new connection is accepted only if the current cell has enough available bandwidth and all (or part of) participating cells are ready to reserve or share the bandwidth. In addition, we consider the known mobility pattern of the MTs for better path prediction in implementing the differential bandwidth reservation policy.

The proposed admission control mechanism attempts to keep the CDR below a target QoS value by judiciously using our bandwidth reservation algorithm. A BS is assumed to have some knowledge of the moving pattern of a new call. Using this information, the BS checks if a new bandwidth reservation in the appropriate cells can be done to accommodate the new call.

2.2.1 System Model

In cellular networks, each cell is supported by a BS located in the center of the cell. The BSs are connected to each other by a static wired network. Each MT in a cell communicates with the BS by wireless links. Each cell is assigned a fixed number of channels (or bandwidth). An MT may make a connection request from anywhere in any cell. Since an MT tends to keep its original moving direction, it has a lower probability of making a sudden turn compared to maintaining the current direction. Therefore, let us assume that an MT moves straight, left, right, lower left, lower right, and back with probabilities of P_S, P_L, P_R, P_{LL}, P_{LR}, and P_B, respectively. Further, we assume that the MT moves with a constant speed in a cell and changes (or may not change) direction at the cell boundary.

We assume that a BS is capable of predicting the direction of a new connection based on its path history, which is recorded at the MT. Typically, a BS periodically broadcasts control messages to the MTs in its area for location and bandwidth management. A BS's *id* consisting of a few bits can be easily added to the control message without much overhead. Usually, an MT stays in the *power-saving* mode to reduce battery consumption when it is idle. However, it still listens to the control channel. Whenever the MT moves into a new cell, it identifies the BS *id* using control messages, and caches the *id* in its memory. When the MT makes a new call, it sends the list of *ids* with the connection request message to the BS. The BS can predict the next cell to which the MT might move from the path *(id)* list. Similar assumptions have been used in prior research [9, 79]. More accurate moving directions can be obtained by using the GPS [66].

Since an MT has a much higher probability of maintaining its current direction, or moving left, or moving right than the other three directions, there exists a sector of cells covered within an angle (θ) to which the MT might move in the near future. This is similar to the MLC concept [9]. This sector of cells can be classified based on their distance from the current cell. Let $r_{i,d}$ denote the cluster of cells located at a distance[1] d to which the MT might move with a higher probability from the current cell c_i. Let $R(\theta)$ denote the sector of cells located within θ. Then, we have:

$$r_{i,d} = \{c_j \mid \mathcal{N}(\text{distance}(c_i, c_j)) = d \quad \wedge \quad c_j \in R(\theta)\}, \quad \text{where } d = 1, 2, \cdots, n. \quad (2.1)$$

---

[1] In this chapter, we use distance (d) as the normalized distance, $\mathcal{N}(\text{distance}(c_i, c_j))$, between the current cell (e.g. c_i), where the MT is located and the center of the other cell (e.g. c_j).

As shown in Figure 2.1, cluster $r_{i,1}$ consists of three cells, c_1, c_2, and c_3, while cluster $r_{i,4}$ has 7 cells, c_{12} to c_{18}. As the distance d increases, the number of cells in that cluster increases. Depending on the moving speed and mobility pattern of MTs, the number of cells in a cluster and the maximum cluster distance n should be carefully chosen. Also, the sector angle θ is likely to affect the system performance. For the sake of simplicity, we assume that all the cells in the sector are under the same switch such as a mobile station center (MSC), as used in [63].

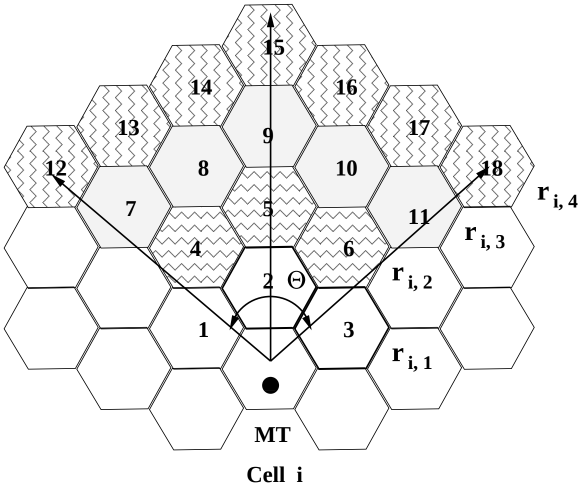


Fig. 2.1. Classification of a sector of cells into different clusters based on their distance from the current cell.

2.2.2 A Differential Bandwidth Reservation Algorithm

Bandwidth reservation is essential for seamless handoff of an MT from one cell to the next cell. When making bandwidth reservations, it is more effective to consider a cluster of cells, which includes cells located around or along the way to which the MT might move. Furthermore, there should be different bandwidth reservation policies for different cells depending on their distance from the current cell. For example, suppose an MT in a cell c_i is likely to perform a handoff. BS_i can predict the direction in which the MT might move based on the path history of the MT, and then construct a sector of cells for the bandwidth reservation. It is easy to see that the MT will most likely move to the cells closer to c_i, such as the cells in $r_{i,1}$ and $r_{i,2}$ in Figure 2.1. Since the MT may change its direction at any time, it has a relatively lower probability of moving to cells located far away from the cell c_i, such as the cells in $r_{i,4}$. Based on this observation, for each cell c_i, cells that are required for bandwidth reservations are classified into two different regions: $R_I(i)$ and $R_{II}(i)$.

- $R_I(i)$: It contains the cells located closer to c_i. For example, cells in $r_{i,1}$ and $r_{i,2}$ are close to c_i and they have a direct impact on CDR because the handoff will fail if the requested bandwidth is not available.
- $R_{II}(i)$: It contains the cells further away from c_i. For example, cells in $r_{i,3}$ and $r_{i,4}$ are 3 and 4 cell distant away from c_i, and they only have indirect effects on the handoff because the MT may not move to those cells.

These regions are referred to as the *inner* and *outer* regions, respectively. For each handoff request, a BS reserves or shares bandwidth of the cells in both regions.

Since several BSs could request reservations for the corresponding handoff requests, a cell c_m may be involved in multiple reservations from different BSs. As an example, consider two handoff requests from two cells c_i and c_j. BS_i and BS_j make bandwidth reservations in their corresponding regions of cells; i.e., $R_I(i)$ and $R_{II}(i)$, and $R_I(j)$ and $R_{II}(j)$. As a result, c_m can be included in $R_I(i)$ or $R_{II}(i)$, and in $R_I(j)$ or $R_{II}(j)$. In this case, c_m reserves bandwidth for handoff requests from c_i and c_j. The differential bandwidth reservation policies for the two regions (inner and outer) are described next.

Prior to initiating a handoff, BS_i first constructs the sector configuration and broadcasts the region information along with the reservation request to the corresponding cells. When BS_m receives a handoff request from BS_i, it already knows its reservation region. If c_m is in $R_I(i)$, BS_m makes bandwidth reservation using the condition:

$$N_{used}(m) + bw \leq N_{total} - N_{resv}(m). \tag{2.2}$$

where $N_{used}(m)$ is the currently used bandwidth, *bw* is the requested bandwidth, N_{total} is the total bandwidth of the cell, and $N_{resv}(m)$ is the currently reserved bandwidth for handoffs. If the above condition is not satisfied, BS_m checks if it can *share* the already reserved bandwidth with other cells. For bandwidth sharing, the reserved bandwidth is not strictly assigned to any particular MT, and hence, can be used by any MT on a first come first serve (FCFS) manner. To implement this idea, BS_m checks if the sum of $N_{resv}(m)$, *bw*, and the existing shared bandwidth $(N_{share,I}(m) + N_{share,II}(m) * \frac{\epsilon}{\eta})$ is less than or equal to ϵ times of $N_{resv}(m)$. Note that the shared bandwidth could be different in the inner and outer regions, and they are specified by two different terms.

We use a weight factor ($\frac{\epsilon}{\eta}$), which is less than 1.0 for $N_{share,II}(m)$, because the MT has a lower probability of moving to this region. Here, ϵ is a system parameter, which enables BS_m to reserve more than the actually allocated bandwidth, i.e. $\epsilon \geq 1$ based on the QoS requirement of the system. If any of the two conditions (reservation/sharing) is satisfied, BS_m sends a positive reply (*ack*) to BS_i. Otherwise, the reservation request from BS_i is rejected.

If c_m belongs to $R_{II}(i)$, the reservation may waste a large amount of bandwidth since the MT may not move into c_m. Thus, only bandwidth sharing is applied in the outer region. Specifically, BS_m checks if the sum of $N_{resv}(m)$, bw, and the existing shared bandwidth, $(N_{share,II}(m) + N_{share,I}(m) * \frac{\eta}{\epsilon})$, is less than or equal to η times of $N_{resv}(m)$, and then decides whether it can share the bandwidth or not. In contrast to $R_I(i)$, we use a weight factor ($\frac{\eta}{\epsilon}$), which is greater than 1.0 for $N_{share,I}(m)$ due to its high probability of being used. Similar to ϵ, η is a system QoS parameter, and $\eta \geq 1$. Intuitively, η should be higher than ϵ to facilitate more sharing in the outer region. The pseudo code for the bandwidth reservation is given in Fig. 2.2.

2.2.3 An User Profile-Based DBR Algorithm

We extend the DBR algorithm presented in Section 2.2.2 for MTs whose moving path is known *a priori*. We refer it to as a *User Profile-based DBR (UPDBR)* algorithm.

Several researchers have used user profiles for better location and handoff managements [63, 69, 74, 75]. Sen et al. [74] use profile information to optimize the location management cost. They consider individual user mobility patterns as well as connection arrival patterns, and apply them to determine the transition probabilities between

```
if ($c_m \in R_I(i)$) {
    if ($N_{used}(m) + bw \leq N_{total}$ - $N_{resv}(m)$) {
        send ack to allocate bandwidth to $c_i$;
        $N_{resv}(m) = N_{resv}(m) + bw$;      /* done only after getting confirmation from $BS_i$ */
    }
    else {
        if ($N_{resv}(m) + bw + N_{share,I}(m) + N_{share,II}(m) * \frac{\epsilon}{\eta} \leq N_{resv}(m) \times \epsilon$) {
            send ack to share bandwidth to $c_i$;
            $N_{share,I}(m) = N_{share,I}(m) + bw$;      /* done only after getting confirmation from $BS_i$ */
        }
        else
            send nack to $c_i$;
    }
}
else { /* $c_m \in R_{II}(i)$ */
    if ($N_{resv}(m) + bw + N_{share,II}(m) + N_{share,I}(m) * \frac{\eta}{\epsilon} \leq N_{resv}(m) \times \eta$) {
        send ack to share bandwidth to $c_i$;
        $N_{share,II}(m) = N_{share,II}(m) + bw$;      /* done only after getting confirmation from $BS_i$ */
    }
    else
        send nack to $c_i$;
}
```

Fig. 2.2. The differential bandwidth reservation scheme used in c_m when it receives a reservation request from c_i.

location areas based on a long period of observation about the movements of each MT. Rudrapatna et al. [69] utilize a repetitive travel pattern of MTs such as daily commuters. A statistical bandwidth reservation policy is used for such MTs that may need high QoS. However, none of these studies consider the cell regions, and only use the current cell instead of a sector of cells. Intuitively, if the user profile of a set of MTs can be known, the DBR algorithm can be used more effectively, because the $\mid r_{i,d} \mid$ becomes small, and hence, the number of cells used in the DBR negotiation will be small. This in turn should reduce the communication overhead. In the following, we describe the UPDBR scheme.

In the UPDBR approach, most of the MTs choose their favorite paths, use them frequently whenever they move, and do not change the path unless otherwise required (For example, the path of a daily commuter is typically fixed.). Therefore, we can extract the mobility pattern from the user profile and use it to predict the moving paths of such MTs with better accuracy. For the UPDBR algorithm, we assume that there are two groups of MTs. One is a user profile-based group, denoted as G_f, and the other is a non-user profile-based group, denoted as $G_{\overline{f}}$. The MTs in the G_f group have predictable travel behavior, whereas MTs in the $G_{\overline{f}}$ group exhibit little predictable behavior. The cellular network has the most recent statistical data for each MT in the G_f group. An MT in the G_f group has a probability of Φ to follow the predicted path. Also, the user profile for a particular user in the UPDBR algorithm is assumed to be under the same switch[2] (MSC) as used in [63]. When many switches are involved in retrieving the user

---

[2] In current cellular network standards such as GSM and IS-41 [8, 46], a number of BSs are connected to a base station controller (BSC). The BSC is connected to a mobile station center (MSC), which is attached to the public networks. Two databases, i.e., the visitor location

profile, the proposed technique may incur a longer call set up delay. However, since the BSs are connected through high speed networking, the setup delay may not be too high.

Whenever a new connection is generated in a cell c_i, BS_i checks whether the MT belongs to the G_f or $G_{\overline{f}}$ group. For bandwidth reservation, if the MT belongs to G_f, BS_i establishes $R_I(i)$ and $R_{II}(i)$ in the same manner as the DBR algorithm except that fewer cells are involved in each cluster. If the MT belongs to $G_{\overline{f}}$, BS_i applies the bandwidth reservation policy described in Section 2.2.2. Figure 2.3 shows the difference between the DBR scheme and the UPDBR scheme. For UPDBR, each cluster contains only one cell rather than several cells as in the DBR.

When a new connection request is generated in c_i, BS_i first determines the cells involved in P_{UPDBR} [3], and then it checks available bandwidth using Eq. 2.2. If the condition is satisfied, BS_i establishes $R_I(i)$ and $R_{II}(i)$, and sends reservation request messages to the cells in P_{UPDBR}. The procedure after that is the same as the DBR scheme. When a handoff occurs from a cell c_i to a cell c_j, BS_j checks the available bandwidth for handoff using $N_{used}(j) + bw \leq N_{total}$. In certain situations, an MT may not use the predicted path due to conditions such as traffic congestion or other abnormal

---

register (VLR) and the home location register (HLR) are used to save information about the MTs. The VLR and HLR are connected to the MSC and public networks. These three entities communicate with each other by out-of-band signaling system number 7 (SS7) [46]. The *user profile*, information saved in the VLR and HLR, is a statistical set of data collected for each MT. The collected data records the MT's routes of movement, connection time, and traffic environment based on daily, weekly, and even monthly statistics. Once an MT makes a connection request, the corresponding BS initiates a search for the user profile through the BSC and MSC. The user profile, if available, is retrieved from the VLR or the HLR, and transmitted to the BS through the MSC and the BSC. Since the BS is transparent to the VLR or HLR from the MT's view, we use the terms BS, VLR, and HLR interchangeably in this chapter.

[3] P_{UPDBR} denotes the set of cells involved in the predicted path when using the UPDBR algorithm.

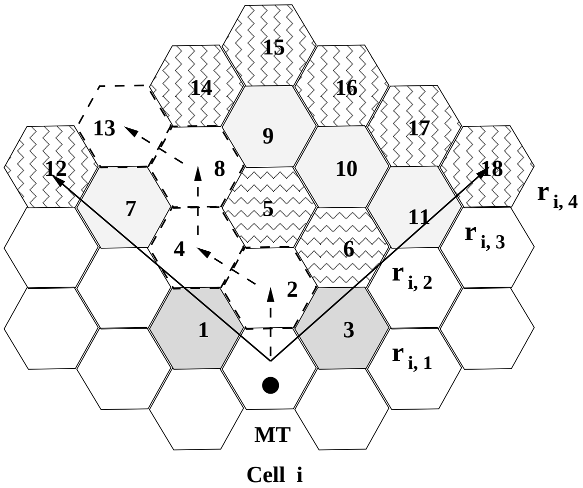


Fig. 2.3. An example of the UPDBR reservation policy. For the original DBR policy, all the cells starting from c_1 to c_{18} may participate in bandwidth reservation. Whereas in the UPDBR approach, only cells c_2, c_4, c_8, and c_{13} are involved in the bandwidth reservation.

events. In such cases, whenever BS_i detects that the MT deviates from the usual path, it transfers the MT from the G_f to the $G_{\overline{f}}$ group.

2.2.4 A QoS-Aware Admission Control Scheme

Next we present an admission control policy for new connections. Whenever a new connection request is generated in a current cell c_i, BS_i first checks if Eq. 2.3 can be satisfied. If the condition cannot be satisfied, the new connection is blocked. Otherwise, BS_i constructs $R_I(i)$ and $R_{II}(i)$ based on the current MT's moving direction. For cells which always follow a similar moving pattern, a look-up table of the moving pattern can be saved and used subsequently to reduce the computation overhead [86]. BS_i then sends reservation request messages to the corresponding cells in $R_I(i)$ and $R_{II}(i)$ to check the feasibility of the handoff, given by Eq. 2.3.

$$N_{used}(i) + bw \leq N_{total} - N_{resv}(i). \tag{2.3}$$

When a cell, say c_j, in $R_I(i)$ or $R_{II}(i)$ receives a reservation request message from c_i, BS_j executes the algorithm shown in Figure 2.2, computes its current CDR (ρ_j), and replies to c_i. After BS_i receives reply messages from all the cells in $R_I(i)$ and $R_{II}(i)$, it computes ρ_i, checks average CDR of all cells in $R_I(i)$ and $R_{II}(i)$, denoted as ρ_{all}^{avg}, and decides whether it can accept the new connection or not.

Let $\mid c_{i,ack} \mid$ and $\mid c_{i,all} \mid$ be the number of cells which sent *ack* messages, and the total number of cells in $R_I(i)$ and $R_{II}(i)$, respectively. Then $c_{i,all}$ is defined as $c_{i,all} = \{c_j \mid c_j \in R_I(i) \;\; \vee \;\; c_j \in R_{II}(i)\}$, and $c_{i,ack}$ is defined as $c_{i,ack} = \{c_j \mid c_j \in c_{i,all} \;\; \wedge \;\; c_j$

has sent *ack* to $c_i\}$. The average CDR is given by $\rho_i^{avg} = \frac{1}{|c_{i,all}|+1} \cdot (\sum_{c_k \in c_{i,all}} \rho_k + \rho_i)$. After receiving all the response, the new connection is accepted if Eq. 2.4 is satisfied:

$$\mid c_{i,all} \mid \cdot \psi_i \ \leq \ \mid c_{i,ack} \mid \quad \wedge \quad \rho_i^{avg} \leq \ T_{QoS}. \tag{2.4}$$

Here, ψ_i is a system parameter maintained in c_i, and can vary dynamically. Once BS_i accepts the new connection, it sends reservation confirm messages to the cells, which sent an *ack* message for bandwidth reservation or sharing. Otherwise, the new connection is blocked. When BS_j receives a confirm message from c_i, it reserves or shares the bandwidth as specified in Figure 2.2.

After BS_i makes a decision for a new connection, it changes ψ_i depending on the ρ_i^{avg} value. If the ρ_i^{avg} is higher than T_{QoS}, then it implies that $c_{i,all}$ currently does not have enough reserved or shared bandwidth for the future handoff to keep the ρ_i^{avg} below T_{QoS}. Therefore, accepting a new connection in c_i can degrade ρ_i^{avg}. BS_i can block a new connection by increasing ψ_i by an amount δ. This implies that BS_i has more cells in $c_{i,all}$ to reserve or share the bandwidth, which in turn lowers the probability of accepting a new connection. However, if the ρ_i^{avg} is less than T_{QoS}, then the system bandwidth in $c_{i,all}$ is underutilized. In this case, BS_i should accept more new connections by decreasing ψ_i by δ. This helps BS_i in accepting a new connection even though less number of cells in $c_{i,all}$ reserve or share the bandwidth. The complete admission control algorithm for a new connection is shown in Fig. 2.4.

```
if $(N_{used}(i) + bw \leq N_{total} - N_{resv}(i))$ {
    send reserve message to $c_{i,all}$;    /* call bandwidth reservation algorithm of Figure 2.2 */
    receive reply from $c_{i,all}$;
    if $(\mid c_{i,all} \mid \cdot \psi_i \leq \mid c_{i,ack} \mid \ \wedge \ \rho_i^{avg} \leq T_{QoS})$ {
        accept the new connection;
        $N_{used}(i) = N_{used}(i) + bw$;
    }
    else
        block the new connection;
    if $(\rho_i^{avg} > T_{QoS})$
        $\psi_i = min(\psi_i + \delta, 1.0)$;    /* $\delta = 0.1$ in our study */
    else
        $\psi_i = max(\psi_i - \delta, 0.1)$;    /* Minimum value of $\psi_i = 0.1$ */
}
else
    block the new connection;
```

Fig. 2.4. The admission control scheme used in c_i.

2.3 Performance Evaluation

2.3.1 Simulation Testbed

We use a (6 × 6) wrap-around cellular network with 1 mile cell diameter to examine the proposed scheme along with two prior schemes [9, 64]. We consider two different types of communication sessions: voice and video. The performance evaluation is conducted using the following assumptions: The average connection time (T) is exponentially distributed with a mean of 180 seconds, and the connection arrival follows Poisson distribution with a rate of λ. Speed of an MT is uniformly distributed between 45 to 70 miles per hour. The total bandwidth (N_{total}) allocated to each cell is 600 kb/s. The bandwidth for audio (B_{audio}) and video (B_{video}) is 10 Kb/s and 100 kb/s, respectively. The percentage distribution of the voice and video connections is given by P_{voice} = 0.8 and P_{video} = 1 - P_{voice} = 0.2. Similar assumptions have been used in most prior studies [9, 20].

We have written an event-driven simulator using CSIM[1] to conduct the performance study. The simulation results are illustrated as a function of the offered load[4] per cell, where offered load is defined as

$$\text{Offered Load} = T \times \lambda \times (P_{voice} \times B_{voice} + (1 - P_{voice}) \times B_{video}). \qquad (2.5)$$

---

[4] The offered load is the total bandwidth required to support all the existing connections in a cell as used in [20, 19, 77, 94]. For example, if the offered load is 100, the system capacity of a cell is fully used to support the connections. Thus, a cell is over-loaded when the offered load is more than 100. We consider a range of offered load from 20 to 300. Since a high connection requests rate could be generated due to special circumstances, we consider a offered load upto 300 to evaluate our scheme.

Note that for a given offered load, we can find the arrival rate λ since all other parameters in Eq. 2.5 are known. For most of the experiments, we use two clusters and the number of cells in the inner and outer cluster is 3 and 3, respectively. Since the probabilities of moving straight (P_S) and moving left or right (P_L, or P_R) are much higher than the other probabilities, we choose $P_S = 0.5$, $P_L = 0.15$, $P_R = 0.15$, $P_{LL} = 0.075$, $P_{LR} = 0.075$, and $P_S = 0.05$. To simulate the user profile-based DBR scheme, we assume that 70% of the traffic in a cell follows the path defined in the user profiles, while the rest (30%) is assumed to be communication sessions with unknown user profiles. Moreover, it is assumed that an MT follows P_{UPDBR} 80% of the time ($\Phi = 0.8$). In the admission control algorithm, ψ is incremented or decremented by δ ($= 0.1$) in our experiment. The simulation parameters are summarized in Table 2.1.

Table 2.1. Simulation parameters.

parameter	value
Total bandwidth per cell (N_{total})	600 kb/s
Bandwidth for audio (B_{voice})	10 kb/s
Bandwidth for video (B_{video})	100 kb/s
Average connection time (T)	180 s
Sharing rate (ϵ or η)	1.0 to 3.0
Target QoS for handoff (T_{QoS})	0.01 or 0.02

2.3.2 Simulation Results

The performance parameters measured in this study are connection blocking rate (CBR), connection dropping rate (CDR), bandwidth utilization, and the number of communication messages. The results are collected using 90% confidence interval and the predicted values lie within ± 10% of the mean. First, we conduct a comparative evaluation of our proposed scheme with respect to two prior schemes and then present in-depth evaluation of our approach.

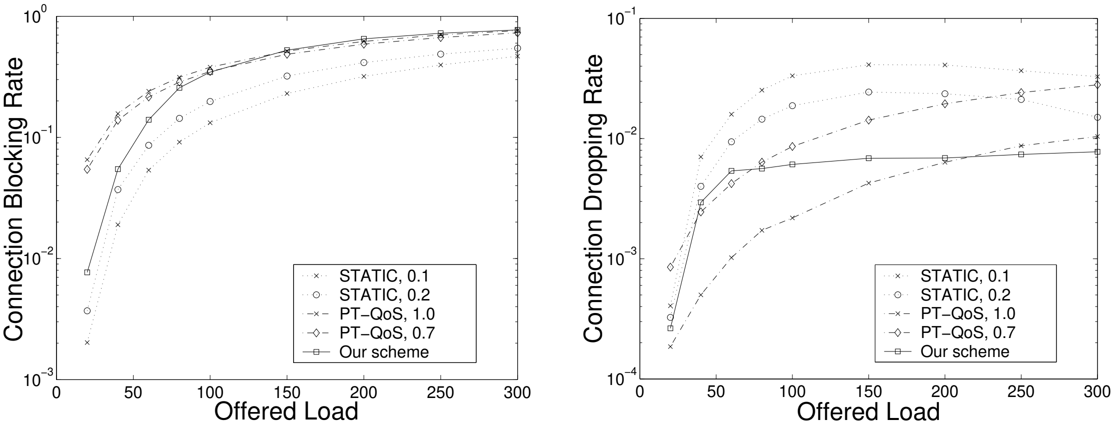
2.3.2.1 Performance Comparison

We start the performance study with a comparative analysis of our approach with the STATIC scheme [64] and the PT_QoS scheme [9]. In case of the *static reservation (STATIC)*, a fraction (g) of the total bandwidth in a cell is exclusively reserved for handoffs. This scheme relies only on the local information to admit a new connection, and the current cell does not negotiate with neighboring cells. We experiment with two different values of g ($g = 0.1$ and $g = 0.2$). The PT_QoS approach [9] defines an MLC (Mostly Likely Cluster) based on the mobility pattern of the MTs, and reserves bandwidths from a fraction of cells (s) in the MLC based on parameters such as the MT's expected arrival time, latest arrival times, and departure time. Unlike our scheme, the PT_QoS algorithm uses a fixed fraction of cells out of all cells in MLC for admission control to admit a new connection, and drops the on-going connection if the MT does not conform the prediction. We also use two different values of s ($s = 1.0$ and $s = 0.7$) for performance evaluation. Both these schemes have no mechanism to support QoS guarantee for handoffs. For the differential bandwidth reservation algorithm, we use one cluster in R_I and one cluster in R_{II}, and the two sharing rates ϵ and η are set to 1.5 and 3.0, respectively.

As shown in Figure 2.5(a), the STATIC approach has the lowest CBR because it only examines the status of the current cell, where the new connection request is initiated. As expected, the static scheme with 20% bandwidth reservation performs better than that with 10% reservation. The PT_QoS scheme has the maximum CBR since it reserves more than the required bandwidth for a longer time, and the results for s = 1.0 and s = 0.7 are almost the same. The proposed scheme has slightly lower CBR than the PT_QoS approach.

Figure 2.5(b) compares the CDRs of the three schemes. As can be seen, the proposed policy keeps the CDR below the T_{QoS} (= 0.01) compared to the other two schemes. An acceptable CDR of 0.01 or 0.02 has been used as a target QoS in prior studies [19, 77]. The PT_QoS approach has a lower CDR than the STATIC scheme except for g = 0.2, which shows a little lower CDR than the PT_QoS approach (for s = 0.7) when the offered load is high. The PT_QoS approach with s = 0.7 has a higher CDR than that with s = 1.0, because the former accepts more connections.

Based on the results of Figures 2.5, we can conclude that the proposed scheme not only provides relatively better performance, but also maintains a stable CDR as specified by T_{QoS} over the entire workload. It has comparable CBR with respect to the STATIC policy. However, the CDR for both STATIC and PT_QoS policies increases with the workload and is higher than the target value of 0.01.



(a) Connection blocking rate

(b) Connection dropping rate

Fig. 2.5. Comparisons of the CBR and CDR of the proposed scheme ($\epsilon = 1.5$ and $\eta = 3.0$) with the STATIC scheme ($g = 0.1$, 0.2), and the PT_QoS scheme ($s = 1.0$, 0.7).

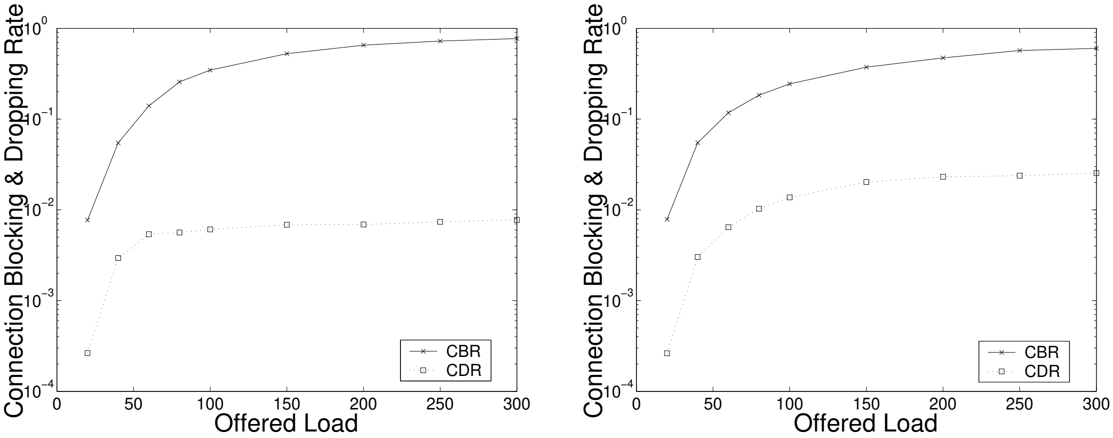
2.3.2.2 Effect of Target QoS

Next, we investigate performance sensitivity of our policy in terms of CBR, CDR, and bandwidth utilization for different QoS settings. Figure 2.6 depicts variations of the CBR and CDR for two target QoS parameters. The results indicate that our integrated scheme keeps the CDR below the target QoS over the entire workload. The CBR, as expected, is slightly higher in Figure 2.6(a) because a BS blocks more new connections to keep the CDR at 0.01.

There is a tradeoff between CDR and bandwidth utilization. Intuitively, if we choose a smaller T_{QoS} value, then more bandwidth will be reserved or shared, CDR will drop, CBR will increase, and the bandwidth utilization will be low. In Figure 2.7, the bandwidth utilization with a smaller T_{QoS} is lower than that with a larger T_{QoS} (= 0.02) and the gap between the two gradually increases with the workload.

In Figure 2.8, we show a snapshot of ψ with time. Note that ψ controls the number of cells that are involved in admission control by incrementing or decrementing its value to keep the CDR below T_{QoS}. The results indicate that at least 60% of the cells participate in the admission decision most of the time.

Since ψ depends on the current average CDR, not all of the cells need to participate in admission control. In Figure 2.9, we plot the average number of cells used in admission control for different workloads. The number of participating cells is higher for a tighter QoS parameter (CDR = 0.01). Also, as the workload increases, we observe a slight decrease in the average number of cells participating in the admission decision because less number of cells are available for bandwidth reservation or sharing. A scheme



(a) $T_{QoS} = 0.01$ (b) $T_{QoS} = 0.02$

Fig. 2.6. Connection blocking and dropping rate with different T_{QoS} values ($\epsilon = 1.5$ and $\eta = 3.0$).

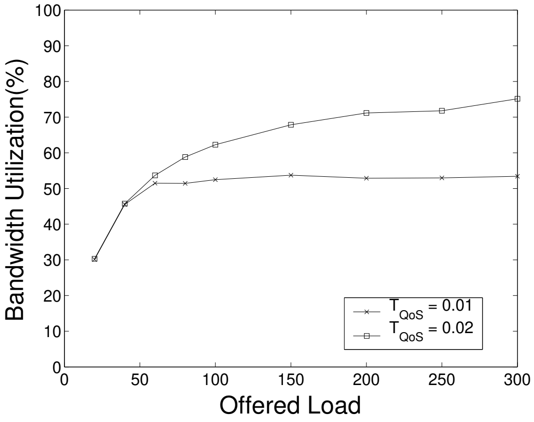


Fig. 2.7. Bandwidth utilization with different T_{QoS} values (0.01 and 0.02).

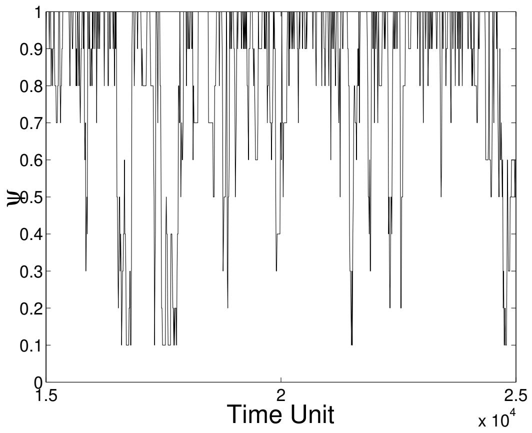


Fig. 2.8. Snapshot of ψ when the offered load and T_{QoS} are 200 and 0.01, respectively.

like PT_QoS uses a fixed number of cells (6 in this case (3 + 3)) irrespective of the target QoS constraint.

2.3.2.3 Effect of the Sector Configurations and System Parameters

Next, we examine the impact of sector configurations and other tunable parameters on the performance of the proposed scheme. Three different sector configurations are used: DBR(1,1), DBR(1,2) and DBR(2, 1), where the notation DBR(x, y) represents x clusters in R_I and y clusters in R_{II}. ϵ is varied from 1.0 to 1.5, while η is varied from 1.0 to 3.0. Note that the DBR approach with $\epsilon = \eta = 1.0$ does not allow any bandwidth sharing, and hence, a connection has a high probability of begin blocked and the dropping rate is almost 0 regardless of the sector configurations. Here, we report results for $\epsilon = 1.5$ and $\eta = 2.0$ or 3.0, respectively. For additional results, please refer to [50]. We use a simple admission control of Eq. 4.1 for comparison.

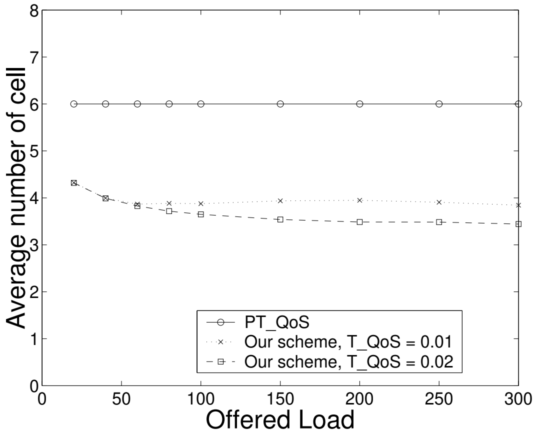
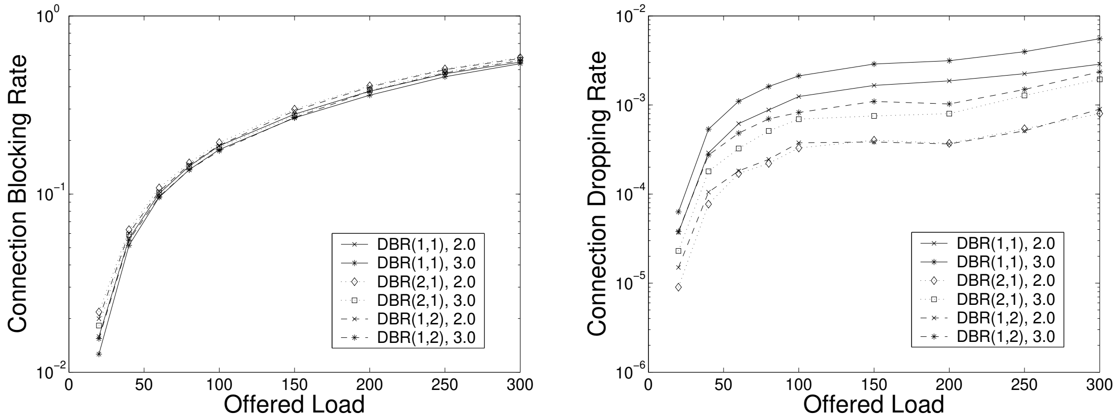


Fig. 2.9. Comparisons of the average number of cell involved in admission control.

Figure 2.10 shows the effect of the sector configurations and system parameters on the CBR and CDR. Figure 2.10(a) depicts that the three DBR configurations have almost similar CBR, although the DBR(1,1) configuration has slightly lower CBR than the other two configurations because it examines a relatively small number of cells during a connection admission control, and thus, the probability of accepting a connection is higher than other sector configurations. In Figure 2.10(b), the CDR increases as the sharing rate (η) increases since it accepts more connections. Because of examining more cells to admit a connection, DBR(2,1) and DBR(1,2) have less probability of dropping a connection than DBR(1,1). With the same sharing rate (η), the CDR increases in the order of DBR(2,1), DBR(1,2), and DBR(1,1). Also, the results indicate that as η increases, CDR increases.

Figure 2.11 plots the bandwidth utilization for different sector configurations and indicates that bandwidth utilization increases as the sharing rate increases. Any sector



(a) Connection blocking rate

(b) Connection dropping rate

Fig. 2.10. CBR and CDR of DBR approach with different sector configurations and sharing rates ($\epsilon = 1.5$, and $\eta = 2.0$ or 3.0).

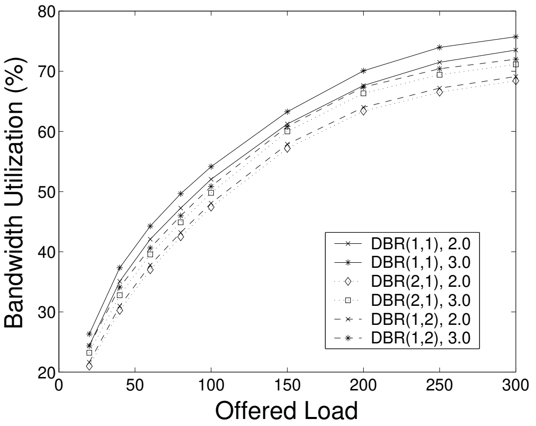
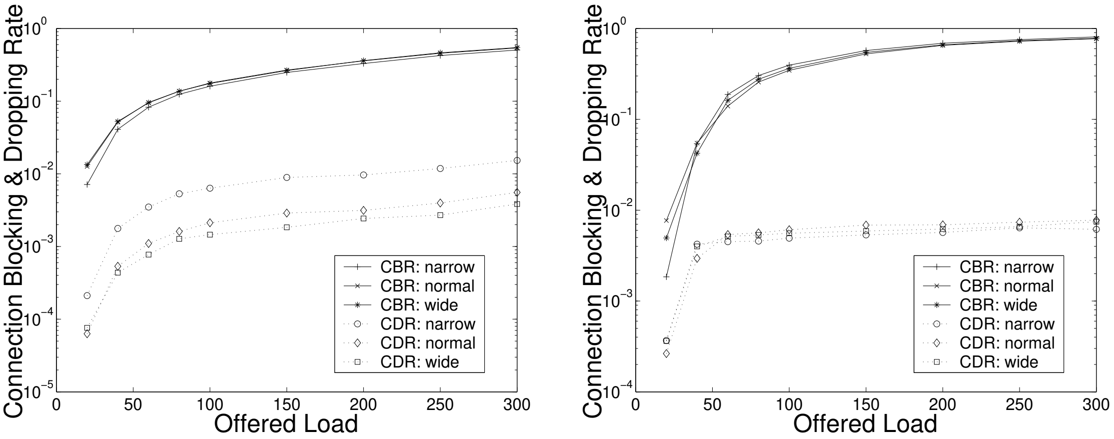


Fig. 2.11. Bandwidth utilization of DBR approach with different sector configurations and sharing rates ($\epsilon = 1.5$, and $\eta = 2.0$ or 3.0).

configuration with $\eta = 2.0$ has lower bandwidth utilization because it shares little bandwidth, and hence, most of the bandwidth is used for handoff reservation. As a result, less bandwidth can be used to accept new connection requests, and a connection may be rejected even though some bandwidth is available. The configurations of DBR(2,1) and DBR(1,2) have relatively lower bandwidth utilizations compared to the DBR(1,1) configuration because DBR(2,1) and DBR(1,2) require more cells to make bandwidth reservations.

To examine the effects of the sector angle θ and hence the different number of cells in each cluster of the sector configuration, we use three sector angles (θ_{narrow}, θ_{normal}, θ_{wide}) and two clusters in a sector. This implies one cluster for each of R_I and R_{II}. For θ_{narrow}, the number of cells used in the first and second clusters is 1 and 3, respectively. θ_{normal}, as has been used in the previous experiments, includes 3 and 3 cells in the inner and outer clusters, respectively. Finally for θ_{wide}, the number of cells used in the first and second clusters is 3 and 5, respectively.

Figure 2.12 shows the CBR and CDR with and without a target QoS for the three sector angles. Neither a target QoS nor a QoS-aware admission control mechanism is used in Figure 2.12(a). The admission decision is made only using the status of the current cell. As θ increases, more cells are involved in each cluster and more bandwidth is reserved or shared. As expected, the CDR increases when the sector angle θ becomes narrow. The CBR exhibits an opposite trend compared to the CDR although the results are much close. Figure 2.12(b) shows the effect of the admission control mechanism. The CDR for all three sector angle configurations are very similar, stable, and are within the specified T_{QoS} value (0.01). This implies that one can use a smaller sector with less



(a) Without T_{QoS}

(b) With T_{QoS} (0.01)

Fig. 2.12. Connection blocking and dropping rate with different number of cells in each cluster.

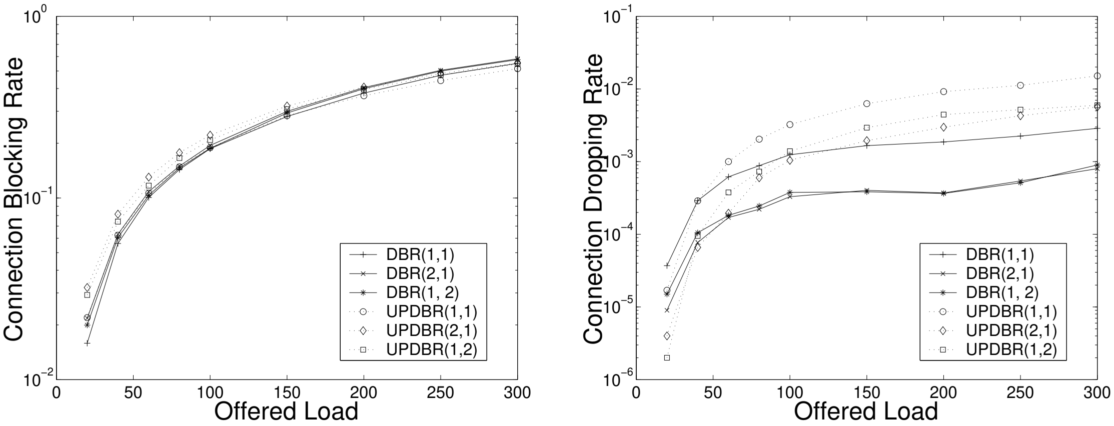
number of cells to satisfy the QoS requirement, and thereby, the other cells can be used for handling additional connections.

2.3.2.4 Effect of User Mobility

Finally, we compare the performance of the UPDBR approach and the DBR approach in terms of CBR, CDR, bandwidth utilization, and communication message overhead. The number of cells in the first, second, and third sectors are confined to 3, 3, and 5 in this study because the probabilities of moving straight (P_S) and moving left or right (P_L, or P_R) are much higher than other probabilities. It was observed that adding more cells to the sectors did not provide additional performance improvements. We neither use a target QoS nor a QoS-aware admission control, but a simple admission control algorithm. A simple admission control decision based on Eq. 2.2 is used for DBR and UPDBR.

The UPDBR approach reserves less bandwidth compared to the DBR approach by making use of the user profile. As a result, more new connections can be accepted. This explains why the UPDBR approach has slightly lower CBR than the DBR approach in Figure 2.13(a). Since the DBR approach reserves more bandwidth for handoffs, its CDR is lower than the UPDBR approach, as shown in the Figure 2.13(b). UPDBR(2,1) and UPDBR(1,2) schemes have lower CDR than UPDBR(1,1) because they use more cells to make bandwidth reservations, and thus, a connection has a lower probability to be dropped if the MT follows the path defined in its user profile.

There is a tradeoff between CDR and bandwidth utilization. Intuitively, if we reserve more bandwidth, the CDR will be reduced, but the bandwidth utilization becomes



(a) Connection blocking rate

(b) Connection dropping rate

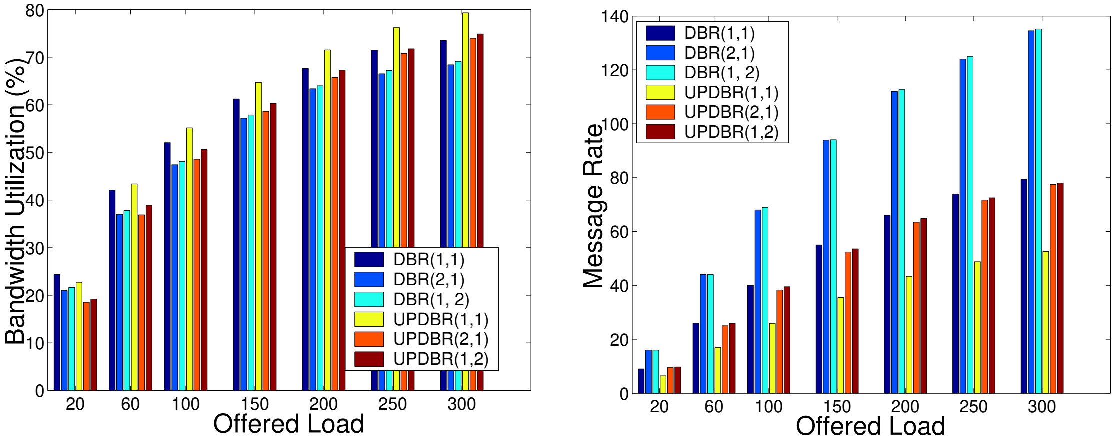
Fig. 2.13. Comparisons of the UPDBR approach to the DBR approach (ϵ =1.5 and η = 2.0).

low. In Figure 2.14(a), the bandwidth utilization of the UPDBR approach is higher than the DBR approach over the entire workload. The DBR approach has low bandwidth utilization since it reserves more bandwidth compared to the UPDBR approach. As explained before, when the number of cells to make bandwidth reservation reduces, the amount of bandwidth, which can be used to support new connections, increases. This explains why UPDBR(1,1) has a higher bandwidth utilization than UPDBR(2,1) and UPDBR(1,2).

As shown in Figure 2.14(b), the UPDBR approach has much less message overhead compared to the DBR approach. Compared to the DBR approach, it reduces the communication messages by almost half when the offered load reaches 300%. In both the UPDBR and DBR schemes, cells need to communicate with each other to make bandwidth reservations. However, in the UPDBR approach, the current cell (where the MT connection request was originated) only sends reservation messages to the cells through which the MT will move, instead of all possible paths used in the DBR. As long as the MT follows the predicted paths, the number of communication messages of UPDBR will be much smaller than that of the DBR approach.

2.4 Concluding Remarks

In this chapter, we have proposed a unified differential bandwidth reservation (DBR) algorithm and a QoS-aware admission control scheme for cellular networks to guarantee a required level of QoS to on-going connections, while maintaining a competitive CBR for new connections. With the DBR scheme, the possible path of an MT that spans over a set of cells is divided into a couple of clusters in the form of a sector.



(a) Bandwidth utilization

(b) Message rate

Fig. 2.14. Comparisons of the UPDBR approach to the DBR approach (ϵ =1.5 and η = 2.0). Six bars for different policies and sector configurations are shown against offered load. The DBR(1,1), DBR(2,1), DBR(1,2), UPDBR(1,1), UPDBR(2,1), and UPDBR(1, 2) schemes are plotted from left to right.

The cells in the sector are further divided into two regions depending on whether they have an immediate impact on the handoff or not. Two different reservation policies are applied for these two regions. In addition, a variation of the DBR algorithm, called UP-DBR, exploits the moving pattern of an user to make efficient bandwidth reservation by minimizing the number of participating cells in handoff. Our admission control scheme, like prior studies, assumes *a priori* knowledge of the possible path of a new connection. This information is used to check the availability of the required bandwidth in a sector of cells prior to making the admission decision. However, unlike the prior schemes, the admission control algorithm uses a subset of cells in the sector to check if the required CDR can be maintained. The number of cells required for bandwidth reservation varies dynamically based on the average CDR of the cells in the sector.

Extensive simulations were conducted to compare the performance of the proposed schemes with the prior STATIC and PT_QoS policies. Unlike the prior policies, our approach can guarantee the required CDR over the entire workload, while maintaining a competitive CBR. The improved and guaranteed performance is possible by dynamically selecting the number of cells required to satisfy the QoS assurance. Also, the admission control policy is quite adaptable to different QoS requirements.

Chapter 3

A Cache Management Scheme for Internet-based Mobile Ad Hoc Networks

In this chapter, we propose a cache management scheme for improving the communication performance and data accessibility of Internet-based mobile ad hoc network (IMANET).

3.1 Introduction

Over the past decade, Internet has changed our daily life. With the recent advent in wireless technology and mobile devices, ubiquitous communication is touted to change our life further. It is envisaged that in the near future, users will be able to access the Internet services and information anytime and anywhere. To realize this vision, wireless carriers are developing state-of-the-art wireless communication infrastructures. Nevertheless, a mobile terminal (MT) may still have difficulty to connect to a wired network or Internet due to limited wireless bandwidth and accessibility. Under heavy traffic, an MT has to content for bandwidth and may get blocked from a wireless base station. Moreover, in some geographically remote areas, an infrastructure may not be even available. Thus, researchers are exploring an alternative technology, called *Mobile Ad Hoc Network* (MANET), for its low cost and ease of deployment.

A significant volume of research on MANETS has appeared in the literature in the past few years [30, 34, 38, 59, 60, 61, 62, 70]. Most of these efforts, however, have focused on developing routing protocols to increase connectivity among MTs in a constantly varying topology. Due to the users' growing interest and falling cost in accessing the wireless Internet, it has become imperative to consider the integration of MANET with the wired Internet. Thus, to put the MANET technology into the context of real life, we consider an *Internet-based* MANET, called IMANET [21], and investigate the problem of information search and access under this environment. In an IMANET architecture , we assume that some of the MTs are connected to the Internet or wired private networks[1]. Thus, an MT may access Internet information via a direct connection or via relays from other MTs. Although there may exist many potential applications, to the best of our knowledge, none of the previous work has addressed the issues for information search and access in IMANETS. The followings are some of the applicable scenarios for an IMANET:

- **Scenario 1**: During special events such as Olympic games or World Cup Soccer, the demand from users to access the Internet and communicate among themselves are exceedingly high. While a fixed infrastructure may be in place, it is challenging to accommodate all the users due to limited wireless bandwidth. With an IMANET, users can either access the required information directly or indirectly (through relays). Moreover, they can communicate among themselves without going through a wired infrastructure.

- **Scenario 2**: A visitor in a downtown, museum, or shopping mall may need to access various type of information (e.g. exhibition info, tour info including maps, restaurants of choice, hotels, theaters, and so on). A local service provider usually

---

[1] Without loss of generality, we use Internet to refer to both of Internet and wired private network for the rest of paper.

provides an electronic guide such as an info-station [92] that contains the relevant information. Although a visitor may lose connection to the info-station because of mobility, he/she can still access or share the information through relays using IMANET.

- **Scenario 3**: In a battle field or emergency site, one MT may be connected to the Internet by a satellite and serve as a proxy for other MTs. The accessed information and services can be shared by the other MTs via local ad hoc communication.

An IMANET has several constraints. First, not all the MTs can access the Internet. Second, due to mobility, a set of MTs can be separated from the rest of the MTs and get disconnected from the Internet. Finally, an MT requiring multi-hop relay to access the Internet may incur a longer access latency than those which have direct access to the Internet.

To address these constraints, we propose an *aggregate caching* mechanism for IMANETS. The basic idea is that by storing data items in the local cache of the MTs, members of the IMANET can efficiently access the required information. Thus, the aggregated local cache of the MTs can be considered as an unified large cache for the IMANET. The proposed aggregate cache can alleviate the constraints of IMANETS discussed above. When an MT is blocked from direct access to the Internet, it may access the requested data items from the local cache of nearby MTs or via relays. If an MT is isolated from the Internet, it can search other reachable MTs for the requested data item. Finally, if an MT is located further from the Internet, it may request the data items from other closeby MTs to reduce access latency.

Here, two issues are addressed for implementation of an aggregate caching mechanism in IMANETS:

- **Efficient search**: An efficient information search algorithm is fundamental for locating the requested data in IMANETS.
- **Cache management**: To reduce the average access latency as well as enhance the data accessibility, efficient cache admission control and replacement policies are critical. The cache admission control policy determines whether a data item should be cached, while the cache replacement policy intelligently selects a victim data item to be replaced when a cache becomes full.

Information search in an IMANET is different from the search engine based approach used in the wired Internet. An MT needs to broadcast its request to the possible data sources (including the Internet and other MTs within the IMANET) in order to retrieve the requested data efficiently. Thus, we propose a broadcast-based approach, called *Simple Search (SS)* algorithm, which can be implemented on the top of existing routing protocols to locate the requested data. In addition, we propose a cache admission control policy based on the distance between MTs to reduce redundant data caching, and a cache replacement policy based on time and distance, called *Time and Distance Sensitive (TDS)* replacement, to reduce the cache miss ratio and increase the accessibility of the aggregate cache.

We conduct a simulation based performance evaluation to observe the impact of caching, cache management, and access points (APs) (which are directly connected to the Internet) upon the effectiveness of IMANETS. The overall results show that the proposed methodology can relieve limitations of IMANETS and improve system performance significantly. Focusing on the constraints of the IMANET such as accessibility and latency, our contribution is threefold:

- A simple search algorithm is developed to facilitate information search and access in an IMANET;
- An aggregate cache for IMANETS is proposed to address the issues of accessibility and latency;
- A distance based admission control policy and three cache replacement policies (TDS_D, TDS_T, and TDS_N) are proposed as a part of the aggregate caching scheme. These policies are capable of providing better performance than the well known LRU replacement policy.

The rest of this chapter is organized as follows. Work related to the research is reviewed in Section 3.2. The system model and simple search algorithm, and the aggregate cache management mechanism are presented in Sections 3.3 and 3.4, respectively. Section 3.5 is devoted to performance evaluation and comparisons of various policies. Finally, we concludes the chapter with future directions in Section 3.6.

3.2 Related Work

Research on MANET has mainly focused on developing routing protocols such as Destination-Sequenced Distance Vector (DSDV) [61], Dynamic Source Routing (DSR) [38], Ad hoc On Demand Distance Vector (AODV) [62], Temporally-Ordered Routing

Algorithm (TORA) [60], and their variations. These algorithms assume that a sender MT knows the location of receiver MT based on the route information, which is accumulated and analyzed by a route discovery or route maintenance algorithm. Although a route discovery operation captures the current network topology and related information, it has to be executed whenever an MT needs to transmit a data item. To avoid repetitive route discovery, the MTs can cache the previous route information. Hu et al [34] compared the performance of two caching strategies based on the DSR routing protocol: a path cache and a link cache. In the path cache, a complete path from a source to the destination is stored. In the link cache, a group of paths, which are collected from previous route discovery or other operations, is constructed to generate a graph style data structure. In our work, instead of addressing the issue of route discovery and its caching, we emphasize on efficient information search and data caching to enhance data accessibility.

Caching is an important technique to enhance the performance of wired or wireless network. A number of studies has been conducted to reduce the Web traffic and overall network congestion by deploying various caching schemes in the Internet [24, 26, 91]. A cooperative caching scheme is suggested in [24], in which a couple of individual caches are treated as a unified cache and they interact among themselves to eliminate the duplicate copies, and increase cache utilization. Fan et al [26] proposed a summary cache, where proxies share their summary of cache contents represented by bloom filters. When a proxy has a cache miss for a request, it sends the request to other proxies based on a periodically updated summary of cache contents in other proxies. A proxy cache relocation scheme is proposed based on the prediction of user's mobility to reduce delay during a handoff, a mechanism of transferring an on-going call from the current cell to the next cell to which a user moves, in a cellular network [29]. However, no such work has been conducted in a MANET, in which a network topology frequently changes.

Ren et al [67] employed a semantic caching scheme to manage location-dependent data (e.g. weather, traffic, and hotel information), in which an MT maintains semantic description of data in a mobile environment. When an MT needs to generate a query, it processes the query, analyzes the descriptions, and finds out results (or partial results) from the appropriate cache. Based on the results, the MT tailors or reduces the query and requests the server to get the rest of results to reduce communication. In contrast to the traditional cache replacement policies, the Furthest Away Replacement (FAR) is used in this study. With this policy, a victim is selected such that it is not on the way in which the MT might move, but is located far away from the current location of the MT.

In particular in MANETS, it is important to cache frequently accessed data not only to reduce the average latency, but also to save wireless bandwidth in a mobile environment. Hara [30] proposed a replica allocation method to increase data accessibility in MANETS. In this scheme, an MT maintains a limited number of duplicated data items if they are frequently requested. Replicated date items are relocated periodically at every relocation period based on the followings: each MT's access frequency, the neighbor MTs' access frequency or overall network topology. Update of the replicated data is further considered in [31]. Since an MT cannot access data when it is isolated from others, replication is an effective means to improve data accessibility. Due to the limited size of information that an MT can maintain, however, simply replicating data items and accessing them in MANETs cannot fulfill users' requirements to access a wide variety of information, available over the Internet.

To overcome the limited information availability in MANETs, Sailhan et al [70] proposed a cooperative caching scheme to increase data accessibility by peer-to-peer communication among MTs, when they are out of bound of a fixed infrastructure. It is implemented on top of a well-known ad hoc routing protocol, called Zone Routing Protocol (ZRP). Papadopouli et al [59] suggested the 7DS architecture, in which a couple of protocols are defined to share and disseminate information among users. It operates either on a prefetch mode, based on the information and user's future needs or on an on-demand mode, which searches for data items in a single-hop multicast basis. Depending on the collaborative behavior, a peer-to-peer and server-to-client model are used. Unlike our approach, this strategy focuses on data dissemination, and thus, the cache management including a cache admission control and replacement policy is not well explored.

To the best of our knowledge, none of previous work has explored an aggregated caching scheme along with an efficient information search algorithm in the realm of IMANETS.

3.3 Information Search in IMANETS

3.3.1 System Model

In this subsection, we describe a generic system model of IMANETS. We assume that an MT can not only connect to the Internet but also can forward a message for communication with other MTs via a wireless LAN (e.g. IEEE 802.11), as used in most prior study [55, 59, 70]. As illustrated in Figure 3.1, an IMANET consists of a set of MTs that can communicate with each other using an ad hoc communication protocols (illustrated by dashed-line). Among the MTs, some of them can directly connect to the Internet, and thus serve as *access points*[2] (AP) for the rest of MTs in the IMANET. Thus, an AP is a gateway for the Internet and is assumed to have access to any information. An MT located out of the communication bound of an AP has to access the Internet via relays through one of the access points. An MT can move in any direction and make information search and access requests from anywhere in the covered area.

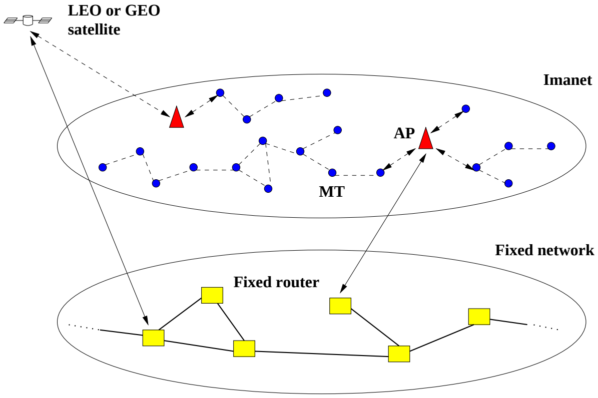


Fig. 3.1. A generic system model of an IMANET.

---

[2] The AP here is a logical notation. An AP equipped with appropriate antennas can directly communicate with the Internet through wireless infrastructures including cellular base stations, and Low Earth Orbit (LEO) or geostationary (GEO) satellites.

When an MT is located near by an AP (e.g. within one-hop), it makes a connection to the AP directly. When an MT is located far away from an AP, however, information access has to go through several hops in the ad hoc network before reaching the AP.

3.3.2 Information Search Algorithm

As mentioned in the Introduction, the main focus of this chapter is to support information access in IMANETS. Unlike a routing protocol, which establishes a path between a known source and destination, any MT can be an information source in the IMANET. Thus, without knowing the destination address for any requested information, a search algorithm is needed for IMANETS as is done in the Internet. In the following, we describe the basic idea of an information search algorithm. This algorithm can be implemented on top of an existing routing protocol for MANETS.

Since an aggregate cache is supported in an IMANET design, requested data data items can be received from the local cache of an MT as well as via an AP connected to the Internet. When an MT needs a data item, it does not know exactly where to retrieve the data item from, so it broadcasts a request to all of the adjacent MTs. When an MT receives the request and has the data item in its local cache, it will send a reply to the requester to acknowledge that it has the data item; otherwise, it will forward the request to its neighbors. Thus, as illustrated in Figure 3.2[3], a request may be flooded

---

[3] A dotted circle represents the communication range of an MT or an AP. For the sake of simplicity, we assume that both an MT and an AP have the same diameter of communication.

in the network and eventually acknowledged by an AP and/or some MTs with cached copies of the requested data item.

Based on the idea described above, we propose an information search algorithm, called *Simple Search (SS)*, to determine an information access path to the MTs with cached data of the request or to appropriate APs. The decision is based on the arriving order of acknowledgments from the MTs or APs. Let us assume an MT (n_i) sends a request for a data item (d) and an MT (n_k) is located along the path in which the request travels to an AP, where $k \in \{a, b, c, j\}$. The SS algorithm is described as follows:

1. When n_i needs d, it first checks its local cache. If the data item is not available in the local cache and n_i cannot directly access to an AP, it broadcasts a *request* packet to the adjacent MTs (g_i)[4]. The *request* packet contains the requester's id and request packet id. After n_i broadcasts the request, it waits for an acknowledgment. If n_i does not get any acknowledgment within a specified timeout period, it fails to get d.

2. When n_k receives a *request* packet, it forwards the packet to adjacent MTs (g_k) if it does not have d in its local cache. If n_k has the data d, it sends an *ack* packet to n_i. When an AP receives the *request* packet, it simply replies an *ack* packet. When an MT or AP forwards or sends the *ack* packet, the id of the MT or AP is appended in the packet to keep the route information. In contrast to a *request*

---

[4] For g_i, $g_i = \{n_j \mid \text{distance}(n_i, n_j) \leq \Upsilon\}$, where distance($n_i$, n_j) is calculated by $\sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}$ and Υ is the diameter of communication range of the MT. The x_i and y_i are the coordinates of n_i.

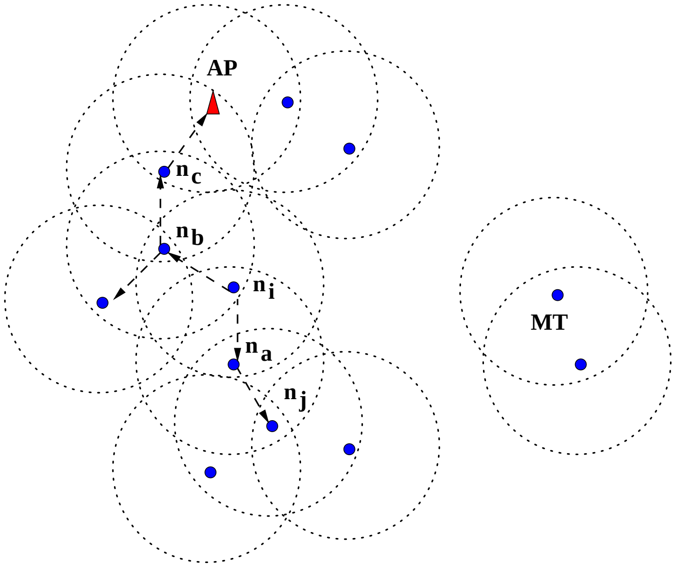


Fig. 3.2. An MT (n_i) broadcasts a request packet which is forwarded to the AP in the Imanet.

packet, which is broadcasted, the *ack* packet is sent only along the path, which is accumulated in the *request* packet.

3. When n_i receives an *ack* packet, it sends a *confirm* packet to the *ack* packet sender, e.g. an AP or n_k. Since an *ack* packet arrives earlier from an MT or AP that is closer to n_i, n_i selects the path based on the first receipt of the *ack* packet and discards rest of the *ack* packets.

4. When n_k or an AP receives a *confirm* packet, it sends the requested data (d) as using the known route.

When an MT receives a *request* packet, it checks whether the packet has been processed. If the packet has been processed, then the MT does not forward it to adjacent MTs, and discards it. For an *ack*, *confirm*, or *reply* packet, the MT also checks if its id is included in the path, which is appended to the packet. Since these packets are supposed to travel only along the assigned path that is established by the *request* packet, if the MT's id is not included in the path, the packet is discarded. We use a hop limit for a *request* packet to prevent floating of packets in the network. Thus, an MT does not broadcast a *request* packet to the adjacent MTs if the number of forwarded hops of the packet exceeds the hop limit. When the MT or an AP receives a *request* packet, it does not send the data item immediately, but sends an *ack* packet because other MTs or APs, which are located closer to the sender might reply earlier. This helps in reducing network congestion and bandwidth consumption by multiple data packets.

When a set of MTs is isolated (as shown in Figure 3.2) and cannot access the data of their interest because they are out of the communication range of an AP, they try to search among themselves with cached copies.

The proposed SS algorithm is illustrated in Figure 3.3, where we assume n_j has the data item in its local cache that n_i requested. Once the MT receives the requested data, it triggers the cache admission control procedure to determine whether it should cache the data item. The cache management scheme is described in the next section.

3.4 Aggregate Cache Management

In this section, we present the aggregate cache management policy including a cache admission control and a cache replacement policy.

3.4.1 An Aggregate Cache

In IMANETS, caching data items in the local cache helps in reducing latency and increasing accessibility. If an MT is located along the path in which the request packet travels to an AP, and has the requested data item in its cache, then it can serve the request without forwarding it to the AP. In the absence of caching, all the requests should be forwarded to the appropriate APs. Since the local cache of the MTs virtually form an aggregate cache, a decision as to whether to cache the data item depends not only on the MT itself, but also on the neighboring MTs.

In the aggregate cache, a cache hit can be of two types: a local cache hit or a remote cache hit. A local cache hit occurs when the requested data item is available in

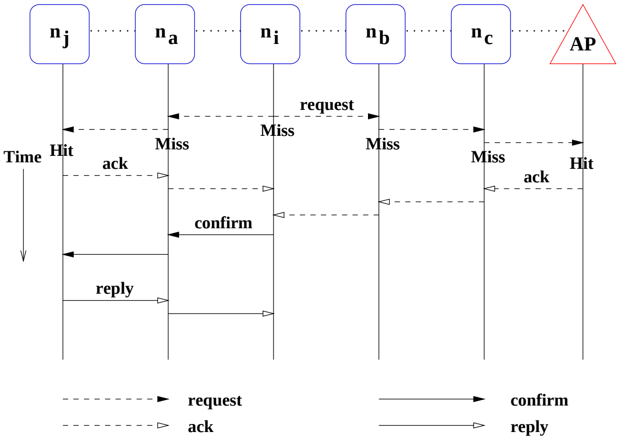


Fig. 3.3. A *Simple Search* algorithm in the IMANET. Let us assume that an MT (n_i) sents a *request* packet for a data item (d) and an MT (n_j) receives a forwarded *request* packet. n_j has the data d in its local cache and replies an *ack* packet to n_i. Then n_i sends a *confirm* packet to n_j, and n_j attaches d to the *reply* packet. Here, the dotted line between MTs or an MT and AP represents that they are located within communication range.

the MT's local cache. A remote cache hit implies that the data item is available in other MTs' local cache.

3.4.2 Cache Admission Control

When an MT receives the requested data, a cache admission control is triggered to decide whether it can cache this data. In this paper, the cache admission control allows an MT to cache a data item based on the distance of other APs or MTs, which have the requested data. If the MT is located within Γ hops from them, then it does not cache the data; Otherwise it caches the data item. Since cached data can be used by closely located MTs, the same data items are cached at least Γ hops apart. Here, Γ is a system parameter.

The primary idea is that, in order to increase accessibility, we try to cache as many data items as possible, while trying to avoid too many replications. There is a tradeoff between access latency and data accessibility in data replication. If the popular data are replicated a lot, then the average access latency to average access is reduced because there is a high probability of finding those data items in another closer MT. With high duplication, however, the number of distinct data items in the aggregate cache is less. Thus, the probability of finding less popular data items from other MTs becomes low. Even though the number of copies of popular data reduces due to the cache admission control, a data is accessible from other MTs/APs with a longer delay.

Although caching popular data aggressively in closer MTs helps in reducing the latency, in this work, we give more weight to data accessibility than to access latency. A rationale behind this is that it is meaningless to reduce access latency when a set of

MTs is isolated from other MTs or the AP, and they can not access any interested data items. Instead of waiting until the network topology changes, it is better for the MTs to have even higher probability of finding the requested data. Since Γ value enables more distinct data items to be distributed over the entire cache due to admission control, the overall data accessibility is increased.

3.4.3 Cache Replacement Policy

A cache replacement policy is required when an MT wants to cache a data item, but the cache is full, and thus it needs to victimize a data for replacement. Two factors are considered in selecting a victim. The first issue is the distance (δ), measured by the number of hops away from an AP or an MT, which has the requested data. Since δ is closely related to the latency, if the data item with a higher δ is selected as a victim, then the access latency would be high. Therefore, the data item with the least δ value is selected as the victim.

The second issue is the access frequency of data items. Due to mobility of the MTs, the network topology may change frequently. As the topology varies, the δ values become obsolete. Therefore, we use a parameter (τ), which captures the elapsed time of the last updated δ. The τ value is obtained by $\frac{1}{t_{cur} - t_{update}}$, where t_{cur} and t_{update} are the current time and the last updated time of δ for the data item, respectively. If τ is closer to 1, δ has recently been updated. If it is closer to 0, the updated gap is long. Thus, τ is used as an indicator of δ to select a victim.

An MT maintains the δ and t_{update} values for each data item in the local cache. The mechanism to update δ and t_{update} is described as follows (refer to Figure 3.3).

1. After n_j receives the *confirm* packet, it checks the δ of the requested data item between n_i and n_j. If δ is $\geq \Gamma$ and is less than previously saved δ of the data item, then n_j updates the old δ with the new δ. Otherwise, n_j does not update δ, because d will not be cached in n_i based on the cache admission control. The δ value is obtained by counting the number of MTs' ids accumulated in the packet.

2. When n_i receives the data item in the *reply* packet, it checks the δ value of the data between n_i and n_j, and then chooses a victim and replaces it with d, if δ is $\geq \Gamma$. In addition, n_i saves δ and t_{cur}, which is t_{update} for the data item.

In this research, we suggest a *Time and Distance Sensitive (TDS)* replacement based on these two parameters. Depending on the weight assigned to the two parameters, we propose three schemes below (refer to Figure 3.3).

- *TDS_D*: We mainly consider the distance (δ) value to determine a victim. If there is a tie, then τ is considered the second criteria. We add the two parameters and choose the data item that has the least value of $(\delta + \tau)$. Note that δ is ≥ 1, but τ is in the range of $0 \leq \tau \leq 1$.

- *TDS_T*: A τ value is mainly considered to determine a victim. Thus, a victim is selected with the least τ value. As we mentioned before, t_{update} is updated when n_j receives the *confirm* packet and n_i receives the *reply* packet. Here, δ of the requested data item between n_i and n_j is $\geq \Gamma$.

- *TDS_N*: Both distance and access frequency are considered to determine a victim. We multiply the two factors and select the data item with the least $(\delta \times \tau)$ value.

Notations:

$t_{cur}, t_{update}, \tau, \delta$: Defined before.
C_i: A local cache in MT n_i.
d_n: A data item cached in the n^{th} slot in the local cache, where $0 \leq n < C_{size}$ (C_{size} is the cache size).
τ_n: A calculated τ value of d_n.
δ_n: A δ value of d_n.

(A) When n_i receives a data item d, it calculates δ. /* cache admission control is triggered. */

```
    if ($\delta \geq \Gamma$) {
        if (empty cache slot is available in $C_i$)
            cache $d$;
        else
            call cache_replacement_policy();
        store $\delta$ and $t_{cur}$, which is saved as $t_{update}$;
    }
    else
        do not cache $d$;
```

(B) Procedure cache_replacement_policy()

```
    calculate $\tau$ by $\frac{1}{t_{cur} - t_{update}}$;
    for  $d_n \in C_i$  do {
        calculate $\tau_n$;
        find $d_n$ which has the minimum $\delta_n \times \tau_n$ value;
    }
    replace $d_n$ with $d$;
```

Fig. 3.4. The pseudo code of the aggregate cache management algorithm used in an MT. We use the TDS_N replacement policy. The TDS_D and TDS_T can be implemented by slightly modifying the cache_replacement_policy() procedure.

The TDS_T scheme is different from the traditional *Least Recently Used (LRU)* cache replacement policy, which is associated with the time of reference of the data items (t_{ref}). In the LRU scheme, a requested data is cached without considering an admission control policy. Thus, whenever an MT receives the data item in the *reply* packet, one of the local data items that has the highest (t_{cur} - t_{ref}) value is selected as the victim. In addition, when n_j receives the *confirm* packet and n_i receives the *reply* packet, t_{ref} is updated regardless of the δ values of the requested data item between n_i and n_j.

The overall aggregate cache management algorithm is given in Figure 3.4.

3.5 Performance Evaluation

3.5.1 Simulation Testbed

We use a wrap around network to examine the proposed idea. We assume that an AP is located in the center of an area. The MTs are randomly located in the network. The request arrival pattern follows Poisson distribution with a rate of λ. The speed (s) of the MTs is uniformly distributed in the range ($0.0 < s \leq 1.0$ m/sec). The *random waypoint mobility* model, developed in [38], is used to simulate mobility here. With this approach, an MT travels toward a randomly selected destination in the network. After the MT arrives at the destination, it chooses a rest period (pause time) from a uniform distribution. After the rest period, the MT travels towards another randomly selected destination, repetitively. An MT does not move at all if its pause time is infinite, represented as *Inf*. If the pause time is 0, then it always moves.

To model the data item access pattern, we use two different distributions: Uniform and Zipf distribution [96]. The Zipf distribution is often used to model a skewed access pattern [33, 91], where θ is the access skewness coefficient that varies from 0 to 1.0. Setting $\theta = 0$ corresponds to the uniform distribution. Here, we set θ to 0.95. We have written an event-driven simulator using CSIM [73] to conduct the performance study. The simulation results are illustrated as a function of the pause time. The other important simulation parameters are summarized in Table 3.1.

Table 3.1. Simulation parameters

Parameter	Value
Network size (m)	3000×3000
Number of MTs	200
Number of data items	1000, 10000
Cache size (items/MT)	16
Transmission range (m)	250
Number of APs	1, 4, 16
Inter request time (sec)	600
Pause time (sec)	0, 100, 200, 400, 800, 1600, Inf

3.5.2 Performance Parameters

We evaluate three performance parameters here: throughput or fraction of successful requests (Φ), average number of hops (Ω), and cache hit ratio (h) including local cache hit and remote cache hit. Throughput Φ denotes the fraction of successful requests and is used to measure the accessibility of the MTs in the IMANET. If r_{total} and r_{suc} denote the total number of requests and the number of successfully received data items, then Φ is defined as,

$$\Phi = \frac{r_{suc}}{r_{total}} \times 100\% \ .$$

The average number of hops (Ω) represents the average hop length to the APs or MTs of successfully received data items. If Ω_r denotes the hop length for a successful request r, then Ω is expressed as,

$$\Omega = \frac{\sum_{r \in r_{suc}} \Omega_r}{r_{suc}} \ .$$

Since the number of hops is closely related to the communication latency, we use Ω to measure average latency. Finally, the hit ratio h is used to evaluate the efficiency of the aggregate cache management. If n_{local} and n_{remote} denote the number of local hits and remote hits respectively, then h_{local}, h_{remote}, and h are expressed as:

$$h_{local} = \frac{n_{local}}{n_{local} + n_{remote}} \times 100\% \ ,$$

$$h_{remote} = \frac{n_{remote}}{n_{local} + n_{remote}} \times 100\% \ ,$$

$$h = \frac{n_{local} + n_{remote}}{r_{suc}} \times 100\% \ .$$

3.5.3 Simulation Results

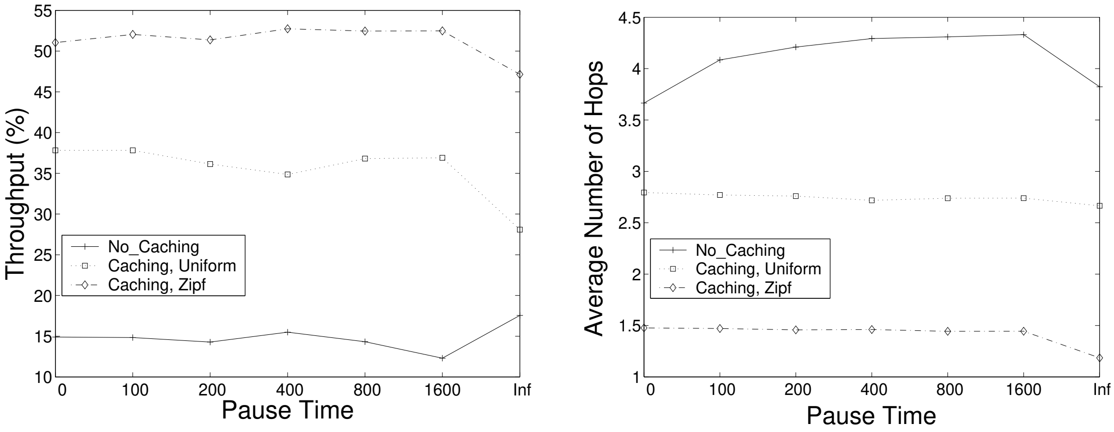
In this subsection, we examine the impact of caching and cache management including admission control and replacement policy on the IMANET performance. Then we discuss the impact of number of APs. Since there are only few APs available in a given area due to limited resource environment in an IMANET, in all the discussion, we use a single AP unless otherwise stated.

3.5.3.1 Impact of Caching

We investigate the performance implications of the aggregate cache, using two data access patterns: uniform and Zipf distributions. In Figure 3.5, the TDS_D and TDS_T cache replacement policies are used for caching with data access pattern of uniform and Zipf distribution, respectively. We have simulated all other policies, but discuss only a subset of the important results. For a system without any cache, an access pattern does not make any performance difference, because a request can not be satisfied by any MT but by an AP.

In Figure 3.5(a), data accessibility is greatly improved when we use the aggregate cache. Φ is increased more than twice compared to the no cache case. With caching, there is a high probability of the requested data being cached in the MT's local cache or at other MTs. Even though a set of MTs is isolated from an AP, in contrast to the no cache case, they still try to access the cached data items among themselves. Further improvement is possible depending on the access pattern. Note that almost 200% improvement is achieved compared to the no cache case, when data access pattern follows Zipf distribution.

Figure 3.5(b) shows the effect of the aggregate cache on the average latency. Since a request can be satisfied by any one of the MTs located along the path in which the request is relayed to the AP, unlike to the no cache case, data items can be accessed much faster. As expected, Ω is reduced with caching by more than 50%. The results clearly demonstrate the effectiveness of the aggregate caching scheme.



(a) Throughput (Φ)

(b) Latency (Ω)

Fig. 3.5. Throughput (Φ) and latency (Ω) as a function of pause time.

3.5.3.2 Impact of Cache Management

In this subsection, we evaluate the cache management policy in terms of the impact of Γ on admission control and impact of the cache replacement policy. We compare the performance of our TDS schemes against the *Least Recently Used (LRU)* policy.

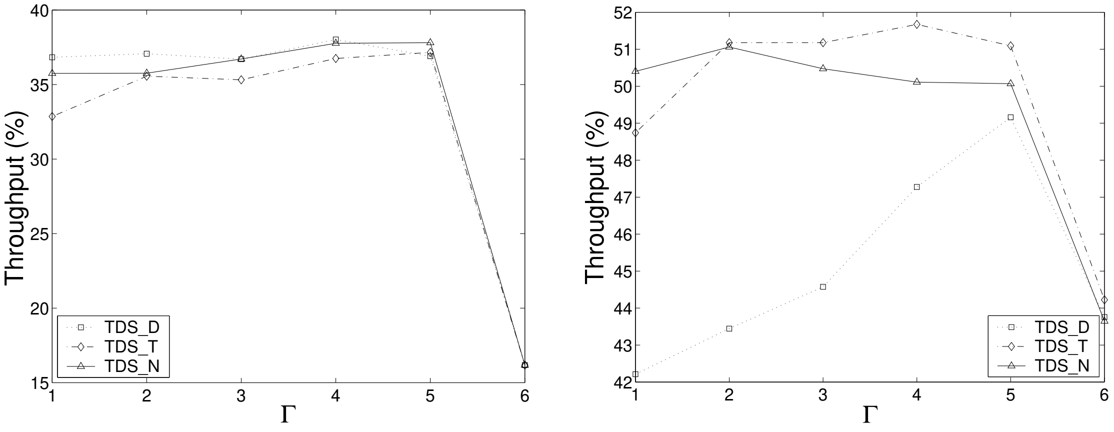
Impact of Γ in Admission Control: We examine the performance effect of parameter Γ, which determines which data item can be cached. Although a high Γ value enables more data items to be distributed over the entire cache, so that more distinct data items will be cached, the average access latency will increase. In this research, as mentioned before, data accessibility is considered more important than access latency.

In Figure 3.6(a), throughput Φ degrades after $\Gamma = 5$. An MT does not cache a data item according to the admission control policy, when the data is available within five hops. Thus, performance is almost similar to the no cache case at $\Gamma = 6$, because only a few data items are cached. TDS_D has the highest Φ followed by the TDS_N and then TDS_T. Due to the uniform access pattern, δ has more effect on the performance than that of τ. Since TDS_N gives equal importance to δ and τ, it shows higher Φ than TDS_T but lower than TDS_D.

In Figure 3.6(b), Φ of all schemes drops after $\Gamma = 5$ for similar reason discussed above. When the access pattern follows the Zipf distribution, however, TDS_T shows the best performance. Since t_{update} of popular data items is more frequently updated than that of less popular data items, there is a high probability of a less popular data item being selected as a victim. Also, the probability of a popular data item to be found in other MTs is high. As the result indicates, τ has more impact on throughput than that of δ. Throughput can be further enhanced by tuning the Γ value.

3.5.3.3 Impact of Cache Replacement Policy

The impact of the suggested cache replacement polices on performance is investigated with different data access patterns. Based on Figure 3.6(a), we set Γ as four, five,



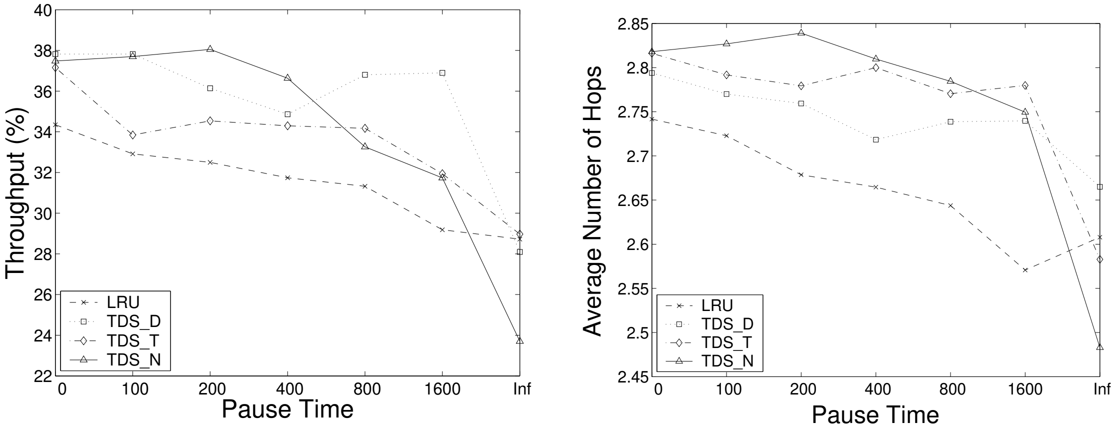
(a) Uniform distribution (b) Zipf distribution

Fig. 3.6. Throughput (Φ) as a function of Γ.

and five for TDS_D, TDS_T, and TDS_N policies, respectively. In addition, we simulate the LRU policy for comparison.

In Figure 3.7, we use uniform distribution and set the total number of data items to 1000. In Figure 3.7(a), as the pause time increases, overall Φ of the TDS schemes and LRU decreases. It implies that the isolation period of a set of MTs from other MTs or the AP becomes longer due to slow movement of MTs. For instance, when an MT does not move (pause time is *Inf*) and is isolated, its data accessibility is very low for the entire simulation time. TDS_D and TDS_N have higher Φ than TDS_T in high mobility. The LRU scheme shows the lowest performance due to data access pattern.

Figure 3.7(b) demonstrates the effect of the aggregate cache on the latency, where TDS_D has lower Ω than TDS_T and TDS_N. The LRU scheme shows the lowest Ω because it does not filter an accessed data item but simply caches it.



(a) Throughput (Φ)

(b) latency (Ω)

Fig. 3.7. Throughput (Φ) and latency (Ω) comparison with Uniform distribution.

Figure 3.8 shows h_{local} and h_{remote} for different pause times and the h_{remote} is almost up to 90% of h. h_{local} is quite small compared to h_{remote}, because the aggregated cache size is larger than a local cache. Our TDS schemes show higher h than the LRU policy.

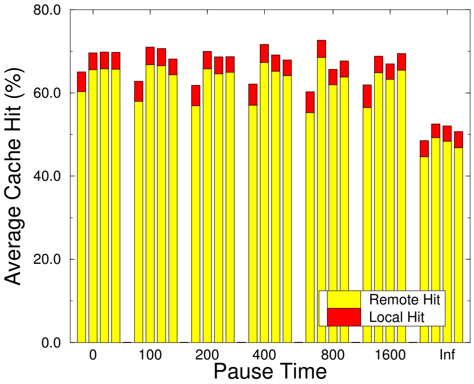


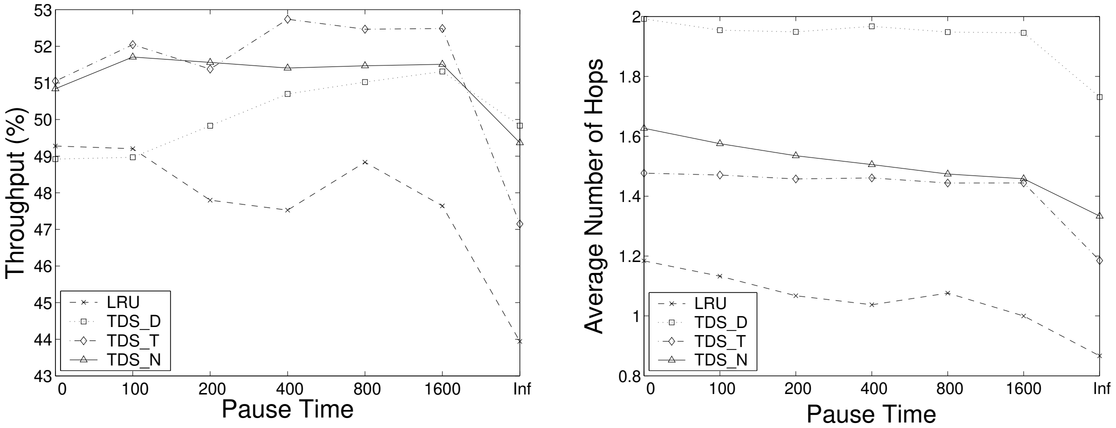
Fig. 3.8. Average cache hit ratio (h) comparison with Uniform distribution (Four stack bars for different replacement policies are shown against pause time. The LRU, TDS_D, TDS_T, and TDS_N are plotted from left to right).

Next, we examine the impact of data access pattern using Zipf distribution with 10000 data items in Figure 3.9. Based on Figure 3.6(b), we set Γ as five, four, and two for TDS_D, TDS_T, and TDS_N, respectively.

Figure 3.9(a) demonstrates that the effect of the aggregate cache is more significant when the access pattern follows Zipf distribution. TDS_T has the best performance followed by TDS_N, TDS_D, and LRU. Since the popular data items are frequently requested, all the TDS schemes and LRU gain more benefit compared to the uniform distribution. In general, all variations of TDS replacement show better performance than the LRU scheme with the Zipf distribution. A drawback of the LRU scheme is that is has too much replication of popular data items, and thus results in lower data accessibility. However, with the TDS policies, data items are treated more fairly in the sense that the number of replications for the most popular data items is restricted due to the cache admission control.

In Figure 3.9(b), all the TDS schemes have higher Ω than the LRU policy because the cache admission control allows caching only when a data item is Γ hops away. Since the LRU policy caches the requested data without using the admission control, most frequently accessed data items are stored in multiple MTs. Due to higher replication of popular data items, LRU has a smaller Ω.

Even if a data item is popular and can be received from a near by MT, it cannot be cached due to the cache admission control. Thus, the average popularity of local cache in TDS_D, TDS_T, and TDS_N is smaller compared to the LRU policy. For instance, when an MT is isolated, it can only access data items in its local cache. Because of the less popularity of data items, an MT will have less h_{local} for the TDS policies. However, in contrast to the LRU scheme, where the h_{local} is high, the TDS_D, TDS_T, and TDS_N have higher remote cache hit due to cache admission control, which prevents arbitrary data replication. This is shown in Figure 3.10. Note that the TDS_T policy has slightly better hit ratio along with its high throughput.



(a) Throughout (Φ)

(b) Latency (Ω)

Fig. 3.9. Throughout (Φ) and latency (Ω) comparison with Zipf distribution.

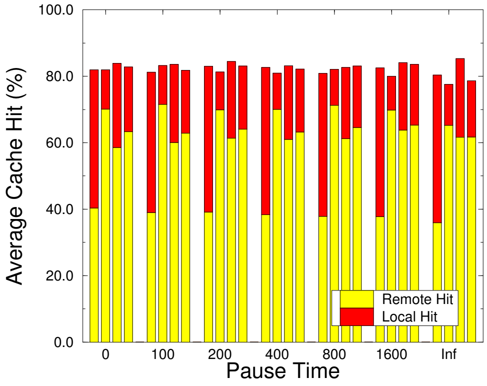


Fig. 3.10. Average cache hit ratio (h) comparison with Zipf distribution (Four stack bars for different replacement policies are shown against pause time. The LRU, TDS_D, TDS_T, and TDS_N are plotted from left to right).

In summary, the aggregate cache improves IMANET performance significantly and the proposed TDS scheme is a viable cache replacement policy.

3.5.3.4 Impact of Number of APs

Since the number of APs can affect the performance in an IMANET, we disable the caching ability of the MTs to study the impact of number of APs. As the number of APs increases, Φ increases up to 90% (at AP = 16). Intuitively, if more APs are deployed in a given area, the probability of an MT being connected to an AP (either directly or indirectly) increases and thus throughput is increased.

For the effect of number of APs on the access latency, as the number of APs increases, Ω reduces as expected. This implies that the accessibility of an MT to an AP increases. These results are not included here since the focus of the paper is on caching, and the results can be found in [53].

3.6 Concluding Remarks

In this chapter, we proposed an aggregate caching scheme to improve the communication performance of an IMANET, a ubiquitous communication infrastructure consisting of both the wired Internet and wireless MANET. An IMANET is envisioned to provide access to Internet information and services from anywhere anytime. The aggregate caching concept combines the local cache of each user (MT) in forming an unified cache that can alleviate the limited data accessibility and longer access latency problems. The caching scheme includes a broadcast based search and a cache management technique. The proposed simple search (SS) algorithm ensures that a requested data is obtained from the nearest MT or AP. The aggregate cache management scheme has two parts: a cache admission control and a cache replacement policy. The admission control prevents high data replication by enforcing a minimum distance between the same data items, while the replacement policy helps in improving the cache hit ratio and accessibility. Three variations of the replacement policy are considered in this paper by assigning different weights to the time and distance parameters of the TDS scheme.

A simulation based performance study was conducted to examine the advantages of the proposed scheme from three different perspectives: impact of caching, impact of cache management, and impact of number of APs. The three variations of the TDS replacement policy were compared against the traditional LRU policy. It was observed that regardless of the cache replacement policies, caching in IMANETS can significantly improve communication performance in terms of throughput and average access latency compared to an infrastructure without any cache. The performance advantage of the aggregate cache was magnified for skewed access patterns. Also, performance improvement due to caching was better with even a single access point to the Internet.

There are many challenges that need further investigation to exploit the full potential of IMANETS. Currently, we are examining the following issues:

- For simplicity, we assumed that data items are never updated. We would relax this assumption to incorporate data modification capability. This brings in the cache invalidation and cache update issues. In an IMANET, cache invalidation and update are challenging because of link disconnection and change of network topology. In light on this, we are currently developing various cache invalidation techniques [52] suitable for IMANETS.

- We did not consider various network topologies that may lead to network partitions. Thus, we plan to investigate the impact of the caching scheme on communication performance under different mobility patterns including Manhattan grid and modified random waypoint mobility models.

Chapter 4

Cache Invalidation Strategies for Internet-based Mobile Ad Hoc Networks (IMANETs)

In previous chapter, we assumed that data items are never updated. In this chapter, we relax this assumption and propose several push and pull-based cache invalidation strategies for IMANETS.

4.1 Introduction

Caching frequently accessed data items in MTs is an effective techniques to enhance the communication performance in IMANETS. In the previous chapter, we proposed an *aggregate cache* for IMANETS where, by storing the data items in the local caches of the MTs and making them available to other MTs, members of the IMANET can efficiently access the data items. Thus, the aggregated local cache of the MTs can be considered as a unified large cache for the IMANET. By using a data item cached in the aggregate cache, we can reduce the average access delay, and thus, save the scarce wireless bandwidth. However, a critical design issue, *cache invalidation*, needs to be addressed for applications requiring data consistency with the server. When a data item in a server is updated, it is necessary to make sure that the cached copies of this data item are validated before they can be used.

Cache invalidation problem has been well studied in mobile computing environments [11, 12, 32, 37, 40, 45, 68, 72, 90]. However, these studies are all limited to cellular wireless networks in which the MTs only communicate with the base stations (BSs) but not with each other. In the cellular environment, a message from the BS can be delivered to all the MTs in one broadcast. This helps a great advantage to the server initiated invalidation methods. Indeed, many invalidation report (IR) based schemes were proposed in the literature to explore this advantage [11, 12, 32, 37, 90]. In these schemes, the server periodically broadcasts an IR. By knowing the arrival time of the IRs, the MTs may save energy by staying in the sleep mode or turning off their radio components.

Our study of cache invalidation in an IMANET environment is motivated by the following observations: 1) the communication model in an IMANET is different from that in the cellular environment - in an IMANET, a multi-hop communication is employed; 2) the cost for a broadcast in an IMANET is different from that in the cellular environment - a broadcast may lead to flooding of the whole network; 3) the connectivity to a server is less stable than that in the cellular environment - due to the user mobility, an MT may be disconnected or isolated from the server or other MTs. Thus, the cache invalidation problem in an IMANET is unique from that in a cellular network due to multi-hop message relay, operation cost model, and uncertainty in message delivery.

In this chapter, we examine several push and pull-based cache invalidation schemes for IMANETS. In order to adapt and enhance these schemes for IMANETS, we propose a scheme, called *global positioning system-based connectivity estimation (GPSCE)*, for assessing the connectivity of an MT to a server (i.e., AP). With this enhancement, an MT can check whether it can access a server directly or indirectly through a multi-hop relay. In addition, we extend the simple search protocol [51], proposed in the previous chapter, to better meet the requirements of cache invalidation schemes. With GPSCE and the modified search protocol, we adapt three existing cache invalidation schemes for IMANETS: the first one, called *aggregate cache-based on demand (ACOD)*, is based on the *pull* strategy; while the other two, namely *modified timestamp (MTS)* and *MTS with updated invalidation report (MTS+UIR)* and *asynchronous (ASYNC)*, are based on the *push* strategy.

We conduct simulation-based performance evaluation using the *ns-2* package to observe the impact of query interval, cache update interval, aggregate cache size, and the search algorithm on system performance. The results indicate that the pull-based ACOD strategy provides high throughput, low query latency, and low communication overhead. The revised search protocol, applicable to all cache invalidation schemes, helps in reducing the number of communication messages significantly. The overall results show that the ACOD cache invalidation scheme is superior in all aspects, and thus, is a viable approach for implementation in IMANETS.

The rest of chapter is organized as follows. The related work is reviewed in Section 4.2. The system model is introduced in Section 4.3, and the GPS-based connectivity estimation scheme is described in Section 4.4. The search algorithm and four cache invalidation schemes are presented in Section 4.5. Section 4.6 is devoted to performance evaluation and the conclusions are drawn in the last Section.

4.2 Related Work

There has been a lot of research efforts [11, 12, 32, 37, 40, 45, 68, 72, 90] on the caching issue in wired and wireless networks. All caching schemes assume a stateful or a stateless server depending on whether it maintains the information about which data items are cached by which clients. In the stateful approach, when a data item is updated, the server sends an invalidation message to those clients who have the cached data item, e.g. Andrew File System (AFS) [45]. On the other hand, in the stateless approach, a client sends a message to the server to verify the validity of the cached data before the data item is used, e.g. Network File System (NFS) [72]. With this approach, it may incur high message traffic.

In wired networks, Roussopoulos et al [68] suggested an update propagation mechanism for peer-to-peer (P2P) networks. An intermediate client caches the index entries to locate the clients, where the contents are cached or stored, to reduce access latency and to balance the workload. By propagating the updated index entries, intermediate clients' index entries are maintained. However, cached index entries may become obsolete due to mobility of MTs that incurs changes of network topology in mobile environments.

Various cache invalidation schemes [11, 12, 32, 37, 40, 90] have been suggested for cellular networks, where the MTs are one hop away from the BS or a server. Barbara et al [11] proposed a scheme for a stateless server, where the server periodically broadcasts the *invalidation report (IR)*, in which the updated data items are marked. Depending on the report size, there is a tradeoff between the efficiency of cache invalidation and latency for a given communication bandwidth. The timestamp (TS) scheme limits the report size by broadcasting only the updated data items during a window of w IR intervals (w being a fixed parameter). Most of the prior approaches [12, 32, 37, 90] are variations of the TS scheme. In [40], an asynchronous scheme (AS) is suggested, where the server broadcasts an IR whenever a data item is updated and saves it at the home location cache (HLC), located in the server. However, none of these techniques can be directly applied to an Imanet, where an MT may be multi-hops away from the server.

For Manets, Hara [31] suggested a replication scheme for periodically updated data items, based on the previously suggested replica allocation method [30]. Replicated data items are reallocated periodically depending on the access frequency, remaining time to next update, and network topology. However, estimation of optimal reallocation period and remaining time to the next update are not practically feasible. In [42], a data dissemination strategy has been suggested in the presence of network partitions in Manets, where a subset of MTs (quorum) is selected as a server. Since not all the servers in the quorum are reachable from the MTs, several heuristics are suggested to make the update/query operations efficient. For the similar purpose, a probabilistic quorum system for ad hoc networks (PAN) [56] has been proposed to provide a reliable storage for query and update operations.

For the Imanet environment, Sailhan et al [70] proposed a cooperative caching scheme to increase data accessibility by peer-to-peer communication among the MTs when they are out of bound of a fixed infrastructure. Papadopouli et al [59] proposed the 7DS architecture, in which several protocols are defined to share and disseminate information among users. It operates either in a prefetch mode, based on the information popularity and user's future needs or in an on-demand mode, which searches for data items using in one hop multicast communication. Depending on the collaborative behavior, a peer-to-peer or server-to-client model is used. Unlike our research, they mainly focus on data dissemination, and thus, cache invalidation and cache update issue are not explored.

4.3 System Model

As illustrated in Figure 4.1, an IMANET consists of a set of MTs that can communicate with each other using an ad hoc communication protocols (illustrated by the dashed-line). An MT can move in any direction and make information search and access requests from anywhere in the covered area. Among the MTs, some of them can directly connect to the Internet, and thus serve as APs[1] for the rest of MTs in the IMANET. An MT located out of the communication range of the AP has to access the Internet via relays through one of the APs. Thus, an AP is a gateway for the Internet and is assumed to have access to any information. An AP is located in a communication area and is connected to a database server[2]. Thus, an MT can not only connect to the Internet, but also can forward a message for communication with other MTs via a wireless LAN (e.g. IEEE 802.11), as used in most recent studies [51, 55, 59].

A database may be attached to an AP, a fixed router or a database server. We assume that the database consists of a total of n data items (DB_{max}). A data item (d) is the basic unit of an update or query operation. The database is only updated by the server, while a query is generated by the MTs for read-only requests. An MT can cache a limited number of data items since the size of the cache space in an MT is relatively small compared to the total number of data items in the database. Depending on the cache

---

[1] An AP here is a logical notation. An AP equipped with appropriate antennas can directly communicate with the Internet through wireless infrastructures including cellular base stations (BSs), Low Earth Orbit (LEO), or geostationary (GEO) satellites.

[2] From an MT's point of view, since an AP is transparent to a database server, we use the terms AP and a database server (later in short, server) interchangeably.

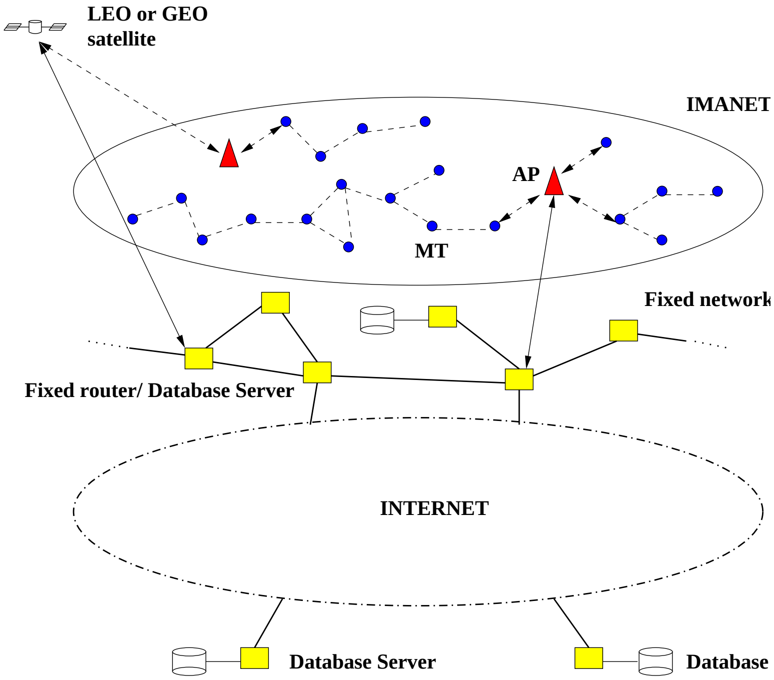


Fig. 4.1. A system model of an Imanet.

invalidation strategy, a server can broadcast the list of *ids* of updated data items to the MTs to ensure data consistency, either synchronously (periodically) or asynchronously.

4.4 A GPS-based Connectivity Estimation Scheme

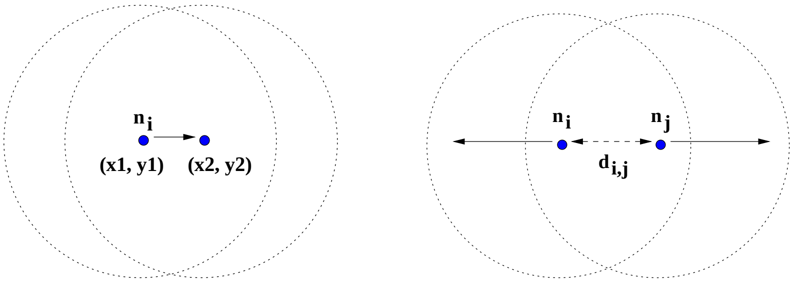
In an IR-based cache invalidation scheme for cellular networks [11, 12, 32, 37, 90], even if a query can be answered by a cached data item, an MT always waits for the next IR. Then it invalidates the cached data items accordingly, and answers the query if the cached copy is still valid. Otherwise, the MT sends a request to the server for a valid copy. Here, the basic assumption is that an MT is supposed to receive an IR regularly. However, in an IMANET, an MT may not receive an IR due to link disconnections and time-varying network topologies. Without knowing whether an MT can receive an IR from the server directly or indirectly through multi-hop relays, it may wait for the next IR, which may not arrive at all. It implies that, unlike a cellular network, connectivity with the server is not always assured.

To address this issue, we propose a *GPS-based connectivity estimation (GPSCE)* scheme, which is used as a base algorithm for supporting any cache invalidation mechanism in IMANETS. In this scheme, by using the location information obtained by the GPS, an MT can know whether it is currently connected to a server. When an MT is unable to send (receive) a message to (from) a server, a query cannot be processed, because an MT neither can check the validity of a cached data item nor can receive a valid data item from a server. Once the MT is aware of its status, it can decide what operation should be followed, such as wait for an IR, send a request message or deny a query.

Providing a location tracking capability in mobile devices is becoming popular due to the decreasing cost of GPS IC chips and increasing number of potential applications such as finding an emergency caller's location, and a car's navigation system. Thus, we assume that an MT is equipped with an on-board GPS receiver and is aware of its current location. Although other approaches [15, 66] are also feasible to find out the current location, we do not consider those in this chapter. Similar GPS-based techniques [6, 48] have been used for developing routing algorithms for Manets.

4.4.1 The Proposed GPSCE Scheme

Using the GPS, we develop a connectivity estimation algorithm, called GPSCE. This algorithm estimates the maximum communication period of an MT, which is located a single-hop or multi-hops away from a server, based on the distance. Then, the MT is aware of its communication status to a server.



(a) (b)

Fig. 4.2. A vector of n_i and the distance between n_i and n_j in (a) and (b), respectively.

The algorithm works as follows: Let us assume that an MT i (n_i) samples its location periodically. In Figure 4.2(a), the locations of n_i at time t and $t + \Delta t$ are $i(t) = (x_1, y_1)$ and $i(t + \Delta t) = (x_2, y_2)$, respectively. Based on these two locations, a vector of n_i is given by

$$\vec{i} =< x_2 - x_1, y_2 - y_1 > . \tag{4.1}$$

By using the vector $\vec{i}$, we can predict a new location of n_i after time t' by $i(t') = (a_i \cdot t' + x_i,\ b_i \cdot t' + y_i)$, where a_i $(= x_2 - x_1)$ and b_i $(= y_2 - y_1)$ are the x and y vectors of n_i, respectively.

Let us assume that the current locations, and the vectors of two MTs n_i and n_j are $i = (x_i, y_i)$ and $\vec{i} =< a_i, b_i >$, and $j = (x_j, y_j)$ and $\vec{j} =< a_j, b_j >$, respectively. In Figure 4.2(b), if n_i and n_j move in opposite directions, they will be out of each other's communicate range. To estimate the period in which both n_i and n_j are located within each other's communication range, first we calculate the distance between them. The new locations of n_i and n_j after time t' are $i(t') = (a_i \cdot t' + x_i,\ b_i \cdot t' + y_i)$ and $j(t') = (a_j \cdot t' + x_j,\ b_j \cdot t' + y_j)$. Thus, the distance between them, $d_{i,j}$, is given by

$$d_{i,j} = \sqrt{((a_j - a_i) \cdot t' + (x_j - x_i))^2 + ((b_j - b_i) \cdot t' + (y_j - y_i))^2}. \tag{4.2}$$

During communication, both n_i and n_j should be located within each other's communication range (R). Here, we assume that all the MTs in a network use the same communication range, which is already known to them. Thus, $d_{i,j}$ should satisfy the following condition.

$$d_{i,j} \leq R. \tag{4.3}$$

Since $d_{i,j}$ is positive, we have $d_{i,j}^2 - R^2 \leq 0$. From this, we can estimate the maximum period ($P > 0$) in which both n_i and n_j are located within each other's communication range as

$$\begin{aligned} P &= \frac{-e + \sqrt{e^2 - 4 \cdot c \cdot h}}{2 \cdot c}, \qquad (4.4) \\ \text{where} \quad c &= (a_j - a_i)^2 + (b_j - b_i)^2, \\ e &= 2 \cdot ((a_j - a_i) \cdot (x_j - x_i) + \\ &\quad (b_j - b_i) \cdot (y_j - y_i)), \text{ and} \\ h &= (x_j - x_i)^2 + (y_j - y_i)^2 - R^2. \end{aligned}$$

Based on the above equation, an MT n_i that is a single-hop away from a server can obtain the P_i to the server. Once n_i knows the P_i, it can calculate the connection expire time (T_i), the time when it is out of the communication range of a server. This is given by

$$T_i = t_{cur} + P_i, \qquad (4.5)$$

where t_{cur} is the current time. For the MTs, which are multi-hop away from a server, let us assume that an MT n_j accesses the server by relaying a message through n_i (refer to Figure 4.3). Also, T_i and T_j are the connection expire times between the AP and n_i, and n_i and n_j, respectively. If any one of these connections is broken, n_j loses its connection to the server. Thus, we choose the minimum value of T, and T_j is given as

$$T_j = \text{Min}(T_i, T_j). \qquad (4.6)$$

Once n_j calculates T_j, it can decide whether it is currently connected to a server. In this research, the value of T is used as an initial condition to proceed with a query for any cache invalidation strategy.

4.4.2 A Communication Protocol for the GPSCE Scheme

In this subsection, we describe the communication protocol of the GPSCE scheme. When the connection expire time is estimated, a quintet ($[id, l, v, h, t_{event}]$), called a *mobility quintet* (Q), is attached to a message. Here id is the identification of a server or an MT. l and v are the current location and vector, respectively. h is the number of hops from a server, and t_{event} is the time when a message is sent. Each MTs keeps a status bit (f) to indicate whether it is accessible to a server. Initially, all the MTs set f to 0.

We now describe three situations to estimate the communication period; when an MT is single-hop away from a server, multi-hop away from a server, and when an MT changes its direction. First, we explain the case when an MT n_i is at a single-hop from a server.

Single-hop Protocol

1. When f_i is 0, n_i broadcasts an one-hop *hello* packet to the adjacent MTs, periodically.
2. When a server, r, receives a *hello* packet, it replies a *hello_ack* packet with a quintet, $Q_r = [id_r, l_r, v_r, h_r, t_{event}]$. Here, h_r is 0.

3. When n_i receives the *hello_ack* packet, it calculates P_i based on the received Q_r (l_r and v_r), and its current location and vector by using Eqs. 4.2,4.3, and 4.4 . Then n_i obtains T_i using Eq. 4.5, sets f_i to 1, and caches Q_r. Also, n_i sets h_i to 1 by increasing h_r.

Note that if n_i does not move ($v_i = 0$), T_i is infinite. After setting f_i to 1, n_i stops broadcasting the *hello* packet. To keep a seamless connection with a server, we proactively broadcast a *hello* packet before t_{cur} becomes T_i. Otherwise, the MTs cannot response during the *hello* message interaction to estimate a new T, even though they are connected to a server. When n_i receives a *hello_ack* packet, it repeats step 3. If n_i does not receive a packet within a specified timeout period, n_i sets f_i to 0 and initiates to broadcast a *hello* packet periodically.

Next, we describe the protocol when an MT is multi-hop away from a server. Let us assume that an MT n_i and the server are within each other's communication range ($f_i = 1$), as shown in Figure 4.3, and n_j tries to decide whether it is accessible to the server.

Multi-hop Protocol

1. When f_j is 0, n_j broadcasts a *hello* packet.

2. When n_i receives a *hello* packet from n_j, it checks the f_i value to decide whether it can reply. Since f_i is 1, n_i replies a *hello_ack* packet including both Q_i and T_i to n_j. If f_i is 0, then n_i ignores the *hello* packet.

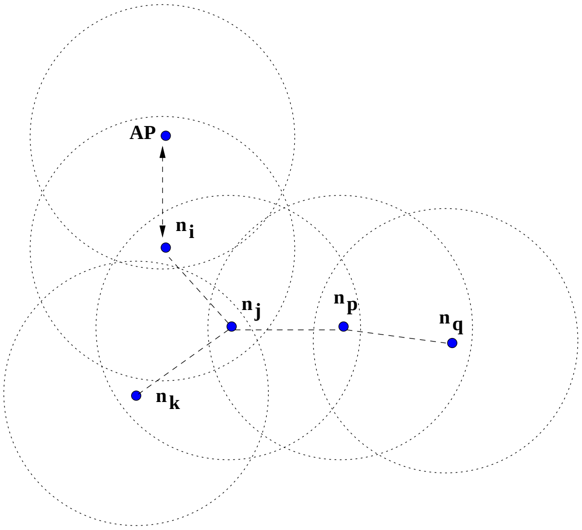


Fig. 4.3. A sample network topology.

3. When n_j receives a *hello_ack* packet from n_i, it calculates T_j in the same manner described in the single-hop protocol. Then n_j finds the minimum value of T_j by using Eq. 6 ($T_j = \text{Min}(T_i, T_j)$), caches the *hello_ack* packet sender's quintet (Q_i), sets f_j to 1, and stops broadcasting the *hello* packet.

Since n_j accesses the server through n_i, if n_i is out of the communication range of the server, n_j can no longer communicate with the server regardless of its connectivity with n_i. Also, n_j may receive multiple *hello_ack* packets, but it chooses the packet in a first come first serve (FCFS) manner.

Finally, we describe the protocol when an MT changes its direction that leads to the change of the position vector, and its connection expire time. When an MT is a single-hop away from a server, it recalculates T based on the server's location and vector that are fixed values, and its new location and vector. When an MT is multi-hop away from a server, it should recalculate T and broadcast a message to the MTs that access the server through it. Let us assume that n_i, n_j, n_k, n_p and n_q have connection with a server directly or indirectly through multi-hop relays (refer to Figure 4.3). Assume that, n_j changes its direction.

1. When n_j changes its direction, it updates its position vector and recalculates T_j. To recalculate T_j, n_j requires the location of n_i that replied a *hello_ack* packet before. For this, we utilize the Q_i cached in n_j to predict the location of n_i. Since n_j knows the vector and location of n_i at t_{event}, it can estimate the current location of n_i at t_{cur} ($i(t_{cur}) = (v_{i,x} \cdot (t_{cur} - t_{event}) + l_{i,x}, v_{i,y} \cdot (t_{cur} - t_{event}) + l_{i,y})$, where $l_{i,x}$ and $l_{i,y}$, and $v_{i,x}$ and $v_{i,y}$ are the location and vector of n_i, respectively.). Based

on the new location of n_i, n_j can recalculate T_j by using Eqs. 4.2, 4.3, and 4.4. Then n_j broadcasts an one-hop *loc_update* packet, attached with the updated Q_j to the MTs that rely on accessing the server through n_j (e.g. n_k, n_p and n_q).

2. When n_k receives a *loc_update* packet from n_j, it compares the *id* of the packet sender with the *id* cached in the quintet. If it matches, n_k recalculates T_k based on the received quintet by using Eqs. 4.2, 4.3, and 4.4. Then n_k chooses the minimum value ($T_k = \text{Min}(T_j, T_k)$), and caches the received quintet. Similarly, n_p and n_q update their Ts. If the *id* does not match, the *loc_update* packet is ignored.

When n_i receives a *loc_update* packet, it ignores the packet because any direction change does not affect n_i's connectivity to the server as long as it is in the communication range.

An MT may also determine its connectivity to a server using an alternative technique. For example, an MT keeps a list of one-hop neighbor by exchanging an one-hop *hello* message without using GPS information. In this approach, the MT should keep sending a message periodically to check whether it has access to the server, even though it is in the communication range of a server. Thus, this approach would generate high message traffic. In our scheme, we do not generate a *hello* packet if $f = 1$.

4.5 Cache Invalidation Strategies for IMANETs

In this section, we first discuss a *modified simple search (MSS)* algorithm for implementing the invalidation policy. Then we present the four cache invalidation schemes.

4.5.1 A Modified Simple Search Algorithm

In an IMANET, the data items from a server may be cached in some of the MTs. Thus, a search scheme needs to be deployed for finding a data item whether from the server or from an MT. In this chapter, for efficient search of queried data items, we propose a *modified simple search (MSS)* algorithm that is an extended version of a *simple search (SS)* scheme [51], in which the cache update is not considered. To reduce the query delay and number of communication messages, we use an aggregate cache concept in which a data item can be received from either the local cache of the MTs or from a server.

In the original SS scheme, when a query is generated, an MT broadcasts a request to all of the adjacent MTs. If an MT receives the request and has the data item in its local cache, it sends a reply to the requester to acknowledge that it has the data item. Otherwise it forwards the request to the neighbors. Thus, a request may be flooded in the network until it is acknowledged by a server and/or some MTs, which have cached copies of the queried data item. The sender thus can receive a data item from the closest MT or server, which is detailed below.

For implementing a cache invalidation scheme, when a query is generated which can be answered by a cached data item in an MT, the MT sends a message to a server to verify whether the cached data item is a valid copy. If the data item is invalid, the MT receives the valid copy from the server and uses it. Basically, the SS algorithm broadcasts a request to all neighboring MTs for a response. Since the cache update is only received from a server, in the MSS scheme, the broadcasted requests are sent only to the MTs, which are along the path closer to the server. This is different from the SS scheme, where a request is broadcasted to all neighboring MTs irrespective of the direction. The MSS scheme is thus a directed broadcast, and the direction can be obtained from the hop count, encoded in a packet. The MSS scheme can be implemented using any existing routing protocol. The detail steps are described below. Let us assume that an MT n_p generates a request for a data item as shown in Figure 4.4.

1. When a query is generated, n_p checks its local cache. If the queried data item is available, n_p immediately broadcasts a *request* packet to a server. A *request* packet contains the requester's id (id), id of the queried data item (d_s, where $0 \leq s < DB_{max}$), the most recently updated timestamp of d_s (t_s), and number of hops (h) from the server.

2. When n_j receives the request, it checks its local cache. If the queried data item is cached, n_j compares its updated timestamp with t_s attached in the packet. n_j updates t_s with the most recent value, and attaches its id to indicate that it has a cached copy. Then n_j forwards the request to the server to check the validity of the data item.

3. When n_q receives the *request* packet, it compares the current number of hops from the server with h attached in the packet. Since n_q is further away from the server, it does not forward the packet, and discards it. Thus, the MSS algorithm attempts to find the shortest path to the server.

4. When a server receives the *request* packet, it compares the updated timestamp of the data item stored in the database with t_s, attached in a packet. The server

replies with an *ack* packet attached with the response (either the updated data item or a valid bit telling that the data item is valid in the cache) to n_j, which subsequently sends it to the requester.

5. When n_p receives an *ack* packet, it uses the cached data item to answer the query, if it has a valid copy. Otherwise, it sends a *confirm* packet for a valid data item to n_j. Here, n_p may receive multiple *ack* packets, but it chooses the packet which arrives first. Since the packets are forwarded through different paths, n_j selects the path for a *confirm* packet based on the first receipt of the *ack* packet, and discards rest of the *ack* packets.

6. When n_j receives a *confirm* packet, it sends a *reply* packet to the *confirm* packet sender using the known route with the cached data item, which is already verified by a server. When a server receives a *confirm* packet, it attaches the valid data item to the *reply* packet.

When an MT forwards a *request* packet, the id of the MT is appended in the packet to keep the route information. Also, when an MT receives multiple copies of a *request* packet, due to broadcasting, it only processes one of them and discards the rest. For an *ack*, *confirm*, or *reply* packet, the MT also checks if its id is included in the path, which is appended to the packet. Since these packets are supposed to travel only along the assigned path that is established by the *request* packet, if the MT's id is not included in the path, the packet is discarded. When the server receives a *request* packet, it does not send the data item immediately, but sends an *ack* because other MTs, which are

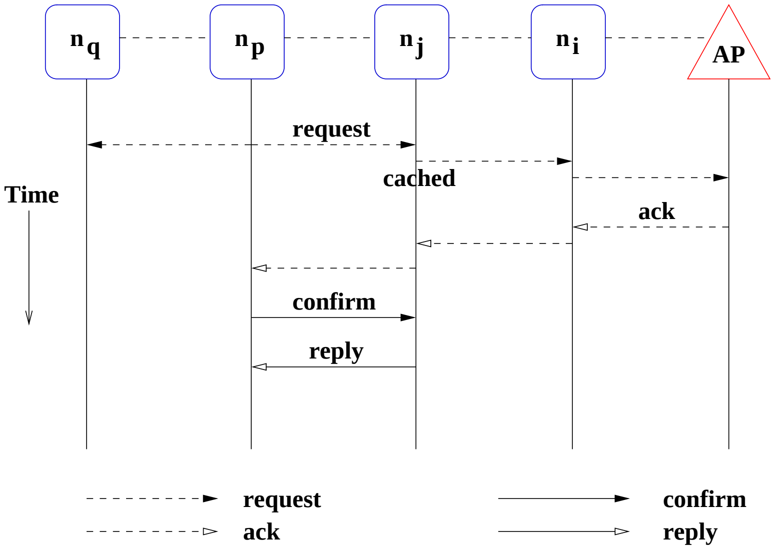


Fig. 4.4. A *modified simple search (MSS)* algorithm for querying data items in the IMANET. Let us assume that an MT (n_p) sends a request for a data item, and another MT n_j receives the request. n_j has a copy of the queried data item in its local cache. n_j sends a request to the server to check the validity of the data item. When the server receives the request, it verifies the status of the queried data item, and replies an *ack* packet. Then n_p sends a *confirm* packet to n_j, and n_j attaches the queried data item to the *reply* packet. Here, the dotted line between the MTs or an MT and an AP represents that they are located within the communication range.

located closer to the sender might cache the valid data item. This helps in reducing network congestion and bandwidth consumption by multiple replies.

4.5.2 An Aggregate Cache based On Demand (ACOD) Scheme

In contrast to the prior push-based schemes [11, 12, 32, 37, 40, 90], where the server periodically/non-periodically broadcasts a message, we propose a *pull-based* approach. In this approach, whenever a query is generated, an MT sends a message to a server to verify a cached data item before it is used for answering the query.

The basic mechanism is described as follows. When a query is generated, an MT checks its f value, which is obtained by the GPSCE scheme. If f is 0, which implies that the MT cannot access a server, the query fails. If f is 1, the MT checks its local cache, and broadcasts a *request* packet by using the proposed MSS algorithm. When an MT receives the request, it checks its f value. If f is 1, the MT checks its local cache, and forwards the packet to the MTs closer to the server. Thus, the sender receives either an acknowledgment from a server to indicate that its cached data item is valid or the queried data item from a server or an adjacent MT, who has the valid data item. The pseudo code for the ACOD scheme is given in Figure 4.5.

4.5.3 A Modified Timestamp (MTS) Scheme

In the TS scheme [11] proposed for cellular networks, a server periodically broadcasts an IR that carries the current timestamp (t_{cur}) and a list of tuples (d_s, t_s) such that $t_s > t_{cur} - w \cdot L$, where d_s is a data item id and t_s is the most recently updated timestamp of d_s ($0 \leq s < DB_{max}$). Here, w is an invalidation broadcast window and L

Notations:

d, t: A data item and its timestamp, respectively.
C_p: A local cache in MT n_p.
d^c: A cached data item.
t^c: A timestamp of the cached data item d^c.

```
(A) When $n_p$ generates a query $q$ for a data item $d$,
    if ($f_p == 1$) {
        if ($d \in C_p$) { /* the data item is locally cached. */
            Attach id of $n_p$, id of $d$, and $t^c$ to a request packet;
            Broadcast the request packet by using the MSS algorithm to the server/other MTs to validate the data item;
        }
        else /* the data item is not locally cached. */
            Broadcast a request packet by using the MSS algorithm to the server/other MTs for the data item;
    }
    else
        Fail to answer the query;

(B) When $n_j$ ($f_j = 1$) receives a request packet for data item $d$,
    if ($d \in C_j \;\wedge\; t < t^c$) {
        Attach id of $n_j$, and update $t$ with $t^c$ to the request packet;
        Rebroadcast the request packet by using the MSS algorithm to the server/other MTs to validate the data item;
    }
    else
        Rebroadcast the request packet by using the MSS algorithm to the server/other MTs for the data item;
```

Fig. 4.5. The pseudo code of the ACOD scheme.

is the broadcast interval. If an MT is in the active mode, it receives the IR and invalidates its cached data items accordingly. If an MT is disconnected or is in the sleep mode for more than $w \cdot L$, it invalidates the entire cached data items because it cannot decide which data items are valid. In this research, we do not consider the long disconnection problem for the sake of simplicity.

We modify the MTS scheme for IMANETS. The basic mechanism is described as follows. When a query is generated, an MT checks its f value. If f is 1, then the MT either waits for the next IR, or broadcasts a *request* packet by using the MSS algorithm. After the MT receives the IR, it invalidates the cached data items accordingly, and forwards the IR to adjacent MTs. If a queried data item is still valid, the MT uses it to answer the query. Otherwise, the MT broadcasts a *request* packet. When an MT receives a *request* packet, it forwards the packet to a server rather than waiting for the next IR. Although an MT can reply an *ack* packet while waiting for the next IR, it forwards the packet to a server. Otherwise, the query delay may increase. Here, we use a hop limit for an IR to prevent flooding of the IR packets over the network. An MT does not rebroadcast an IR, when it already has received the same IR, or the number of forwarded hops of the IR exceeds the hop limits. The pseudo code for the MTS scheme is shown in Figure 4.6.

A major drawback of the MTS scheme is the unavoidable query delay due to periodic broadcast of IRs.

Notations:

t_{cur}: Defined before.
Q_p: A query queue in MT n_p.
t_{last_bcast}: A timestamp of the last arrival time of IR.

```
(A) When $n_p$ generates a query $q$ for a data item $d$,
    if ($f_p == 1$) {
        if ($d \in C_p$)
            Queue $d$ into $Q_p$ and wait for next IR;
        else
            Broadcast a request packet by using the MSS algorithm to the server/other MTs for the data item;
    }
    else
        Fail to answer the query;

(B) When $n_p$ ($f_p = 1$) receives an IR,
    if ($t_{last\_bcast} < t_{cur} - w \cdot L$)
        Invalidate the entire cached data items;
    else {
        for  $\forall d \in$ IR  do {
            if ($t^c < t$)
                Invalidate $d^c$;
            else
                $t^c = t_{cur}$;
        }
    }
    for  $\forall d \in Q_p$  do { /* dequeue the data item. */
        if ($d \in C_p \;\wedge\; d^c$ is marked as valid)
            Use $d^c$ to answer the query;
        else
            Broadcast a request packet by using the MSS algorithm to the server/other MTs for the data item;
    }
    $t_{last\_bcast} = t_{cur}$;
    Rebroadcast the IR;

(C) When $n_j$ ($f_j = 1$) receives a request packet for the data item $d$,
    if ($d \in C_j \;\wedge\; t < t^c$) {
        Attach id of $n_j$, and update $t$ with $t^c$ to the request packet;
        Rebroadcast the request packet by using the MSS algorithm to the server/other MTs to validate the data item;
    }
    else
        Rebroadcast the request packet by using the MSS algorithm to the server/other MTs for the data item;
```

Fig. 4.6. The pseudo code of the MTS scheme.

4.5.4 An MTS with Updated Invalidation Report (MTS+UIR) Scheme

This scheme is the same as the MTS protocol except that an *updated invalidation report (UIR)*, which is partially derived from [12], is added. An UIR packet contains a timestamp of the last broadcasted IR from a server (t_{last_bcast}), and a list of tuples (d_s, t_s) that have been updated after the last IR has been broadcasted, where $t_s > t_{last\_bcast}$. The server inserts a number of UIRs into the IR intervals. Thus, an MT can process a query after receiving either an IR or an UIR. Here, we use four UIR replicates within each IR interval.

The MTS+UIR scheme follows the same procedure as the MTS scheme except followings. When an MT receives an UIR, it compares $t^c_{last\_bcast}$ saved in the local cache with t_{last_bcast} received from the UIR. If $t^c_{last\_bcast} > (t_{last\_bcast} - L)$ is true, the MT invalidates the cached data items accordingly, and answers the query or sends a *request* packet using the MSS algorithm. Otherwise, the MT ignores an UIR and waits for the next IR.

While this scheme reduces the query delay compared to the MTS scheme, it has higher message overhead than that of the MTS scheme due to additional UIR broadcasts.

4.5.5 An Asynchronous (ASYNC) Scheme

In this scheme, unlike the periodically broadcasted IR, a server broadcasts an IR whenever a data item is updated. If an MT has access to a server, it can decide the validity of current cached data items based on the broadcasted IR. Thus, when a query is generated, which can be answered by a cached data item, an MT does not contact the server for checking its validity. This approach is different from a prior scheme [40] proposed for cellular networks, where a stateful server broadcasts an IR asynchronously to the MTs that have the cached copy of the data item. However, the stateful approach is not feasible in IMANETS because of the high message overhead to maintain such information.

The mechanism of this scheme is described as follows. When a query is generated, an MT checks its f value. If f is 1 and the queried data item is found in its local cache, the MT uses the cached data item to answer the query. However, if f is 1 and the queried data item is not available in the local cache, the MT broadcasts a *request* packet by using the SS algorithm. Since an MT can response to the request, the ASYNC scheme uses the SS algorithm that searches the queried data item in any direction for better performance. Thus, the MT eventually receives the queried data item from a server and/or some closer MTs. The pseudo code for the ASYNC scheme is given in Figure 4.7.

When an MT is disconnected from a server, it can neither receive an IR nor answer a query. Since an IR is non-periodically broadcasted, an MT cannot ensure validity of a cached data item without connection with a server. Thus, when an MT is reconnected to the server, it sends a *cache_update* packet including a list of tuple (d_S, t_S) of all cached data items to a server for validation. When the server receives the packet, it replies a *cache_update_reply* packet including the data items that have been updated. When an MT receives the *cache_update_reply* packet, it updates its cache accordingly.

Although the ASYNC scheme can increase the throughput as well as reduces the query delay, an MT may use a stale data item to answer the query because it answers the query immediately without waiting for the IR from the server. For example, if an IR takes δ time to reach an MT and an MT answers a query during this time, then it

```
(A) When $n_p$ generates a query $q$ for a data item $d$,
    if ($f_p == 1$) {
        if ($d \in C_p$)
            Use $d^c$ to answer the query;
        else
            Broadcast a request packet by using the SS algorithm to the server/other MTs for the data item;
    }
    else
        Fail to answer the query.

(B) When $n_p$ ($f_p = 1$) receives an IR,
    Invalidate cached data items, accordingly;
    Rebroadcast the IR;

(C) When $n_j$ ($f_j = 1$) receives a request packet that requests $d$,
    if ($d \in C_j$)
        Reply an ack packet to the request packet sender by using the SS algorithm;
    else
        Rebroadcast the request packet by using the SS algorithm to the server/other MTs for the data item;
```

Fig. 4.7. The pseudo code of the ASYNC scheme.

may use a stale copy of the data item. To overcome the problem, an MT should wait for δ units of time (from the arrival of a request at time t) to ensure that it receives the validity of the data item before it replies. However, precise estimation of δ is extremely hard in an IMANET, and thus, the ASYNC scheme suffers from using stale copy.

In Figure 4.8, we show the successful query rate, including both valid or invalid data items, by varying both mean update intervals and query intervals. When both intervals are small, about 20% of the successful queries can get stale data items as shown in the figure. This is the due to the fact that, as the mean update and query intervals become smaller, an MT may cache more stale data items and use them for answering a query. As we expect, when both intervals increase, the successful queries answered by invalid data items reduce. Thus, the ASYNC scheme may work well for application, where both intervals are long enough so that the portion of successful query answered by invalid data item becomes negligible. Because of this problem, here, we do not compare the ASYNC scheme with other cache invalidation strategies, unless otherwise specified.

4.6 Performance Evaluation

4.6.1 Simulation Testbed

We use a discrete event network simulator *ns-2* with the CMU's wireless extension, to conduct the experiments. The radio model simulates Lucent's Technologies WaveLAN [84] with a nominal bit rate of 2Mbps and a nominal range of 250m. The radio propagation model is based on the free-space model [65]. For the link layer, the IEEE 802.11 MAC protocol and the distributed coordination function (DCF) are used.

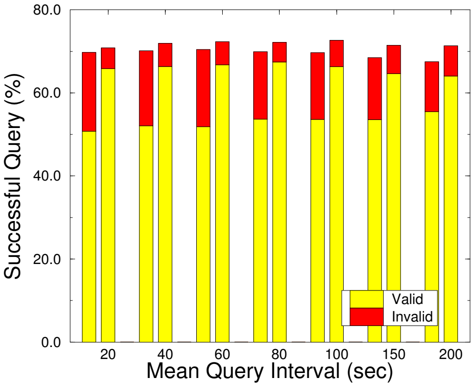


Fig. 4.8. Successful queries with different mean update intervals ($T_{update} = 20$ or 1000) as a function of mean query interval (With the ASYNC scheme $C_{size} = 10$, the first bar is for $T_{update} = 20$ and second bar is for $T_{update} = 1000$).

To examine the proposed idea, we use a 2000 × 750 m^2 rectangular area. We assume that an AP is located in the center of an area. The MTs are randomly located in the network. The speed of an MT is uniformly distributed from 0.0 to 2.0 (m/sec). The *random waypoint mobility* model, developed in [38], is used to simulate mobility here. With this approach, an MT travels toward a randomly selected destination in the network. After the MT arrives at the destination, it chooses a rest period (pause time) from a Uniform distribution. After the rest period, the MT travels towards another randomly selected destination, repetitively.

The query arrival pattern follows the Poisson distribution with a rate of λ. The update inter arrival time to the database is assumed to be exponentially distributed. The entire data items in the database are classified into two subsets: cold and hot data. We assume 80% of the update requests are for hot data items, and an update request is uniformly distributed within the hot and cold subsets. The least frequently used

(LRU) cache replacement policy is applied to the MT's local cache. To model the data item access pattern, we use the Zipf distribution [96] which is often used to model a skewed access pattern [12, 33], where θ is the access skewness coefficient. Setting $\theta = 0$ corresponds to the Uniform distribution. Here, we set θ to 0.9. The important simulation parameters are summarized in Table 4.1.

Table 4.1. Simulation parameters

Parameter	Values
Network size (m)	2000×750
Number of MTs	50
Number of APs	1
Database Size (items)	1000
Cache size (items/MT)	10 to 100
Data item size (KByte)	10
Mean query interval time (sec)	5 to 200
Mean update interval time (sec)	10 to 1000
Hot data items	1 to 50
Cold data items	remainder of DB
Hot data item update prob.	0.8
Broadcast interval (sec)	20
Broadcast window (w)	10 intervals
Pause time (sec)	100

4.6.2 Performance Parameters

We evaluate five performance parameters here: successful query rate (M_{suc}), average query delay (M_{delay}), average number of hop (M_{hop}), cache hit ratio (h_{all}) including local cache hit and remote cache hit, and average number of message (M_{msg}) for request.

The successful query rate, M_{suc}, is used to measure the accessibility of valid data items of the MTs in the IMANET. If r_{total} and r_{suc} denote the total number of queries and the number of successfully processed queries, then M_{suc} is defined as,

$$M_{suc} = \frac{r_{suc}}{r_{total}} \times 100\% \ .$$

The average query delay, M_{delay}, represents the average time to answer a query. If q_r denotes the delay for a successfully processed query r, then M_{delay} is expressed as,

$$M_{delay} = \frac{\sum_{r \in r_{suc}} q_r}{r_{suc}} \ .$$

The average number of hops, M_{hop}, is the distance to a server or an MT to check the validity of a data item. If a query is satisfied by a cached data item in the local cache, then M_{hop} is 0. If q_h denotes the number of hop for a successfully processed query r, then M_{hop} is expressed as,

$$M_{hop} = \frac{\sum_{r \in r_{suc}} q_h}{r_{suc}} \ .$$

The hit ratio h is used to evaluate the efficiency of the aggregate cache in our scheme. The remote cache hit is the ratio that a query is satisfied in other MTs' local cache. If n_{local} and n_{remote} denote the number of local hits and number of remote hits respectively, then h_{local}, h_{remote}, and h_{all} are expressed as:

$$h_{local} = \frac{n_{local}}{n_{local} + n_{remote}} \times 100\% \ ,$$

$$h_{remote} = \frac{n_{remote}}{n_{local} + n_{remote}} \times 100\% \ ,$$

$$h_{all} = \frac{n_{local} + n_{remote}}{r_{suc}} \times 100\% \ .$$

Finally, the average number of messages, M_{msg}, is counted to measure the overhead of an algorithm. Let n_{bcast}, n_{sent}, and $n_{forward}$ denote the total number of broadcasts, sent, and forwarded packets respectively. Then M_{msg} is expressed as,

$$M_{msg} = \frac{n_{bcast} + n_{sent} + n_{forward}}{r_{total}} \ .$$

4.6.3 Simulation Results

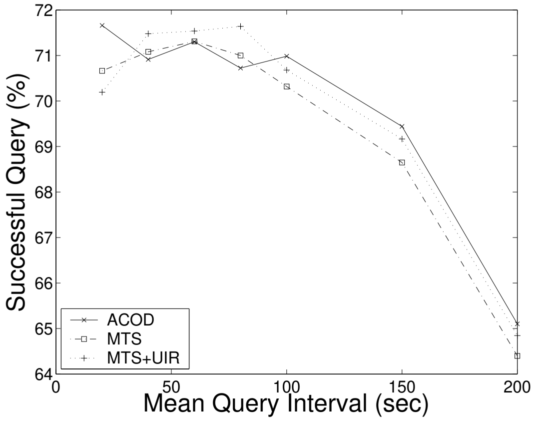
In this section, we evaluate the impact of query interval (T_{query}), cache update interval (T_{update}), size of the aggregate cache, and the search algorithm on the cache invalidation strategies. Also, the overhead of cache invalidation strategies is examined.

4.6.3.1 Impact of Query Interval

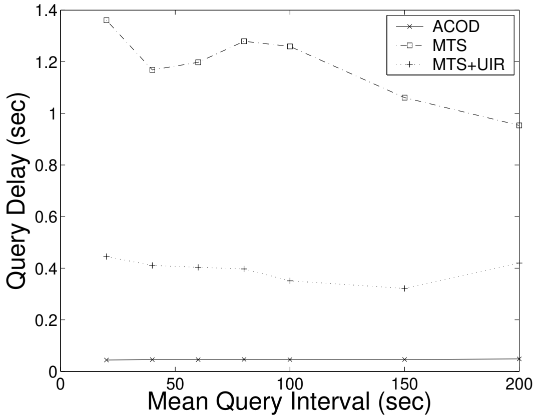
First, we evaluate the M_{suc}, M_{delay}, and M_{hop} of the cache invalidation strategies as a function of T_{query}. In Figure 4.9(a), as the query interval increases, the percent of successful query decreases. Since all the cache invalidation strategies answer a query after either verification of a cached data item or arrival of an IR, they show almost similar performance under the same query interval and network topology. Figure 4.9(b) shows the impact of T_{query} on the query delay. The ACOD scheme shows the lowest M_{delay}, followed by the MTS+UIR and MTS schemes. The MTS scheme shows the longest M_{delay}, due to the long IR broadcast interval. In the MTS+UIR scheme, M_{delay} is reduced compared to that of MTS due to UIRs, which arrive during an IR broadcast interval. Figure 4.9(c) shows the impact of T_{query} on the number of hops. The ACOD scheme shows slightly lower M_{hop} than that of the MTS and MTS+UIR schemes. Since a data item can be received from an intermediate MT, the number of hops is less compared to the IR-based scheme.

4.6.3.2 Impact of Update Interval

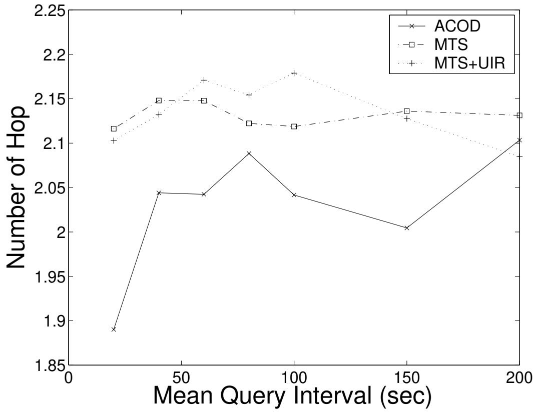
Next, we examine the impact of T_{update} on M_{suc}, M_{delay}, and M_{hop}. In MTS and MTS+UIR schemes, from Figure 4.10(a), M_{suc} increases as the update interval increases. However, the ACOD scheme shows little change of M_{suc} over the entire update intervals, because it checks validity of a cached data item or receives a queried data item from the server, and thus, its performance is not affected by T_{update} much. In Figure 4.10(b), the ACOD scheme shows the lowest M_{delay} over the entire update intervals, followed by the MTS+UIR and MTS schemes. As the update interval increases, the M_{delay} of both MTS and MTS+UIR schemes is increased. Since less number of cached data items are updated at a server, an MT finds more number of cached data items marked as valid. Thus, in the MTS and MTS+UIR schemes, the MT waits for the next IR or UIR arrival for validation frequently, and query delay increases. The MTS scheme has the longest M_{delay}, because it has longer interval to receive the next IR than that of the MTS+UIR scheme. In Figure 4.10(c), as the update interval increases, M_{hop} of all the schemes is proportionally reduced, because more number of cached data items are valid, and thus an MT receives more queried data items from MTs rather than from a server.



(a) Successful query

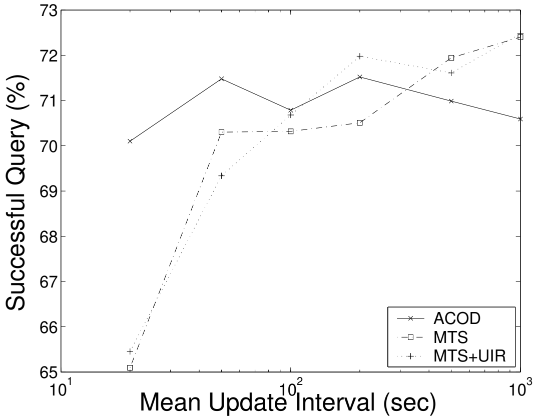


(b) Query delay

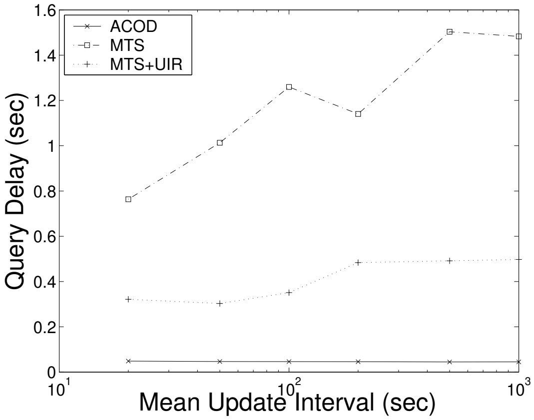


(c) Number of hop

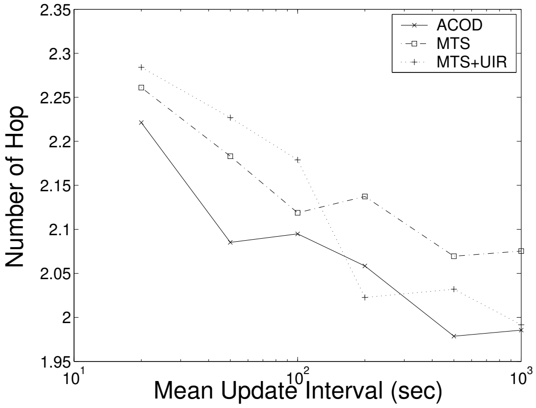
Fig. 4.9. Successful query rate, query latency, the number of hops as a function of mean query interval ($T_{update} = 100$, and $C_{size} = 10$).



(a) Successful query



(b) Query delay



(c) Number of hop

Fig. 4.10. Successful query rate, query delay, number of hops as a function of mean update interval ($T_{query} = 100$, and $C_{size} = 10$).

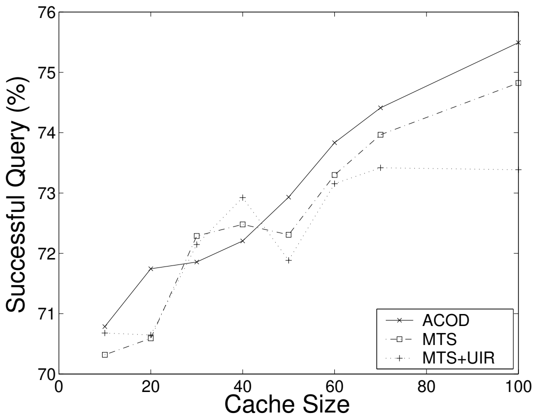
4.6.3.3 Impact of Aggregate Cache

We observe the impact of aggregate cache on M_{suc}, M_{delay}, and M_{hop} by varying the cache size. In Figure 4.11(a), as the cache size increases, M_{suc} of all the schemes is increased. Because more number of queried data items are answered from the local cache or remote caches. In Figure 4.11(b), the MTS scheme shows the longest M_{delay}, followed by the MTS+UIR and ACOD schemes. With a larger cache, as more number of data items are found in the cache, an MT waits for the next IR for validation. The MTS+UIR scheme reduces the M_{delay}, because an UIR arrives during the IR broadcast interval. In Figure 4.11(c), as the cache size increases, the ACOD scheme results in the minimum hop count (M_{hop}). Since more data items can be received from an intermediate MT, the number of hops is much smaller than that of the IR-based scheme.

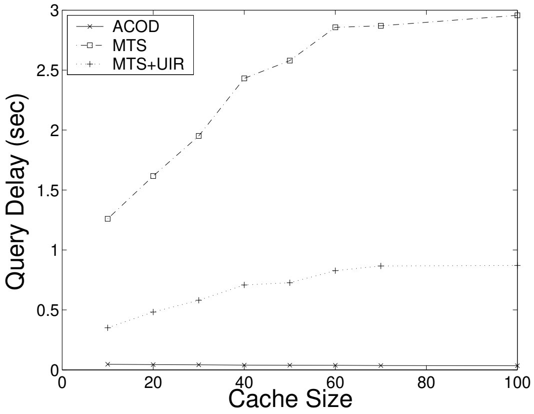
In Figure 4.12, we plot M_{suc} contributed due to only cache hits including local and remote hits. As the T_{update} increases, more cached data items remain valid, and thus, the cache hits of all the schemes increase. As we expect, from Figures 4.12(a) and (b), the cache hits increase as the C_{size} increases. The ACOD scheme shows higher M_{suc} than that of the MTS and MTS+UIR schemes, because it actively searches a queried data item during validation of a cached data item. In summary, the aggregate cache increases the cache hit, and thus, overall M_{suc} increases.

4.6.3.4 Impact of Search Algorithm

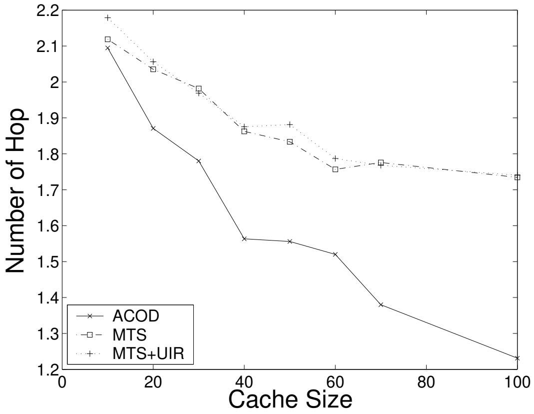
We compare the proposed MSS algorithm with the SS algorithm, and evaluate their performances in terms of M_{msg}, M_{suc}, and M_{delay}. Figure 4.13 shows the impact of MSS and SS algorithms on M_{msg}, M_{suc}, and M_{delay} for different cache invalidation



(a) Successful query

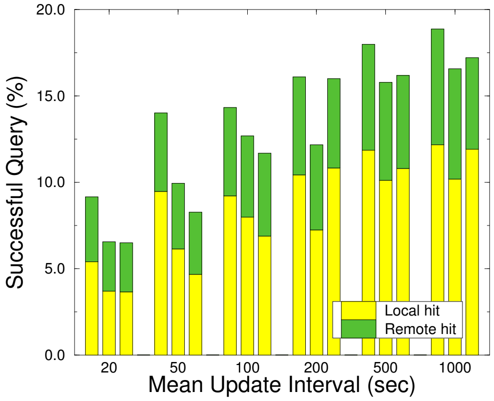


(b) Query delay

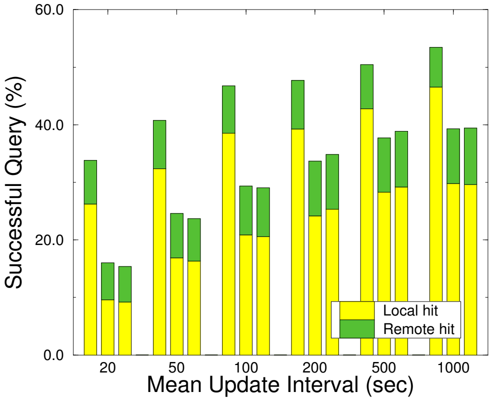


(c) Number of hop

Fig. 4.11. Successful query rate, query delay, number of hops as a function of cache size ($T_{query} = 100$, and $T_{update} = 100$).



(a) $C_{size} = 10$



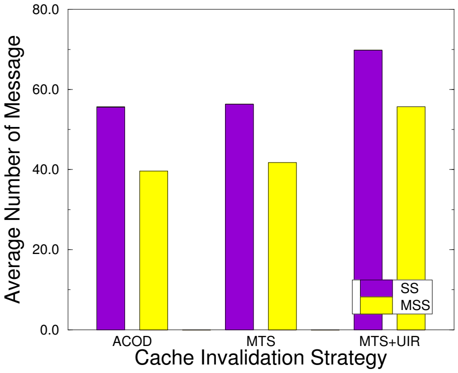
(b) $C_{size} = 100$

Fig. 4.12. Successful query rate in terms of average cache hit ratios is shown against mean update interval (T_{query} = 100, and C_{size} = 10 or 100). The ACOD, MTS, and MTS+UIR schemes are plotted from left to right.

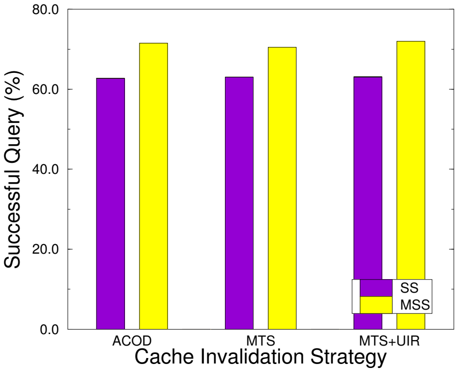
strategies. In Figure 4.13(a), when the MSS algorithm is used, M_{msg} is reduced for all the cache invalidation strategies. In the MSS algorithm, unlike the SS algorithm, a request packet is broadcasted toward the MTs, which are located along the path/closer to the server. Thus, the MSS algorithm eliminates unnecessarily broadcasted messages. In Figures 4.13(b) and (c), since the SS algorithm generates more number of messages over the network, it may cause traffic congestion and larger message delay. The MSS algorithm achieves higher M_{suc} and lower M_{delay} than that of the SS algorithm for all the cache invalidation strategies. In summary, the proposed MSS algorithm reduces the number of messages and increases communication performance, and thus, it is suitable for implementing cache invalidation strategies for IMANETS.

4.6.3.5 Message Overhead

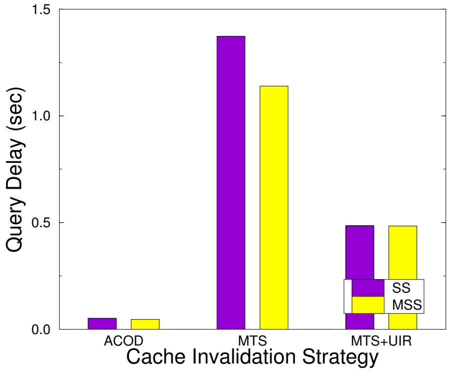
Finally, we evaluate the overhead of each cache invalidation strategy in terms of average number of messages, including broadcast, sent, and forwarded messages from the server and MTs. Figure 4.14 shows the impact of update interval on M_{msg} with different cache sizes. The MTS+UIR scheme shows the highest M_{msg} because both IR and UIR are periodically broadcasted over the network, regardless of the update interval. When the update interval is small ($T_{update} = 20$), the ASYNC scheme has the high M_{msg}, because an IR is frequently broadcasted. Note that the ASYNC scheme shows the lowest M_{msg}, when the update interval is very large ($T_{update} = 1000$). This is due to the fact that an IR is less frequently broadcasted, and less number of messages is used because more number of queried data items are received from the MTs. The ACOD scheme shows the lowest M_{msg} for the entire update intervals.



(a) Average number of message



(b) Successful query



(c) Query delay

Fig. 4.13. Average number of message, successful query rate, and query delay with different search algorithms are shown against different cache invalidation strategies (T_{query} = 100, T_{update} = 200, and C_{size} = 10).

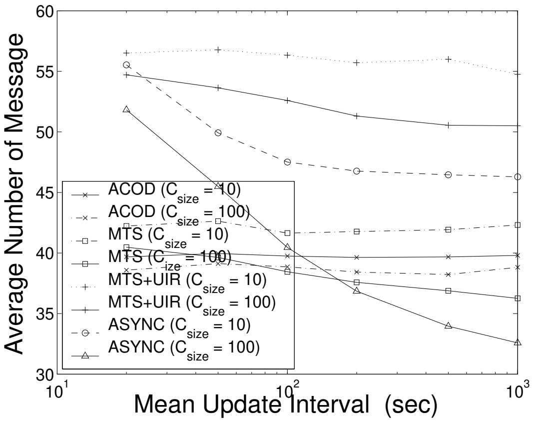


Fig. 4.14. Average number of messages as a function of mean update interval (T_{query} = 100, and C_{size} = 10 or 100).

4.7 Concluding Remarks

In this chapter, we investigated the cache invalidation problem in an IMANET, where the issues are different from that in a cellular network due to multi-hop message relay, operation cost model, and uncertainly in message delivery. In light on this, several push and pull-based cache invalidation strategies are carefully compared and analyzed. First, we proposed a *GPS-based connectivity estimation (GPSCE)* scheme as a base algorithm to support any cache invalidation strategy in IMANETS. Then, we proposed a *pull-based* cache invalidation policy, called ACOD. In addition, we extended three prior *push-based* cache invalidation strategies to work in IMANETS.

We conducted a simulation based performance study to examine the performance of different cache invalidation strategies. In the MTS and MTS+UIR schemes, there is an unavoidable delay before processing a query, since a request has to wait for the next IR/UIR. Also the message overhead due to broadcasted IR/UIRs leads to waste of wireless bandwidth. The ASYNC scheme suffers from receiving the updated/valid data in a given time (δ) due to the mobility and disconnection in an IMANET. Thus, it may supply a stale copy depending on update frequency. In addition, it also suffers from high message overhead. The ACOD scheme provides high throughput (successful query), lowest query delay, and minimal communication overhead for different workload configurating. Thus, it is a viable approach for implementation in IMANETS.

For the future work, we will investigate the power issue of different cache invalidation strategies. Due to multi-hop IRs and message relays, it needs to schedule the sleep and wake up modes of the MTs judiciously during the IR interval without degrading the performance. We will analyze these details for more realistic evaluation.

Chapter 5

A Randomized Communication Scheme for Energy Efficient MANETs

In this chapter, we propose a new communication mechanism that integrates the IEEE 802.11 power saving mode (PSM) and the dynamic source routing (DSR) protocol for supporting energy-efficient communication in Manets.

5.1 Introduction

One of the most critical issues in *mobile ad hoc networks* (MANETs) is energy conservation. Since mobile nodes usually operate on batteries, a prudent power saving mechanism (PSM) is required to guarantee a certain amount of device lifetime. It also directly affects the network lifetime because mobile nodes themselves collectively form a network infrastructure for routing in a MANET. Energy efficiency can be improved in two different ways: reducing the energy used for active communication activities and reducing the energy spent during an inactive period. *Power-aware routing* [16, 76, 88, 89] and *transmit power control* (TPC)-based algorithms [7, 27, 39, 44] are examples of the first category. However, more energy saving can be obtained by carefully designing the behavior during the inactive period because idle listening during an inactive period consumes almost the same energy as during transmission or reception, and it usually accounts for the most part of the total energy consumption [71]. Therefore, it is important to turn off the radio or switch it to a low-power sleep state to maximize the energy efficiency. For this reason, many radio hardwares support low-power states, where substantially less amount of energy is consumed by limiting the normal communication activity [41]. For instance, the Lucent IEEE 802.11 WaveLAN-II consumes 1.15W and 0.045W in the idle-listening and low-power state, respectively [41] and the radio transceiver TR 1000 [3], used in Berkeley Motes [2], consumes 13.5mW and 0.015 mW, respectively. 25 to 900 times difference in energy consumption has motivated the use of PSM in energy-constrained mobile environments.

The IEEE 802.11 standard, which is the most popular wireless LAN standard, exploits this hardware capability to support the power management function in its *medium access control* (MAC) layer specification [5]. Each mobile device can be in one of the two power management modes: *active mode* (AM) or *power save* (PS) mode. A device in the PS mode periodically wakes up during the packet advertisement period, called *Ad hoc (or Announcement) Traffic Indication Message* (ATIM) window to see if it has any data to receive. It puts itself into the low-power state if it is not addressed, but stays awaken to receive any advertised packet otherwise. However, this IEEE 802.11 PSM is difficult to employ in a multihop MANET because of routing complexity not alone the difficulty in synchronization and packet advertisement in a dynamic distributed environment [17, 83].

The main goal of this research is to make the IEEE 802.11 PSM applicable in multihop MANETs when the popular *Dynamic Source Routing* (DSR) [38] is used as the network layer protocol. A major concern in integrating the DSR protocol with the IEEE 802.11 PSM is *overhearing.* Overhearing improves the routing efficiency in DSR by eavesdropping other communications and gathering route information. It incurs no extra cost if all mobile nodes operate in the AM mode because they are always awake and idle listening anyway. However, if mobile nodes operate in the PS mode, it brings on a high energy cost because they should not sleep but receive all the routing and data packets transmitted in their vicinity. A naive solution is to disable overhearing and let a node receive packets only if they are destined to it. However, it is observed that this solution reduces network performance significantly because each node gathers less route information due to the lack of overhearing, which in turn incurs a larger number of broadcast flooding of *route request* (RREQ) messages resulting in more energy consumption[1]. In short, overhearing plays an essential role in disseminating route information in DSR but it should be carefully re-designed if energy is a primary concern.

Here, we propose a message overhearing mechanism, called *RandomCast or Rcast*, via which a sender can specify the desired level of overhearing when it advertises a packet. Upon receiving a packet advertisement during an ATIM window, a node makes its decision whether or not to overhear it based on the specified overhearing level. If *no overhearing* is specified, every node decides not to overhear except the intended receiver and if *unconditional overhearing* is specified, every node should decide to overhear. *Randomized overhearing* achieves a balance somewhere in between, where each node makes its decision probabilistically based on network parameters such as node density and network traffic. Rcast helps nodes conserve energy while maintaining a comparable set of route information in each node. Since route information is maintained in the *route cache* in DSR, Rcast effectively avoids unnecessary effort to gather redundant route information and thus saves energy. The key idea behind the Rcast scheme is to explore the temporal

---

[1] This is what happens with Ad-hoc On-demand Distance Vector (AODV) [62], which is another popular on-demand routing algorithm. AODV takes a conservative approach to gather route information: It does not allow overhearing and eliminates existing route information using timeout. However, this necessitates more RREQ messages. According to Das et al., 90% of the routing overhead comes from RREQ in [22].

and spatial locality of route information, as is done in the CPU cache. Overheard route information will probably be overheard again in the near future and thus it is possible to maintain the same quality of route information, while overhearing only a small fraction of packets. Even though a node misses a particular route information, it is highly probable that one of its neighbors overhears it and can offer the information when the node asks for it. Note that we have chosen DSR in this chapter because other MANET routing algorithms usually employ periodic broadcasts of routing-related control messages, such as link states in *table driven protocols* or *Hello* messages in AODV [62], and thus tend to consume more energy with the IEEE 802.11 PSM.

Key contributions of this research chapter are three-fold:

- First, it considers integration of DSR and IEEE 802.11 PSM, in which nodes consistently operate in the PS mode. This has not been studied elsewhere in the literature to the best of the authors' knowledge.
- Second, we observe that there is a semantic discrepancy when a node transmits a unicast packet but it wishes that all its neighbors overhear it. Rcast mechanism addresses this problem by providing a way to clearly specify who should receive and who should overhear.
- Third, we study and pose an open question on the routing efficiency in the context of power management rather than for the sake of routing algorithm itself. Limited overhearing with Rcast consumes less energy but it is questionable that conventional route caching strategies would work well and maintain a rich set of route information.

The performance of the proposed Rcast scheme is evaluated using the ns-2 network simulator [4]. According to the simulation results, the proposed algorithm reduces the energy consumption as much as 236% and 131% compared to the original IEEE 802.11 PSM and *On-Demand Power Management* (ODPM) [95] protocol, which is one of the competitive schemes developed for multihop networks, respectively, at the cost of at most 3% reduction in packet delivery ratio. It is also found that Rcast improves the energy balance among the nodes and increases the network lifetime. Simulation results indicate that variance of energy consumption of nodes is four times higher in ODPM than in Rcast. Note that the concept of Rcast can also be applied to broadcast messages in order to avoid redundant rebroadcasts, as studied by Ni et al. [82] and thus can usefully be integrated with other MANET routing algorithms.

The rest of the chapter is structured as follows. Section 5.2 presents the background and the related work on the PSM of IEEE 802.11 and DSR routing protocol. Section 5.3 describes the proposed Rcast scheme and its use with DSR, while Section 5.4 is devoted to extensive performance analysis. Section 5.5 draws conclusions and presents future directions of this study.

5.2 Background and Related Work

We assume that mobile nodes operate as the IEEE 802.11 PSM for energy-efficient medium access and use DSR for discovering and maintaining routing paths. Section 5.2.1 summarizes the DSR routing protocol with an emphasis on *route cache*. It also discusses the stale route and load unbalance problem in DSR and argues that unconditional overhearing is the main reason behind them. Section 5.2.2 explains the IEEE 802.11 PSM and previous research work on its use in multihop networks.

5.2.1 DSR Routing Protocol

5.2.1.1 Route discovery and Maintenance

Route cache is one of the most important data structures in DSR and is used to reduce the routing-related control traffic. When a node has a data packet to send but does not know the routing path to the destination, it initiates the route *discovery procedure* by broadcasting a control packet, called *route request packet* (RREQ). When a RREQ reaches the destination, it prepares another control packet, called *route reply packet* (RREP) and replies back to the source with the complete route information. Upon receiving a RREP, the source begins transmitting the data packets to the destination, and it saves the route information in its route cache for later use.

Since data transmission in wireless networks is broadcast in nature, intermediate relaying nodes as well as other nearby nodes also learn about the path to the destination via overhearing. Therefore, the number of RREQ packets can be minimized because a node may have cached the path to a destination in its route cache. Route caching reduces the number of RREQ packets even further by allowing an intermediate node to reply to a RREQ if it has the destination route information. This mitigates network-wide flooding of RREQ packets and also saves energy significantly.

Since nodes move randomly in a MANET, link errors occur and a route information that includes a broken link becomes obsolete. When a node detects a link error during its communication attempt, it sends a control packet, called *route error packet* (RERR), to the source and deletes the stale route from its route cache. In addition. RERR informs nearby nodes about the faulty link so that they can also delete the path including the broken link.

5.2.1.2 Stale Route Problem in DSR

However, since link errors (or RERR) are not propagated “fast and wide”, as pointed out by Marina and Das [57], route caches often contain stale route information for an extended period of time. In addition, the erased stale routes are possibly un-erased due to in-flight data packets carrying the stale routes. When a node has an invalid route in its route cache or receives a RREP that contains an invalid route, it would attempt to transmit a number of data packets without success while consuming energy. Hu and Johnson studied design choices for route cache in DSR and concluded that there must be a mechanism, such as *cache timeout*, that efficiently evicts stale route information [34].

While the main cause of the stale route problem is node mobility, it is unconditional overhearing that dramatically aggravates the problem. This is because DSR generates more than one RREP packets for a route discovery to offer alternative routes in addition to the primary route to the source. While the primary route is checked for its validity during data communication between the source and the destination, alternative routes may remain in route cache unchecked even after they become stale. This is the case not only for the nodes along the alternative routes, but also for all their neighbors because of unconditional overhearing.

5.2.1.3 Load Unbalance Problem in DSR

On-demand routing algorithms such as DSR exhibit another undesirable characteristic, called load unbalance. In a multihop mobile network, each node plays an important role as a router to forward packets on other nodes' behalf. In an ideal case, each mobile node takes equal responsibility of packet forwarding to others. However, it is observed that this is not usually the case and several inconsistencies may arise due to over-dependence of packet forwarding functionality on a few overloaded nodes [93]. For example, overloaded nodes can exhaust their battery power much faster than other nodes and critically decrease the network lifetime.

It is argued in [93] that the non-uniform load distribution is caused primarily by "*preferential attachment*" [10] in the dynamics of route information construction in route caches, together with the *expand ring search* algorithm used in DSR protocol. For example, suppose a node, say node S, has route information to a number of destination nodes. When a neighboring node, say node T, wishes to discover a route to one of those destinations, node S would supply the desired information to node T. Thus, node S becomes an intermediate node for the route from node T to the destination and node S will have an additional entry from S to node T. In other words, an overloaded node becomes more overloaded with time due to unconditional overhearing.

5.2.2 Power Saving Mechanism (PSM) in IEEE 802.11

In this Subsection, we assume that the network layer software, DSR, interacts with a lower layer protocol conforming to the IEEE 802.11 standard [5]. According to its specification, there are two modes of operation depending on the existence of an access point (AP). These are referred to as the *Distributed Coordination Function* (DCF) and the *Point Coordination Function* (PCF). The DCF uses a contention algorithm to provide access to all traffic based on the *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) and delay, known as *InterFrame Space* (IFS). The PCF is an optional access method implemented on top of the DCF and provides a contention-free service coordinated by an AP. In the PCF, an AP has a higher access priority than all other mobile nodes, and thus, it not only sends downlink packets but also controls uplink packets by polling stations [78].

5.2.2.1 IEEE 802.11 PSM in One-hop Networks

Power saving in PCF mode is achieved by the coordination of the AP, which operates in AM. The AP periodically sends beacon signals to synchronize other mobile nodes that operate in the PS mode and informs them whether they have packets to receive or not using the *Traffic Indication Map* (TIM), which is included in the beacon in the form of a bitmap vector. If a node is not specified as a receiver in the TIM, it switches off its radio subsystem during the data transmission period.

In the DCF, power saving is more difficult to achieve. In the absence of an AP, nodes in the PS mode should synchronize among themselves in a distributed way[2]. In addition, a beacon does not contain the TIM any more and each sender should advertise its own packet by transmitting an ATIM frame during the packet advertisement period,

---

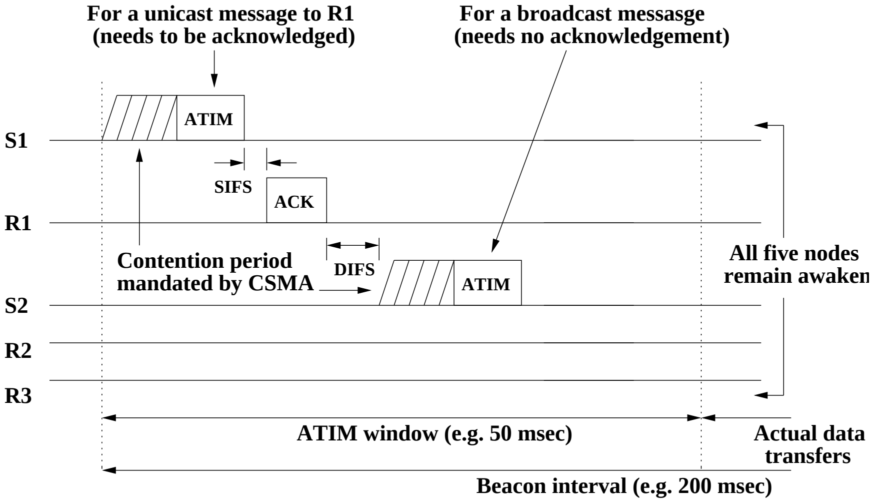
[2] Tseng et al. [83] and Huang and Lai [35] studied the clock synchronization problem. We do not discuss this issue in detail in this paper and assume that all mobile devices operate in synchrony using one such algorithm.

called *ATIM window*. Each packet is buffered at the sender and is directly transmitted to the receiver during the data transmission period.

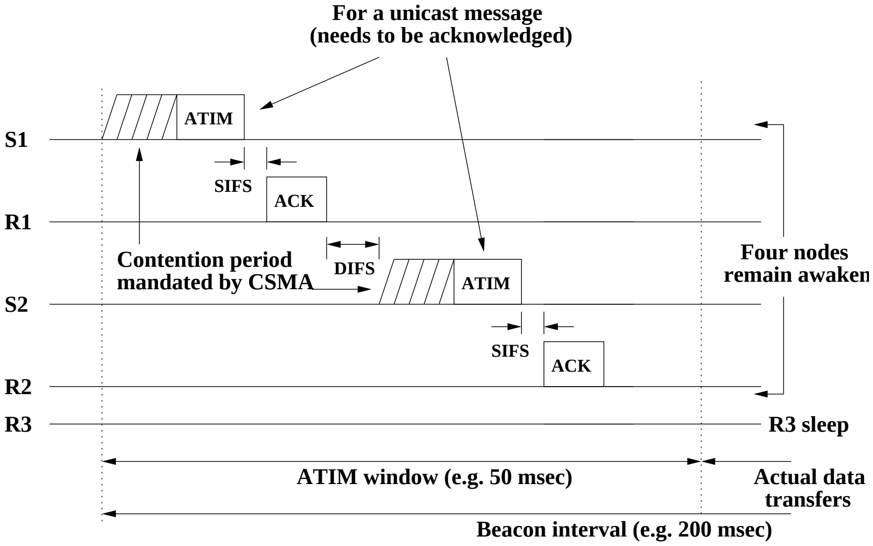
Fig. 5.1 shows the PSM protocol in the DCF with an example mobile network of five nodes, S_1, R_1, S_2, R_2, and R_3. In Fig. 5.1(a), node S_1 has a unicast packet for node R_1 and node S_2 has a broadcast packet. They advertise them during the ATIM window. Note that nodes S_1 and S_2 compete with each other using the same CSMA/CA principle for transmitting the ATIM frames. Node S_1 needs acknowledgment from node R_1 but node S_2 does not. In this scenario, all five nodes remain awaken during the data transmission period in order to receive the unicast and/or broadcast packets. Consider another example in Fig. 5.1(b). Here, node S_2 also has a unicast packet to R_2, and thus nodes S_1, R_1, S_2, and R_2 must be awaken, but node R_3 can switch to the low-power sleep state immediately after the ATIM window, because it does not have any packet to receive. It is important to note that node R3 should remain awaken if the unconditional overhearing is used.

5.2.2.2 IEEE 802.11 PSM in Multihop Networks

Note that PSM in both PCF and DCF assumes that every node is within every other's radio transmission range. Thus, they are not directly applicable in multihop mobile networks. Recently, this problem has been addressed by a number of research groups. SPAN [17] mandates a set of nodes to be in AM, while the rest of the nodes stay in the PS mode. AM nodes offer the routing backbone so that any neighboring node can transmit to one of them without waiting for the next beacon interval. A drawback of this scheme is that it usually results in more AM nodes than necessary and degenerates



(a) One unicast and one broadcast message (all five nodes remain awaken during the entire beacon interval)



(b) Two unicast messages (all nodes except node R_3 should remain awaken during the beacon interval)

Fig. 5.1. IEEE 802.11 PSM (SIFS: Short IFS, and DIFS: DCF IFS).

to all AM-node situation when the network is relatively sparse. More importantly, it does not take the routing overhead into account because it uses the geographic routing and assumes that location information is available for free. This is neither realistic nor compatible for use with DSR or AODV as pointed out in [85].

Zeng and Kravets suggested a similar approach, called *On-Demand Power Management* (ODPM) [95], in which a node switches between the AM and PS modes based on communication events and event-induced timeout values. For example, when a node receives a RREP packet, it is better to stay in AM for more than one beacon interval (timeout) hoping that there will be more data packets to be delivered in the near future. This scheme asks for each node to switch between the AM and PS modes frequently, which may incur non-negligible overhead. It may reduce the packet delay by transmitting data packets immediately if the receiver is believed to be in AM. However, obtaining neighbors' power management mode is not trivial. This requires either an additional energy cost to obtain it or an extended packet delay if it is not accurate. Also, its performance greatly depends on timeout values, which need fine tuning with the underlying routing protocol as well as traffic conditions. For example, consider that a node stays in AM for five consecutive beacon intervals upon receiving a data packet as is assumed in [95]. If data traffic occurs infrequently, say once every six beacon intervals, the node stays in AM for five intervals without receiving any further data packets and switches to low-power sleep state. It receives the next data packet while operating in the PS mode, and thus, decides again to stay five intervals. Packet delay is not affected but it consumes more energy than unmodified IEEE 802.11 PSM.

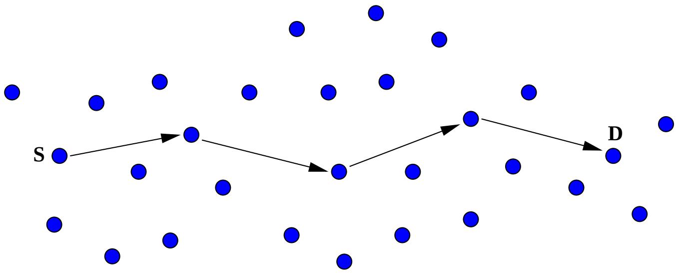
5.3 Randomized Overhearing Using Rcast

This Section describes the *RandomCast* or *Rcast*-based communication mechanism, aimed at improving the energy performance by controlling the level of overhearing without a significant impact on network performance. Compared to the algorithms in Section 5.2.2, the proposed scheme assumes that the mobile nodes consistently operate in the the PS mode and employ the DSR routing algorithm [38]. Section 5.3.1 presents the basic idea of Rcast and its advantages. Sections 5.3.2 and 5.3.3 discuss the implementation of Rcast and its integration with DSR, respectively.

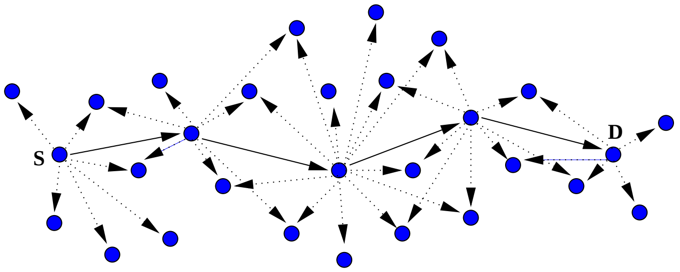
5.3.1 No, Unconditional, and Randomized Overhearing

As explained in Section 5.2.2, a unicast packet is delivered only to an intended receiver if the IEEE 802.11 PSM is employed. Consider that a node S transmits packets to a node D via a pre-computed routing path with three intermediate nodes as shown in Fig. 5.2(a). Only five nodes are involved in the communication and the rest would not overhear it (*no overhearing*). However, if each neighbor is required to overhear as in DSR, each sender should be able to "broadcast" a unicast message. i.e., it specifies a particular receiver but at the same time asks others to overhear it as shown in Fig. 5.2(b) (*unconditional overhearing*).

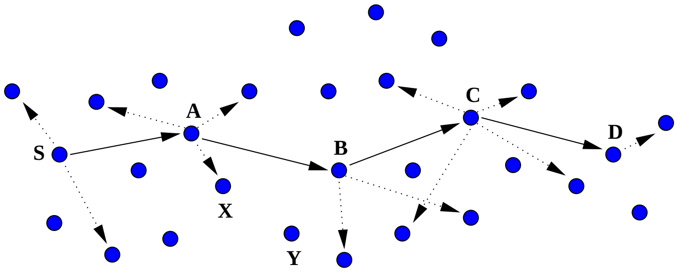
Randomized overhearing adds one more possibility in between unconditional and no overhearing. As shown in Fig. 5.2(c), some of the neighbors overhear, but others do not and these nodes switch to the low-power state during the data transmission period. Randomized overhearing saves substantial amount of energy compared to unconditional



(a) No overhearing



(b) Unconditional overhearing



(c) Randomized overhearing

Fig. 5.2. Delivery of a unicast message with different overhearing mechanisms.

overhearing. With respect to the route information, it does not deteriorate the quality of route information by exploiting the spatial and temporal locality of route information dissemination as explained in the introduction. Consider an example in Fig. 5.2(c), in which nodes X and Y are two neighbors of the communicating nodes A and B. Their communication and overhearing activities are drawn in Fig. 5.3. When node A receives a RREP from node B, it obtains a new route ($S \rightarrow D$) and stores it in its route cache. Nodes X and Y do not overhear the RREP as shown in the figure but, since there will be a number of data packets transferred from node A to B, they will obtain the route information ($S \rightarrow D$). In this figure, node X overhears the second data packet and node Y overhears the second from the last packet. Fig. 5.3 also shows when the route becomes stale and gets eliminated from the route cache.

5.3.2 Rcast Implementation

The Rcast mechanism enables a node to choose no, unconditional, or randomized overhearing when it has a unicast packet to send. Its decision can be specified in the ATIM frame so that it is available to its neighboring nodes during the ATIM window. For practicality, implementation in the context of IEEE 802.11 specification is considered by slightly modifying the ATIM frame format as shown in Fig. 5.4. An ATIM frame is a management frame type and its subtype ID is 1001_2. The Rcast mechanism utilizes two reserved subtype IDs, 1101_2 and 1111_2, to specify randomized and unconditional overhearing, respectively. An ATIM frame with subtype 1001_2 is recognized as no overhearing, and thus, conforms to the standard.

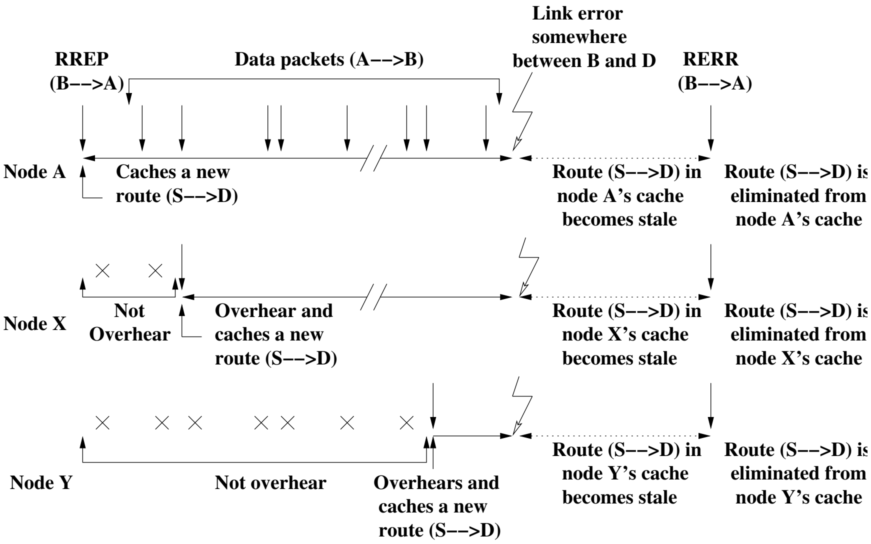


Fig. 5.3. Lifetime of route information at an intermediate node A and neighbor nodes X and Y.

Consider an example when a node (its MAC address MA) wakes up at the beginning of a beacon interval and receives an ATIM frame. It decides whether or not to receive or overhear the packet based on the destination address (DA) and subtype ID. It would remain awaken if one of the following conditions is satisfied.

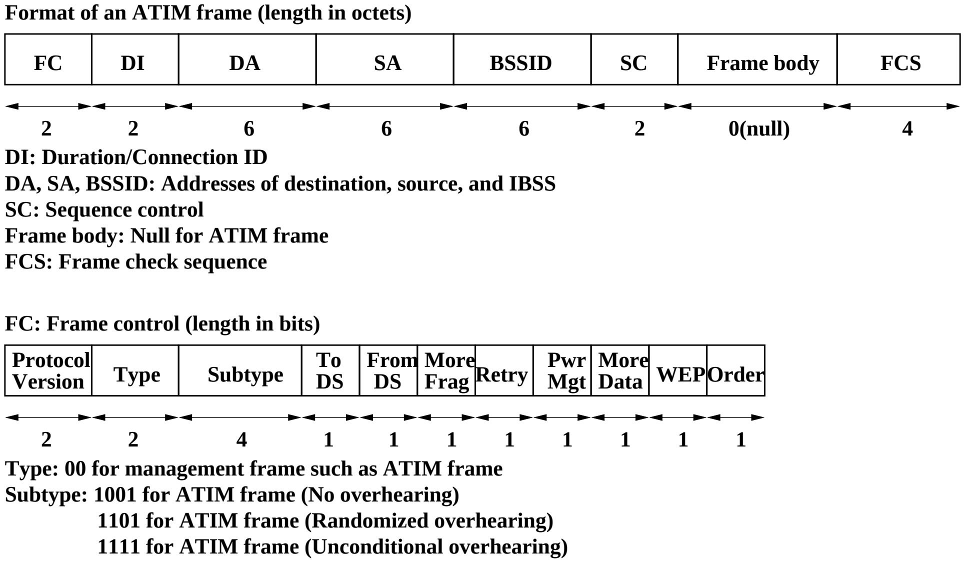
1. The node is the intended destination ($DA = MA$).
2. The node is not the destination but the sender wants unconditional overhearing ($DA \neq MA$ but subtype ID = 1111_2).
3. The node is not the destination, but the sender wants randomized overhearing, and the node randomly decides to overhear the packet ($DA \neq MA$, subtype ID = 1101_2 and decides to overhear).

A key design issue in the Rcast implementation is the mechanism of overhearing when the sender specifies randomized overhearing as in step 3 above. Basically, each node maintains a probability, P_R, determined using the factors listed below and probabilistically makes the overhearing decision based on P_R. In other words, if a randomly generated number is $> P_R$, then a node decides to overhear.

- *Sender ID*: The main objective of Rcast is to minimize redundant overhearing as much as possible. Since a node usually repeats the same route information in consecutive packets, a neighbor can easily identify the potential redundancy based on the sender ID. For instance, when a node receives an ATIM frame with subtype 1101_2, it determines to overhear it if the sender has not been heard or overheard

---

[3] Note that “PwrMgt” in FC indicates the power management mode, either AM or PS, in which the sender of the frame will stay after the current communication is successfully completed.



DI: Duration/Connection ID
DA, SA, BSSID: Addresses of destination, source, and IBSS
SC: Sequence control
Frame body: Null for ATIM frame
FCS: Frame check sequence

Type: 00 for management frame such as ATIM frame
Subtype: 1001 for ATIM frame (No overhearing)
1101 for ATIM frame (Randomized overhearing)
1111 for ATIM frame (Unconditional overhearing)

Fig. 5.4. Format of ATIM frame implementing the Rcast mechanism (IBSS: Independence Basic Service Set, DS: Distribution System, and WEP: Wired Equivalent Privacy)^3.

for a while. The former condition means that the traffic from the sender happens rarely or the packet is for a new traffic. The latter condition holds when the node skips too many packets from the sender.

- *Mobility*: When node mobility is high, link errors occur frequently and route information stored in the route cache becomes stale. Therefore, it is recommended to overhear more conservatively in this case. Each node is not knowledgeable about mobility of every other node, but it can estimate its own mobility based on the connectivity changes with its neighbors.
- *Remaining battery energy*: This is one of the most obvious criteria that helps extend the network lifetime: Less overhearing if remaining battery energy is low. However, it is necessary to consider other nodes' remaining battery energy in order to make a better balance.
- *Number of neighbors*: When a node has a large number of neighbors, it is possible that one of them offers a routing path to the node when it asks for it by sending a RREQ. Therefore, the overhearing decision is related inversely to the number of neighbors.

Overhearing decision can be made based on the above four criteria, but in this paper, we adopt a simple scheme using only the number of neighbors ($P_R = 1$ / number of neighbors) to show the potential benefit of Rcast. In other words, if a node has five neighbors in its radio transmission range, it overhears randomly with the probability P_R of 0.2.

5.3.3 Rcast with DSR

As described in Section 5.2.1, DSR employs three control packets, RREQ, RREP and RERR, in addition to data packets. RREQ is a broadcast and RREP, RERR and data are unicast packets. For each of these unicast packets, DSR uses the following overhearing mechanism.

- *Randomized overhearing for RREP packets*: A RREP includes the discovered route and is sent from the destination to the originator of the corresponding RREQ packet. For example, in Fig. 5.2(c), node D sends a RREP to node S. Intermediate nodes as well as node D will Rcast this message to allow randomized overhearing. Unconditional overhearing of RREP is not a good idea because DSR generates a large number of RREP packets as discussed in Section 5.2.1.

- *Randomized overhearing for data packets*: In DSR, every data packet includes the entire route from the source to the destination. Each intermediate node (e.g., nodes A, B, and C in Fig. 5.2(c)) as well as the source node (e.g., node S in Fig. 5.2(c)) will demand randomized overhearing for these packets so that neighboring nodes (e.g., nodes X and Y in Fig. 5.2(c)) can overhear them randomly.

- *Unconditional overhearing for RERR packets*: When a broken link is detected, an upstream node (e.g., node B in Fig. 5.2(c)) transmits a RERR to the source. Nodes will overhear this message unconditionally because the stale route information must be invalidated as soon as possible from nodes' route caches.

Note that a broadcast packet such as RREQ can also be Rcasted to allow randomized receiving as mentioned in the introduction. This is to avoid redundant rebroadcasts of the same packet in dense mobile networks. In this case, the overhearing decision must be made conservatively to make sure that the broadcast packet such as RREQ is propagated correctly until it reaches the final destination. Note also that the randomization approach described above can avoid the occurrence of preferential attachment, discussed in Section 5.2.1, and lead to a more balanced network with respect to packet forwarding responsibility and energy consumption.

5.4 Performance Evaluation

5.4.1 Simulation Testbed

The performance benefits of employing the Rcast scheme are evaluated using the ns-2 network simulator [4], which simulates node mobility, a realistic physical layer, radio network interfaces, and the DCF protocol. Our evaluation is based on the simulation of 100 mobile nodes located in an area of 1500 × 300 m^2. The radio transmission range is assumed to be 250 m and the *two-ray ground propagation channel* is assumed with a data rate of 2 Mbps. The data traffic simulated is *constant bit rate* (CBR) traffic. 20 CBR sources generate 0.2 to 2.0 256-byte data packets every second (R_{pkt}). *Random waypoint mobility model* [38] is used in our experiments with a maximum node speed of 20 m/s and a pause time (T_{pause}) of 0 to 1125 seconds. With this approach, a node travels towards a randomly selected destination in the network. After the node arrives at the destination, it pauses for the predetermined period of time and travels towards another randomly selected destination. Simulation time is 1125 seconds and each simulation scenario is repeated ten times to obtain steady-state performance metrics.

In our simulation, we use 250ms and 50ms for the size of beacon interval and ATIM window, respectively, as suggested in [87]. We assume that any data packet, which is successfully delivered during the data transmission period, has been successfully announced (and acknowledged) during the proceeding ATIM window. When the traffic is light, this assumption usually holds. When traffic becomes heavier, nodes fail to deliver ATIM frames and would not attempt to transmit packets during the data transmission period. Therefore, the actual performance would be better than the one reported in this chapter.

We compare three different schemes: 802.11, ODPM, and RCAST. 802.11 is unmodified IEEE 802.11 without PSM. As discussed in Section 5.2.2, ODPM [95] is one of the most competitive schemes developed for multihop networks, employing on-demand routing algorithms. For ODPM, a node remains in AM for 5 seconds if it receives a RREP. It remains in AM for 2 seconds if it receives a data packet or it is a source or a destination nodes. These values are suggested in the original paper [95]. RCAST uses no/unconditional/randomized overhearing depending on the packet type as explained in the previous Section. Table 5.1 compares protocol behaviors of the three schemes with their expected performance.

5.4.2 Performance Metrics

Performance metrics we have used in our experiments are energy consumption, packet delivery ratio (PDR), and packet delay. Energy consumption is measured at the radio layer during simulation based on the specification of IEEE 802.11-compliant WaveLAN-II [41] from Lucent. The power consumption varies from 0.045 W (9mA ×

Table 5.1. Protocol behavior of the three schemes.

Scheme	Behavior
802.11	**Behavior**: Does not incorporate PSM and nodes are always awake. Thus, packets are transmitted immediately whenever they are ready. **Expected Performance**: Best PDR and delay, but consumes maximum energy.
ODPM	**Behavior**: Nodes remain in the AM mode for a pre-determined period of time when they receive a RREP or a data packet or they are souce or destination. **Expected Performance**: Less packet delay than RCAST because some packets are transmitted immediately. Higher energy cost than RCAST because some nodes remain in the AM mode.
RCAST	**Behavior**: All nodes operate in the PS mode consistently and overhearing is controlled. Packets are deferred until the next beacon interval. **Expected Performance**: Less energy and better energy balance than ODPM.

5 Volts) in a low-power sleep state to 1.15 to 1.50 W (230 to 300mA × 5 Volts) in idle listening, receiving or transmitting states. In our experiment, we assume nodes consume 1.15W during AM and 0.045W during the low-power sleeping mode. The instantaneous power is multiplied by the time delay to obtain energy consumption. In order to examine the performance tradeoffs, a combined metric has been used in this paper: Energy consumption to successfully deliver a bit or *energy per bit* (EPB).

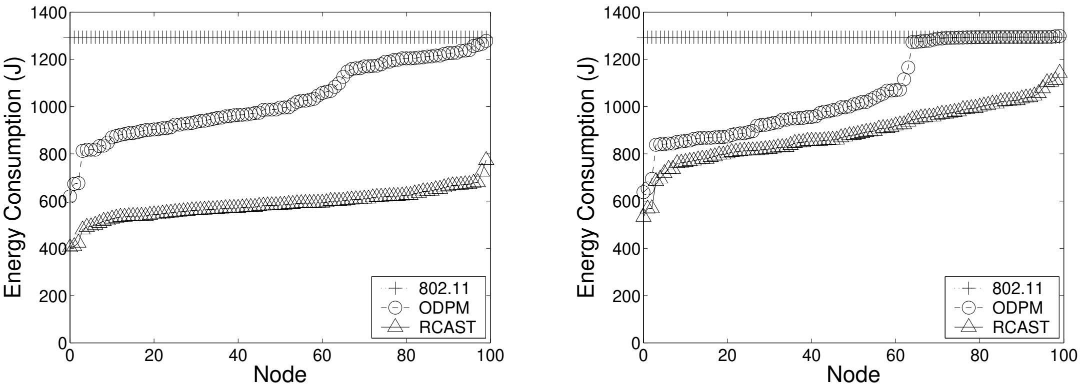
Energy balance is another important performance measure. We compare the variance of energy consumption of different nodes. As discussed in Section 5.2.1.3, we suspect the energy unbalance is mainly caused by non-uniformity in packet forwarding responsibility. In order to examine it, we define and compare the *role number* of a node as a measure of the extent to which the node lies on the paths between others. It can be considered as a measure of the influence, or utility of a specific node when forwarding packets in a network. The role number of a node is calculated by examining each node's route cache to find all intermediate nodes stored during all packet transmissions.

5.4.3 Simulation Results

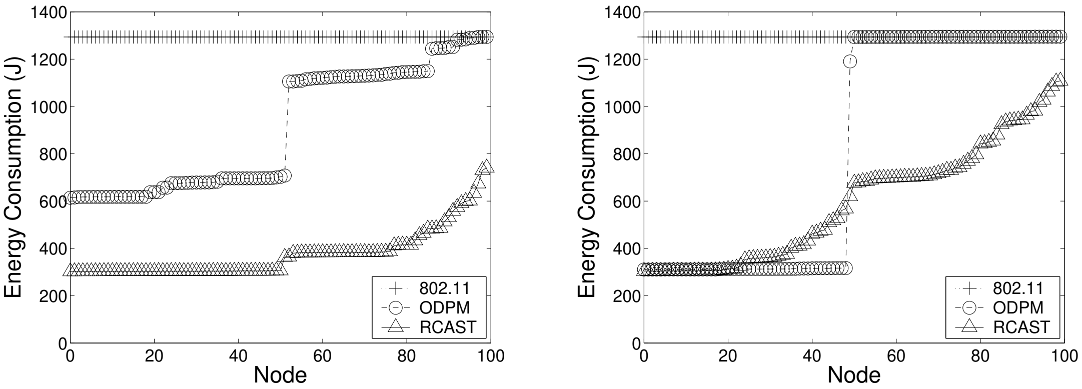
Fig. 5.5 shows the energy consumption of all 100 nodes drawn in an increasing order. Figs. 5.5(a) and 5.5(b) use 60 seconds of pause time, while Figs. 5.5(c) and 5(d) use 1125 seconds (static scenario). Figs. 5.5(a) and 5.5(c) simulate low-traffic condition (0.4 packets/second) and Figs. 5.5(b) and 5.5(d) simulate higher-traffic scenario (2.0 packets/second). In all the figures, 802.11 consumes the maximum energy and it is the same for all nodes since they are awaken during the entire period of simulation time (1.15W × 1125 seconds = 1293.75 Joules). RCAST performs much better than ODPM.

More importantly, RCAST outperforms ODPM with respect to energy balance, which becomes more significant in a static scenario as shown in Figs. 5(c) and (d). In ODPM, with a packet rate of 2.0 packets/second, the source and destination nodes continue to be awaken (in AM) during the entire 1125 seconds because the inter-packet interval (0.5 second) is smaller than the predefined timeout values (2.0 seconds). This is also true for all intermediate nodes between the sources and the destinations. Other nodes would not be bothered and wake up only during the ATIM windows consuming less energy (1.15W × 225 seconds + 0.045W × 900 seconds = 299.25 Joules) as shown in Fig. 5.5(d). When packet rate is 0.4 packets/second, the inter-packet interval (2.5 seconds) is longer than the timeout interval, and thus, the energy balance improves but still much worse than RCAST as in Fig. 5.5(c). It becomes clear in Fig. 5.6, which shows the energy balance in terms of variance of energy consumption between nodes. 802.11 shows no variance simply because all the nodes consume the same (maximum) amount of energy. With respect to ODPM, RCAST improves energy balance by 243% to 400%, and this is consistently true regardless of the traffic intensity and mobility. Thus, ODPM might be acceptable in mobile and low-traffic scenarios, but RCAST looks more promising in every possible scenario, especially under low mobility or high traffic scenario.

Fig. 5.7 shows the total energy consumption, PDR and EPB for the three different schemes as a traffic of packet injection rate (0.2 to 2.0 packets/second). Again, 802.11 consumes the largest amount of energy and RCAST performs better than ODPM by 28% to 75% as shown in Fig. 5.7(a). The performance gap increases between 37% to 131% under a static scenario as depicted in Fig. 5.7(d). Figs. 5.7(b) and 5.7(e) show that all three schemes deliver more than 90% of packets successfully under the traffic

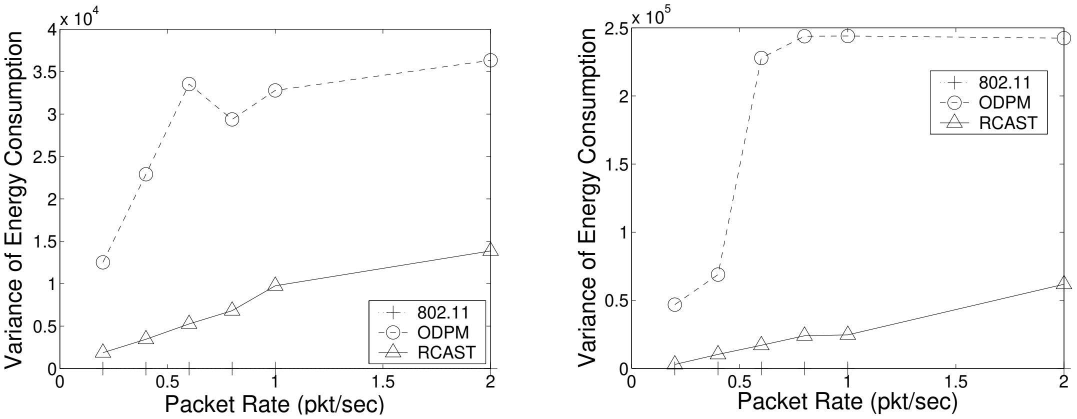


(a) $R_{pkt} = 0.4$, $T_{pause} = 60$ (b) $R_{pkt} = 2.0$, $T_{pause} = 60$



(c) $R_{pkt} = 0.4$, $T_{pause} = 1125$ (d) $R_{pkt} = 2.0$, $T_{pause} = 1125$

Fig. 5.5. Comparison of energy consumption of each node.

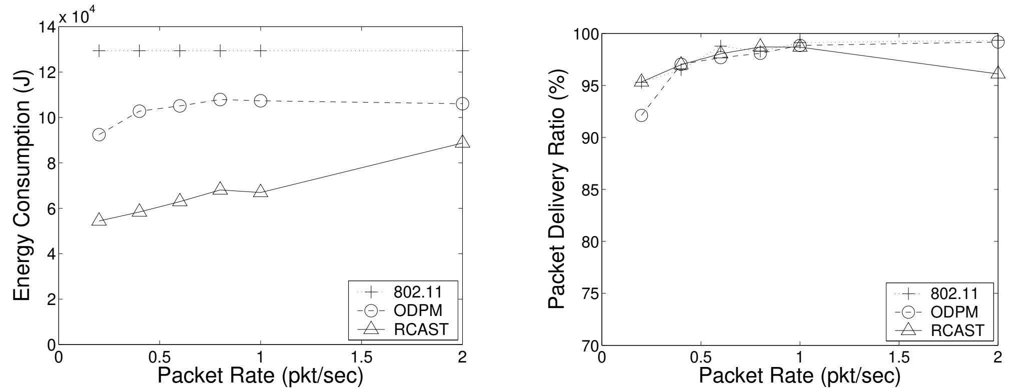


(a) $T_{pause} = 60$ (b) $T_{pause} = 1125$

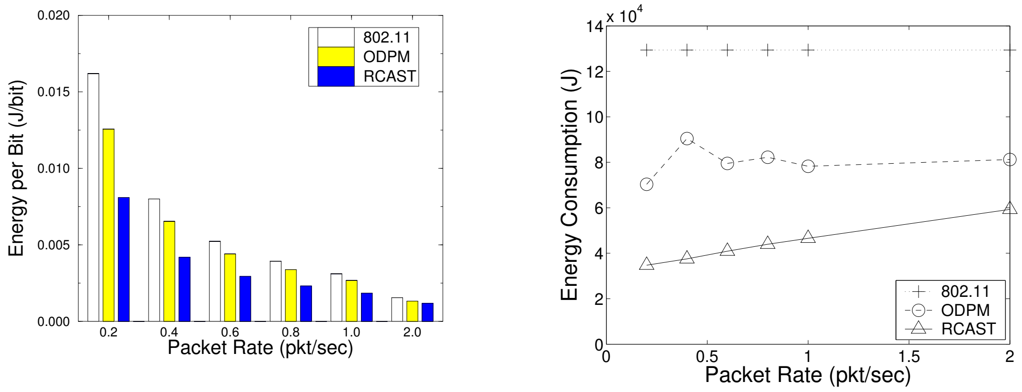
Fig. 5.6. Comparison of variance of energy consumption.

condition simulated. EPB, which is a combined metric of PDR and energy consumption, is drawn in Figs. 5.7(c) and 5.7(f). RCAST requires as much as 75% less energy than ODPM to successfully deliver a bit. 802.11 suffers even though it shows the best PDR because of its high energy cost.

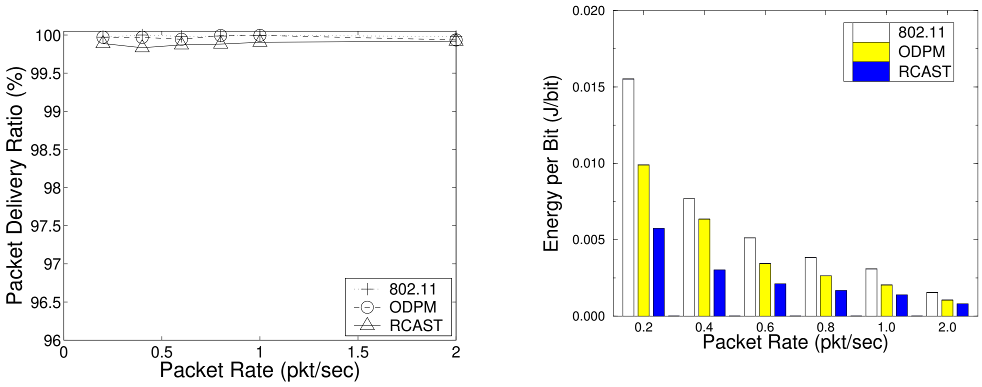
Fig. 5.8 shows the average packet delay and normalized routing overhead. As shown in Figs. 5.8(a) and 5.8(c), the packet delay is the smallest with 802.11 and ODPM. This is because all (802.11) or some (ODPM) data packets are transmitted immediately without waiting for the next beacon interval as discussed in Section 5.4.1. In RCAST, each packet must wait, on an average, half of a beacon interval (125 msec) for each hop, resulting in an extended delay compared to 802.11 and ODPM. On the other hand, routing efficiency is evaluated using the normalized routing overhead, measured in terms of the number of routing-related control packets per a successfully delivered



(a) $T_{pause} = 60$ (b) $T_{pause} = 60$



(c) $T_{pause} = 60$ (d) $T_{pause} = 1125$

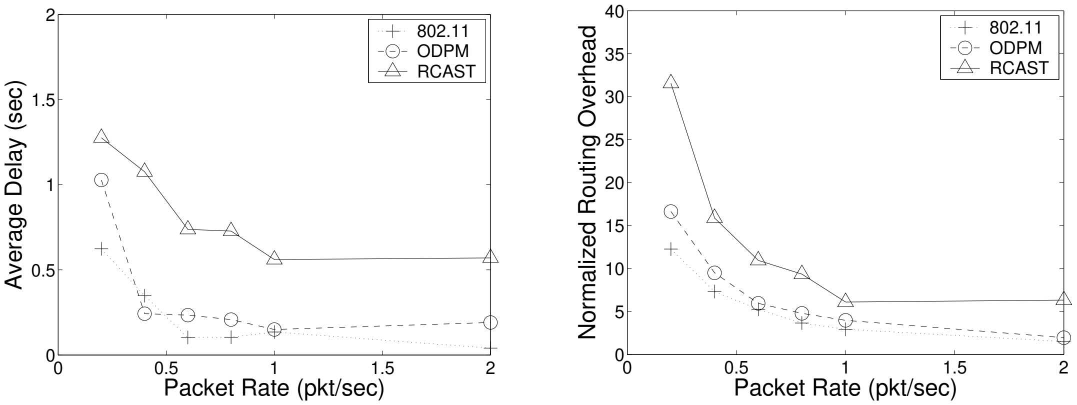


(e) $T_{pause} = 1125$ (f) $T_{pause} = 1125$

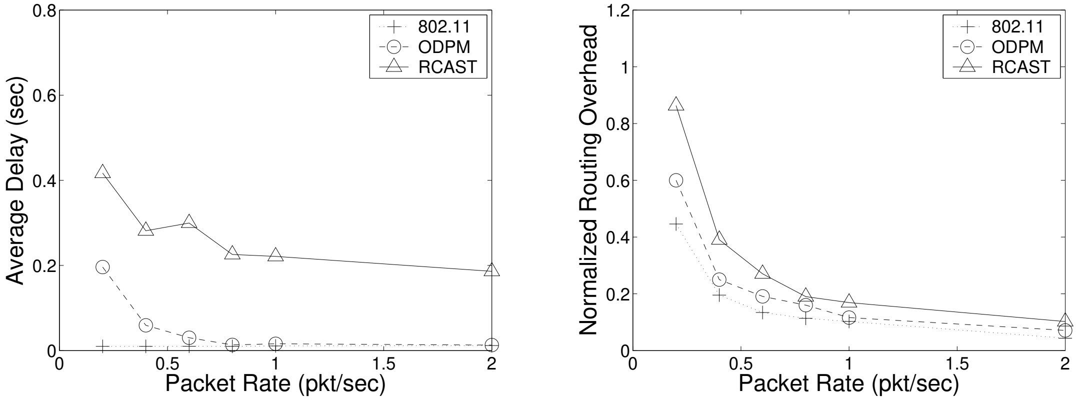
Fig. 5.7. Comparison of total energy consumption, packet delivery ratio, and energy per bit.

data packet. Comparing the static (Fig. 5.8(d)) and mobile (Fig. 5.8(b)) scenarios, we observe that the overheads are significantly different. Since, in a mobile scenario, there would be more link errors and more route discoveries, it is expected that the routing overhead becomes significantly large. In each scenario, it is observed that the overhead is the smallest with 802.11 and the other protocols behave similarly. In other words, RCAST performs at par with ODPM even with limited overhearing.

To investigate the reasons behind energy unbalance, we measured the distribution of role numbers, defined in Section 5.4.2. Fig. 5.9 depicts the scatter plot of role number and energy consumption when node mobility is high ($T_{pause} = 60$). It can be inferred that energy consumption is balanced if data points are located densely along the y-axis. Similarly, role number or packet forwarding responsibility is balanced if data points are located densely along the x-axis. It also shows relationship between energy consumption of a node and its role number. The results for 802.11 is trivial because energy consumption is almost the same as in Figs. 5.9(a) and 5.9(b). RCAST shows better energy balance than ODPM as discussed previously. With respect to role number distribution, RCAST achieves better balance as shown in Fig. 5.9. It is particularly true with higher data traffic. In Fig. 5.9(d), the maximum role number is about 500 in ODPM, while it is about 300 in RCAST. Since a node with a higher role number will forward more packets than others with a lower role number, it is not desirable in terms of device as well as network lifetime. However, the experiment shows that the role number is not directly related to energy consumption; it is not clear whether a better role number distribution results in a better energy balance. The related issues constitute some of our future work.

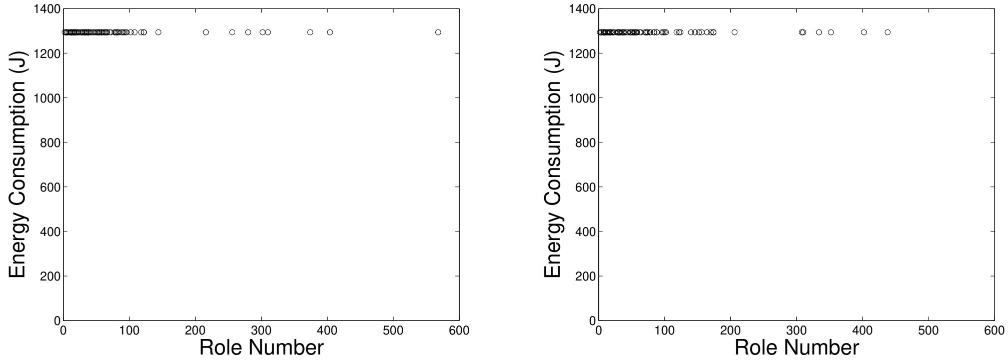


(a) $T_{pause} = 60$ (b) $T_{pause} = 60$

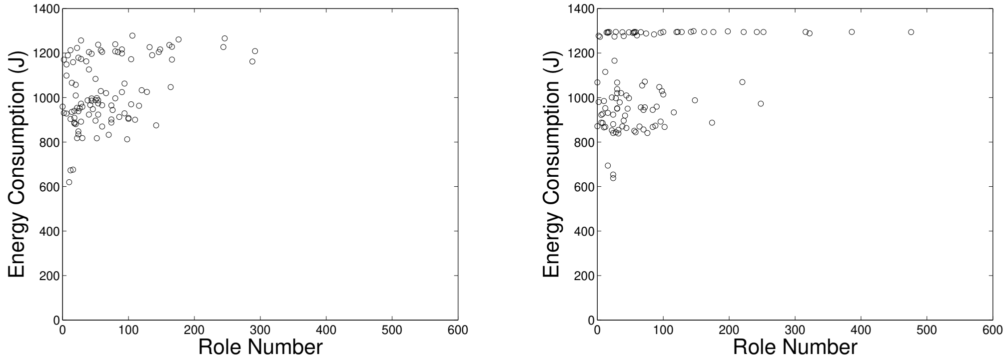


(c) $T_{pause} = 1125$ (d) $T_{pause} = 1125$

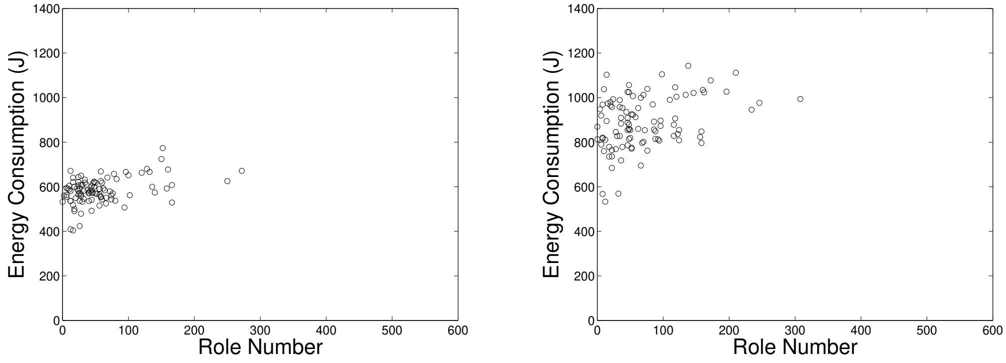
Fig. 5.8. General performance comparison in terms of average delay, and normalized routing overhead.



(a) 802.11, $R_{pkt} = 0.4$ (b) 802.11, $R_{pkt} = 2$



(c) ODPM, $R_{pkt} = 0.4$ (d) ODPM, $R_{pkt} = 2$



(e) RCAST, $R_{pkt} = 0.4$ (f) RCAST, $R_{pkt} = 2$

Fig. 5.9. Comparison of role number and energy consumption.

5.5 Conclusions

IEEE 802.11 is the most widely used wireless LAN standard specifying the physical and MAC layer protocols. While there has been active study on multihop networks with respect to many aspects including energy conservation, there is little effort about how to integrate the well-known IEEE 802.11 PSM with a multihop routing protocol such as DSR. This study addresses this important problem and suggests an efficient solution based on RandomCast (Rcast). The key idea behind the Rcast development is that unconditional overhearing, which is taken for granted without PSM, is not freely available with PSM. This is because packet advertisement is announced independently with respect to actual packet transmission, and thus, nodes which are not interested in receiving a particular packet can sleep during the actual transmission time to conserve energy. Therefore, nodes have an option whether to overhear or not a packet advertisement, and this decision must be made considering the tradeoffs between energy efficiency and routing efficiency. Routing efficiency comes into picture because overhearing is an important tool to gather route information in DSR.

We identify four factors that must be considered for the overhearing decision. These are sender ID, mobility, remaining battery energy, and number of neighbors. We implemented the Rcast scheme using only the last factor (number of neighbors), and compared it with four other schemes in terms of PDR, packet delay, energy consumption, and routing overhead through simulation. Our results indicate that Rcast significantly outperforms ODPM (as much as 28% to 131% less energy), which is the most competitive scheme developed for multihop networks employing on-demand routing algorithms, without significantly deteriorating the general network performance such as PDR. Rcast also improves energy balance by 243% to 400% in terms of variance in battery energy consumption. The performance results indicate that the proposed scheme is quite adaptive for energy-efficient communication in MANETs. In particular, applications without stringent timing constraints can benefit from the Rcast scheme in terms of power conservation.

Rcast opens many interesting directions of research to pursue. First, we plan to investigate the effect of other three factors (sender ID, mobility, and remaining battery energy) for making the overhearing decision. Since these factors increase the corresponding overheads, we also need to assess their tradeoffs. In particular, sender ID is the most compelling idea and can be implemented easily with a simple hashing function. Remaining battery energy will play an important factor if energy balance is critically important. We plan to explore the use of Rcast for broadcast messages and to incorporate the concept with other routing protocols.

Chapter 6

Conclusions

Design and analysis of wireless and mobile networks is the main focus of this thesis. Although it involves many challenging issues, it is a a fertile area for investigation since it has direct bearing on the performance of various realistic applications. The overall idea of this thesis is to develop suitable mechanisms to provide high and predictable performance in wireless and mobile networks, while considering the practical constraints of mobile networks.

6.1 Summary of Contributions

In this thesis, we investigated four major topics. These are (1) Study of a unified approach for QoS provisioning in cellular networks; (2) Study of a cache management mechanism for IMANETS for efficient information search and access; (3) Study of several push and pull-based cache invalidation strategies for IMANETS; and (4) Study of a new communication mechanism for energy efficient MANETS.

The major contributions of this thesis are outlined as follows:

- First, a unified framework consisting of a differential bandwidth reservation (DBR) algorithm and a QoS-aware admission control scheme is developed for cellular networks to guarantee a required level of QoS to on-going connections, while maintaining a competitive CBR for new connections. Comparison of the proposed scheme with two prior schemes shows that our approach is not only capable of providing better QoS guarantees, but also is flexible in terms of using varying number of cells in satisfying the high-level QoS requirements.

- Second, we address the issues involved in information search and access in an Internet-based mobile ad hoc network (IMANET), which is an emerging technique, that combines a wired network (e.g. Internet) and a mobile ad hoc network (MANET) for developing a ubiquitous communication infrastructure. An aggregate caching mechanism and a broadcast-based Simple Search (SS) algorithm are developed for improving the information accessibility and reducing average communication latency in IMANETS. As a part of the aggregate caching scheme, a new cache management policy is developed to reduce the cache miss ratio and improve the information accessibility. The proposed mechanism can significantly improve an IMANET performance in terms of throughput and average number of hops to access data items.
- Third, we develop and analyze several push and pull-based cache invalidation strategies for IMANETS. A global positioning system (GPS) based connectivity estimation (GPSCE) scheme is first proposed to assess the connectivity of an MT for supporting any cache invalidation mechanism. Then, we propose a pull-based approach, called aggregate cache based on demand (ACOD) scheme, to find the queried data items efficiently. In addition, we modify two push-based cache invalidation strategies, proposed for cellular networks, to work in IMANETS. These are a modified timestamp (MTS) scheme, and an MTS with updated invalidation report

(MTS+UIR) scheme. Our proposed strategy can provide high throughput, low query latency, and low communication overhead, and thus, is a viable approach for implementation in IMANETS.

- Finally, we examine the feasibility of integrating of the IEEE 802.11 power saving mode (PSM) with the popular Dynamic Source Routing (DSR) algorithm to develop an energy-efficient communication scheme. In this context, we develop a new communication mechanism, called RandomCast or Rcast, via which a sender can specify the desired level of overhearing in addition to the intended receiver. Thus, it is possible that only a random set of nodes overhear and collect route information for future use. By extensive simulation, we show that Rcast improves not only the energy efficiency, but also the energy balance among the nodes, without significantly affecting the routing efficiency.

6.2 Future Research Directions

We plan to extend our research in several directions and as summarized below.

6.2.1 QoS/Power Aware Wireless Networks

As a natural extension of this research on cache management/invalidation, we plan to consider the impact of the caching scheme on energy consumption and system fault-tolerance. In particular, we plan to investigate the power issue of different cache invalidation strategies discussed Chapter 4. For example, due to multi-hop IRs and message relays, it needs to schedule the sleep and wake up modes of the MTs judiciously during the IR interval without degrading performance. Also, we are interested in extending the caching scheme to handle multimedia Web traffic such as video streams to examine the impact of time/QoS constrained applications.

6.2.2 Design/Development of Medium Access Control (MAC) Protocol for Wireless LANs

Recent advances in high-speed wireless networks and mobile devices, wireless local area networks (WLANs) are becoming increasingly popular. In WLANs, since the communication medium is a scarce shared resource, efficient medium access control (MAC) protocols are critical in designing such systems. Thus, we plan to enhance the IEEE 802.11 MAC protocol to minimize collision and increase utilization, while allow fair access. Due to limited battery power, designing of energy efficient MAC protocol should be considered to reduce energy consumption. We would also like to research on the evaluation of TCP performance over different network topologies including pure multi-hop wireless networks and IMANETS, where both the centralized and distributed access methods co-existed.

6.2.3 Cross-layer Protocol Design for Energy Efficient MANETs

We plan to further investigate an interactive mechanism between an ad hoc routing protocol (e.g. DSR, Destination-Sequenced Distance Vector (DSDV), and Ad hoc On-Demand Distance Vector (AODV)) and the IEEE 802.11 PSM under realistic mobility patterns.

6.2.4 Security for Wireless Networks

Due to users' growing interests and decreasing cost in accessing the Internet wirelessly, wireless Internet networks are becoming a ubiquitous communication infrastructure, where users can access Internet services anywhere and anytime. However, unlike wired networks, where user must be authorized before accessing the media, a MANET is vulnerable to malicious MT's attack due to its lack of centralized control/authorization, time-varying network topology, and cooperative/distributed algorithm. Denial-of-service (DoS) attacks can cause a serious problem to current wireless networks. For example, malicious MTs may either modify/delete received control packets or generate an excessive amount of traffic. For defending against such attacks, we plan to investigate the design and development of security techniques in the context of ad hoc routing protocols, IEEE 802.11 MAC protocol, and various types of wireless applications.

We believe that due to the tremendous interest in wireless and mobile networks, design issues in terms of high-performance, energy, and security will be the central research focus in this area over the next decade.

References

[1] *The CSIM User Guides.* http://www.mesquite.com/userguidespage.htm.

[2] *http://www.cs.berkeley.edu/ awoo/smartdust.*

[3] *RF Monolithics, Inc., http://www.rfm.com.* ASH Transceiver TR 1000 Data Sheet.

[4] *VINT Project, The UCB/LBNL/VINT network simulator-ns (Version 2).* http://www.isi.edu/nsnam/ns.

[5] IEEE Std 802.11-1999, Local and Metropolitan Area Network, Specific Requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. In *http:// standards.ieee.org/getieee802/download/802-11-1999.pdf*, 1999.

[6] A. Agarwal and S. R. Das. Dead Reckoning in Mobile Ad Hoc Networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2003)*, 2003.

[7] S. Agarwal, S. V. Krishnamurthy, R. H. Katz, and S. K. Dao. Distributed Power Control in Ad-Hoc Wireless Networks. In *Proc. IEEE PIMRC*, pages 59–66, 2001.

[8] I.F. Akyukduz and J.S.M. Ho. On Location Management for Personal Communication Networks. *IEEE Communications Magazine*, pages 138–145, Sept 1996.

[9] A. Aljadhai and T. F. Znati. A Framework for Call Admission Control and QoS Support in Wireless Environments. In *Proc. IEEE INFOCOM*, pages 1019–1026, 1999.

[10] A. L. Barabasi. *Linked: The New Science of Networks.* Perseus Publishing, Cambridge, Massachusetts, 2002.

[11] D. Barbara and T. Imielinksi. Sleepers and Workaholics: Caching Strategies for Mobile Environments. In *Proc. ACM SIGMOD*, pages 1–12, 1994.

[12] G. Cao. A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments. In *Proc. ACM MOBICOM*, pages 200–209, 2000.

[13] G. Cao. Integrating Distributed Channel Allocation and Adaptive Handoff Management For QoS-Sensitive Cellular Networks. *ACM/Kluwer Wireless Networks (WINET)*, 9:131–142, 2003.

[14] G. Cao and M. Singhal. Distributed Fault-Tolerant Channel Allocation for Cellular Networks. *IEEE Journal on Selected Areas in Communications*, 18(7):1326–1337, July 2000.

[15] S. Capkun, M. Hamdi, and J. Hubaux. GPS-free Positioning in Mobile Ad-Hoc Networks. *Cluster Computing Journal*, 5(2):118–124, 2002.

[16] J.-H. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *Proc. IEEE INFOCOM*, pages 22–31, 2000.

[17] B. Chen, K. Jamieson, R. Morris, and H. Balakrishnan. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *Proc. ACM MOBICOM*, pages 85–96, 2001.

[18] M. Chiu and M.A. Bassiouni. Predictive Scheme for Handoff Prioritization in Cellular Networks Based on Mobile Positioning. *IEEE Journal on Selected Areas in Communications*, 18(3):510–522, March 2000.

[19] S. Choi and K. G. Shin. Comparison of Connection-Admission Control Schemes in the Presence of Hand-offs in Cellular Networks. In *Proc. ACM MOBICOM*, pages 264–275, 1998.

[20] S. Choi and K. G. Shin. Predictive and Adaptive Bandwidth Reservation for Hand–Offs in QoS–Sensitive Cellular Networks. In *Proc. ACM SIGCOMM*, pages 155–166, 1998.

[21] M. S. Corson, J. P. Macker, and G. H. Cirincione. Internet-Based Mobile Ad Hoc Networking. In *IEEE Internet Computing*, pages 63–70, July–August 1999.

[22] S. R. Das, C. E. Perkins, and E. M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *Proc. IEEE INFOCOM*, pages 3–12, 2000.

[23] S.K. Das, S.K. Sen, and R. Jayaram. A Dynamic Load balancing Strategy for Channel Assignment Using Selective Borrowing in Cellular Mobile Environments. *ACM/Baltzer Wireless Networks (WINET)*, 3(5):333–347, 1997.

[24] S. G. Dykes and K. A. Robbins. A Viability Analysis of Cooperative Proxy Caching. In *Proc. IEEE INFOCOM*, pages 1205–1214, 2001.

[25] B.M. Epstein and M. Schwartz. Predictive QoS-Based Admission Control for Multiclass Traffic in Celluar Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 18(3):523–534, March 2000.

[26] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. In *Proc. ACM SIGCOMM*, pages 254–265, 1998.

[27] J. Gomez, A. T. Campbell, N. Naghshineh, and C. Bisdikian. Conserving Transmission Power in Wireless Ad Hoc Networks. In *Proc. IEEE ICNP*, pages 24–34, 2001.

[28] R. Guerin. Queueing Blocking System with Two Arrival Streams and Guard Channels. *IEEE Trans. on Commun.*, 36:153–63, 1988.

[29] S. Hadjiefthymiades and L. Merakos. Using Proxy Cache Relocation to Accelerate Web Browsing in Wireless/Mobile Communications. In *Proc. World Wide Web (WWW 10)*, pages 26–35, May 2001.

[30] T. Hara. Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility. In *Proc. IEEE INFOCOM*, pages 1568–1576, 2001.

[31] T. Hara. Replica Allocation in Ad Hoc Networks with Period Data Update. In *Proc. 3rd International Conference on Mobile Data Management (MDM)*, pages 79–86, 2002.

[32] Q. Hu and D. Lee. Cache Algorithms based on Adaptive Invalidation Reports for Mobile Environments. *Cluster Computing*, pages 39–48, 1998.

[33] Q. Hu, D. L. Lee, and W. Lee. Performance Evaluation of a Wireless Hierarchical Data Dissemination System. In *proc. ACM MOBICOM*, pages 163–173, 1999.

[34] Y. Hu and D. B. Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks. In *Proc. ACM MOBICOM*, pages 231–242, 2000.

[35] L. Huang and T.-H. Lai. On the Scalability of IEEE 802.11 Ad Hoc Networks. In *Proc. ACM MOBIHOC*, pages 173–182, 2002.

[36] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang. A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks. In *Proc. ACM SIGCOMM*, pages 2–13, 1995.

[37] J. Jing, A. Elmagarmid, A. Helal, and R. Alons. Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environment. *Mobile Networks and Applications*, pages 115–127, 1997.

[38] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, pages 153–181, Kluwer 1996.

[39] E.-S. Jung and N. H. Vaidya. A Power Control MAC Protocol for Ad Hoc Networks. In *Proc. ACM MOBICOM*, pages 36–47, 2002.

[40] A. Kahol, S. Khurana, S. Gupta, and P. Srimani. An Efficient Cache Management Scheme for Mobile Environment. In *Proc. 20th International Conference on Distributied Computing System (ICDCS)*, pages 530–537, April 2000.

[41] A. Kamerman and L. Monteban. WaveLAN-II: A High Performance Wireless LAN for the Unlicensed Band. *Bell Labs Techical Journal*, pages 118–133, Summer 1997.

[42] G. Karumanchi, S. Muralidharan, and Ravi Prakash. Information Dissemination in Partitionable Mobile Ad Hoc Networks. In *Proc. IEEE Symposium on Reliable Distributed Systems (SRDS)*, pages 4–13, 1999.

[43] I. Katzela and M. Naghshineh. Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey. *IEEE Personal Communications*, 3(3):10–31, June 1996.

[44] V. Kawadia and P.R. Kumar. Power Control and Clustering in Ad Hoc Networks. In *Proc. IEEE INFOCOM*, pages 459–469, 2003.

[45] M. Kazar. Synchronization and Caching Issues in the Andrew File System. In *Proc. USENIX Conference*, pages 27–36, 1988.

[46] N.E. Kruijt, D. Sparreboom, F.C. Schoute, and R. Prasad. Location Management Strategies for Cellular Mobile Networks. *Electronics and Communication Engineering Journal*, 10(2):64–72, April 1998.

[47] K. Lee. Adaptive Network Support for Multimedia. In *Proc. ACM MOBICOM*, pages 62–74, 1995.

[48] S. Lee, W. Su, and M. Gerla. Ad Hoc Wireless Multicast with Mobility Prediction. In *Proc. 8th International Conference on Computer Communications and Networks (ICCCN)*, pages 4–9, Oct 1999.

[49] D.A. Levine, I. F. Akyildiz, and M. Naghshineh. A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concepts. *IEEE/ACM Transaction on Networking*, 5(1):1–12, Feb 1997.

[50] S. Lim, G. Cao, and C. R. Das. Performance Evaluation of A Differential Bandwidth Reservation Policy for Multimedia Wirelee Networks. Technical Report CSE–01–015, Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, May 2001.

[51] S. Lim, W. Lee, G. Cao, and C. R. Das. A Novel Caching Scheme for Internet based Mobile Ad Hoc Networks. In *Proc. 12th International Conference on Computer Communications and Networks (ICCCN)*, pages 38–43, 2003.

[52] S. Lim, W. Lee, G. Cao, and C. R. Das. Cache Invalidation Strategies for Internet-based Mobile Ad Hoc Networks (IMANETs). Technical Report CSE–04–011, Dept. of CSE, The PennState Univ., April 2004.

[53] S. Lim, S. Park, W. Lee, G. Cao, C. R. Das, and C. L. Giles. An Aggregate Caching for Internet based Ad Hoc Networks. Technical Report CSE–02–017, Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park,PA 16802, Oct 2002.

[54] S. Lu, K. Lee, and V. Bharghavan. Adaptive Quality of Service Support for Packet-Switched Wireless Cellular Networks. *Kluwer Multimedia Tools and Applications*, 17(2-3):157–179, July-August 2002.

[55] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *Proc. ACM MOBICOM*, pages 353–367, 2003.

[56] J. Luo, J. Hubaux, and P. Th. Eugster. PAN: Providing Reliable Storage in Mobile Ad Hoc Networks with Probabilistic Quorum Systems. In *Proc. ACM MobiHoc*, pages 1–12, 2003.

[57] M. K. Marina and S. R. Das. Performance of Route Caching Strategies in Dynamic Source Routing. In *Int'l Workshop on Wireless Networks and Mobile Computing (WNMC)*, pages 425–432, 2001.

[58] M. Naghshineh and M. Schwartz. Distributed Call Admission Control in Mobile/Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 14(4):711–717, May 1996.

[59] M. Papadopouli and H. Schulzrinne. Effects of Power Conservation, Wireless Converage and Cooperation on Data Dissemination among Mobile Devices. In *Proc. MobiHoc*, pages 117–127, 2001.

[60] V. D. Park and M.S. Corson. Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proc. IEEE INFOCOM*, pages 1405–1413, 1997.

[61] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. ACM SIGCOMM*, pages 234–244, 1994.

[62] C. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *2nd IEEE workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[63] G. P. Pollini and C. L. I. A Profile-Based Location Strategy and Its Performance. *IEEE Journal on Selected Areas in Communications*, 15(8):1415–1424, Oct 1997.

[64] S.S. Rappaport. Models for Call Hand-off Schemes in Cellular Communication Networks. In *3rd WINLAB Workshop: Third Generation Wireless Information Networks, Workshop Record*, Apr 1992.

[65] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 1996.

[66] J.H. Reed, T.S. Rappaport, and B.D. Woerner. Global Positioning Using Wireless Communications on Highways of the Future. *IEEE Communications Magazine*, 34(10):33–41, Oct 1996.

[67] Q. Ren and M. H. Dunham. Using Semantic Caching to Manage Location Dependent Data in Mobile Computing. In *Proc. ACM MOBICOM*, pages 210–221, 2000.

[68] M. Roussopoulos and M. Baker. CUP: Controlled Update Propagation in Peer-to-Peer Networks. In *Proc. USENIX Annual Technical Conference*, 2003.

[69] A.N. Rudrapatna, D.P. Agrawal, P. Agrawal, and C. Giardina. User Profile-Driven Resource Allocation for Cellular Networks. In *Proc. IEEE International Conference on Personal Wireless Communications(ICPWC)*, pages 178–183, 1997.

[70] F. Sailhan and V. Issarny. Cooperative Caching in Ad Hoc Networks. In *Proc. 4th International Conference on Mobile Data Management (MDM)*, pages 13–28, 2003.

[71] M. Sanchez, P. Manzoni, and Z. H. Hass. Determination of Critical Transmission Range in Ad-Hoc Networks. In *Proc. Multiaccess, Mobility, and Teletraffic for Wireless Communications (MMT)*, 1999.

[72] S. Sandberg, S. Kleiman D. Goldberg, D. Walsh, and B. Lyon. Design and Implementation of the Sun Network File System. In *Proc. USENIX Summer Conference*, pages 119–130, 1985.

[73] H. Schewetman. *CSIM User's Guide (Version 18)*. MCC Corp. http://www.mesquite.com, 1998.

[74] S.K. Sen, A. Bhattacharya, and S.K. Das. A Selective Location Update Strategy for PCS Users. *Wireless Networks*, pages 1–30, 1999.

[75] N. Shivakumar and J. Widom. User Profile Replication for Faster Location Lookup in Mobile Environments. In *Proc. ACM MOBICOM*, pages 161–169, 1995.

[76] S. Singh, M. Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *Proc. ACM MOBICOM*, pages 181–190, Oct 1998.

[77] W. Soh and H. S. Kim. Dynamic Guard Bandwidth Scheme for Wireless Broadband Networks. In *Proc. IEEE INFOCOM*, pages 572–581, 2001.

[78] W. Stallings. *IEEE 802.11 Wireless LAN Standard*. Prentice Hall, Inc., 2002. Chapter 14, Wireless Communications and Networks.

[79] W. Su and M. Gerla. Bandwidth Allocation Strategies for Wireless ATM Networks Using Predictive Reservation. In *Proc. IEEE GLOBECOM*, volume 4, pages 2245–2250, 1998.

[80] A. K. Talukdar, B. R. Badrinath, and A. Acharya. On Accommodating Mobile Hosts in An Integrated Services Packet Network. In *Proc. IEEE INFOCOM*, pages 1046–1053, 1996.

[81] S. Tekinay. A Measurement-Based Prioritization Scheme for Handovers in Mobile Cellular Networks. *IEEE Journal on Selected Areas in Communications*, 1:1343–1350, 1992.

[82] Y.-C. Tseng, Y. Chen, and J. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *Proc. IEEE MOBICOM*, pages 151–162, 1999.

[83] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-Saving Protocol for IEEE 802.11-Based Multi-Hop Ad Hoc Networks. In *Proc. IEEE INFOCOM*, pages 200–209, 2002.

[84] Bruce Tuch. Development of WaveLAN, an ISM band wireless LAN. *AT&T Techical Journal*, 72(4):27–33, 1993.

[85] M. Venkataraman and P. Gupta. *Stack Aware Architecture for Ad Hoc Networks.* Internet Draft (draft-venkataraman-stack-00.txt), 2004.

[86] Guang Wan and Eric Lin. Cost Reduction in Location Management Using Semi-realtime Movement Information. *Wireless Networks*, 5:245–256, 1999.

[87] H. Woesner, J. Ebert, M. Schlager, , and A. Wolisz. Power-Saving Mechanisms in Emerging Standards for Wireless LANs: The MAC Level Perspective. *IEEE Personal Communications*, 5(3):40–48, Jun 1998.

[88] K. Woo, C. Yu, H. Y. Youn, and B. Lee. Non-Blocking, Localized Routing Algorithm for Balanced Energy Consumption in Mobile Ad Hoc Network. In *Proc. MASCOTS*, pages 117–124, 2001.

[89] J. Wu, M. Gao, and I. Stojmenovic. On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Network. In *Proc. ICPP*, pages 346–356, 2001.

[90] K. Wu, P. Yu, and M. Chen. Energy-Efficient Caching for Wireless Mobile Computing. In *Proc. 20th International Conference on Data Engineering*, pages 336–345, 1996.

[91] K. Wu, P. S. Yu, and J. L. Wolf. Segment-Based Proxy Caching of Multimedia Streams. In *Proc. World Wide Web (WWW 10)*, pages 36–44, May 2001.

[92] T. Ye, H. A. Jacobsen, and R. Katz. Mobile Awareness in a Wide Area Wireless Network of Info-station. In *Proc. ACM MOBICOM*, pages 109–120, 1998.

[93] C. Yu, S.-M. Park, and H.-N. Kim. Every Node is Born Equal: Attacking Preferential Attachment in Peer-to-Peer Mobile Multihop Networks. In *Proc. IEEE PerCom Workshop on Mobile Peer-to-Peer Computing (MP2P'05)*, 2005.

[94] F. Yu and V. C. M. Leung. Mobility-Based Predictive Call Admission Control and Bandwidth Reservation in Wireless Cellular Networks. In *Proc. IEEE INFOCOM*, pages 518–526, 2001.

[95] R. Zheng and R. Kravets. On-Demand Power Management for Ad Hoc Networks. In *Proc. IEEE INFOCOM*, pages 481–491, 2003.

[96] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley, Cambridge, MA, 1949.

Vita

Sunho Lim is a Ph.D. candidate in the Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park. He received his B.S. degree (summa cum laude) from Dept. of Computer Science and M.S. degree from Dept. of Computer Engineering from Hankuk Aviation University (HAU), Korea, in 1996 and 1998, respectively. His research interests are in the areas of distributed systems, embedded mobile systems, network security, mobile data management, and wireless networks and mobile computing. He was a recipient of the CoManage Award for Excellence in Networking and the best graduate student teaching assistant award of the Dept. of Computer Science and Engineering at the Pennsylvania State University in 2000 and 2002, respectively. He is a member of IEEE. He served as a technical referee for numerous journals and conferences including IEEE Transactions on Wireless Communication, IEEE Communications Letter, Journal of Wireless Communications and Mobile Computing, Journal of Performance Evaluation, International Conference on Parallel and Distributed Processing Symposium, International Conference on Mobile Data Management, and International Conference on Computer Communications and Networks.