國立交通大學

資訊工程學系

碩士論文

整合行動電話網路及無線感測網路之事件驅動訊息系統

Event-Driven Messaging Services over Integrated Cellular and Wireless Sensor Networks: Prototyping Experiences and Analyses

研 究 生：劉衍谷

指導教授：曾煜棋 教授

中華民國九十三年六月

整合行動電話網路及無線感測網路之事件驅動訊息系統

Event-Driven Messaging Services over Integrated Cellular and Wireless Sensor Networks: Prototyping Experiences and Analyses

研究生： 劉衍谷　　Student : Yen-Ku Liu

指導教授： 曾煜棋　教授　　Advisor : Yu-Chee Tseng

國立交通大學
資訊工程學系
碩士論文

A Thesis
Submitted to Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science and Information Engineering

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

整合行動電話網路及無線感測網路之事件驅動訊息系統

學生：劉衍谷　　　　　　　　　　　　　　指導教授：曾煜棋老師

國立交通大學資訊工程學系 (研究所) 碩士班

摘要

在資訊網路中，即時訊息已經變成一種有效率的通訊工具。而在電話網路中，簡訊服務也日益普及。然而，現存的訊息服務被侷限在個別的網路中，且僅能被簡單的事件所驅動 (如：按下”送出”鍵等) 。我們認為根據環境的資訊來驅動即時訊息是合理的發展方向。有鑑於此，本論文提出一套事件驅動訊息系統，嘗試結合 GSM 行動通訊網路，並以藍芽技術作為無線位置感測系統。

結合上述技術，我們完成一套具有多項優點的訪客系統.結合藍芽感測網路。此系統的訊息可以被一些事件 (如使用者位置變化) 所驅動。再者，此訊息系統跨越電話網路和資訊網路的藩籬，使訊息可在兩種網路間流通。最後，在設計此系統時，我們採用模組化的設計，以維持系統擴充性。

此論文將會詳述訪客系統系統架構與實作細節。此外，將就感測能力以及藍芽封包碰撞對系統效能的影響作出探討。最後則就未來的發展作出規劃。

關鍵字: 藍芽，GSM，即時訊息，簡訊服務，感測網路，封包碰撞。

Event-Driven Messaging Services over Integrated Cellular and Wireless Sensor Networks: Prototyping Experiences of a Visitor System

Student: Yen-Ku Liu Advisors: Prof. Yu-Chee Tseng and Ting-Yu Lin

Institute of Computer Science and Information Engineering
National Chiao-Tung University

ABSTRACT

As one of the killer applications, instant messaging has become a simple yet efficient tool for peer-to-peer communications in data networks. In telephone networks, short message services are gaining more popularity as well. However, this kind of services typically operate in their own respective networks and is triggered by fairly simple events (such as pushing a “send” button or pre-scheduling the transmission at later time). A promising direction is to trigger instant messages by environmental information from the physical world. In light of this, this thesis proposes to establish an event-driven messaging service over a cellular-and-sensor-integrated network. We have prototyped a system which adopts GSM as the cellular network, and Bluetooth technology as the sensor network. The latter is to realize a Bluetooth surveillance network with location-sensing capabilities to be deployed within an office building area. While using other technologies is possible, GSM and Bluetooth are two dominating technologies in telephone and data networks. So the proposed technology is immediately feasible, given the fact that many handsets are already Bluetooth-enabled.

Through this combination, we demonstrate a *Visitor System (VS)* that offers several attractive features/services for visitors arriving at an office. First, messaging services in VS are driven by pre-configured events which can be collected from the Bluetooth surveillance network. Simple events might be a person entering/leaving a space, while complicated events might be a compound logic statement involving multiple users in VS. Second, we believe that the proposed system justifies the potential of cross-network applications and services. Third, the proposed architecture takes a modular approach by dividing the system into several subsystems according to their functionality. Logically dispatching jobs is the key to future extensions and further value-added services. A working prototype of this system has been deployed in a university building.

In this thesis, the system architecture and implementation details of our VS are reported. Furthermore, we investigate the performance issues, including sensing capability and collision analysis due to co-channel interference in a multi-Bluetooth piconets environment. Finally, possible future developments are also addressed.

Keywords: Bluetooth, context awareness, GSM, instant message, mobile computing, Short Message Service (SMS), sensor network, wireless communication, co-channel interference, collision analysis.

Acknowledgments

My advisor, prof. Yu-Chee Tseng, is the first one I would like to express my gratitude to. With the wonderful research conditions he provided and his attentive instructions, I came to discover the pleasure of research. I am also grateful to my co-advisor, Ting-Yu Lin. Without her help and suggestions, I would not be able to have this thesis done. Finally, I would like to thank all HSCC members for their generous advices. Discussing with them benefited me in many ways.

YenKu at CSIE, NCTU.

Contents

摘要 i

Abstract ii

Acknowledgments iv

Contents v

List of Figures vii

List of Tables viii

1 Introduction 1

2 Preliminaries 3

2.1 Instant Messaging Services 3

2.2 Wireless Sensor Networks 4

3 Event-Driven Messaging Services: a Visitor System 6

3.1 System Architecture 7

3.2 Events and Actions 11

3.3 Service Operations and Message Flows 12

3.4 Implementation Details and User Interfaces 16

4 Performance Issues 19

4.1 Sensing and Detecting Capability 19

4.2 Packet Collision Analysis 21

4.2.1 Problem Statement 21

4.2.2 Collision Analysis Without Considering Guard Time Effect . . 24

4.2.3 Enhanced Collision Analysis Considering Guard Time Effect . 27
4.2.4 Simulation and Experimental Results 31

5 Conclusions and Future Directions 35

Bibliography 37

Curriculum Vita 39

List of Figures

3.1 Visitor System architecture. 7

3.2 Message flows for a meeting-reminding instant message service. 13

3.3 Message flows for a relative-time instant message service. 15

3.4 Interfaces for laptop/palmtop terminals in VS. 17

3.5 The interactive WAP web page interfaces for handsets in VS. (a) instant message, (b) single event configuration, and (c) hybrid event configuration. 18

4.1 Average detection latency Vs. number of sensors with $T = 60$ sec. and $t = 10$ sec. (a) Regular deployment, and (b) random deployment 21

4.2 Average detection latency Vs. number of sensors with $T = 180$ sec. and $t = 10$ sec. (a) Regular deployment, and (b) random deployment 22

4.3 Frequency-hopping guard time in Bluetooth. 23

4.4 Classification of slot delimiters. 25

4.5 Analysis of success probabilities for (a) 3-slot and (b) 5-slot packets. . 26

4.6 Critical sections and collision analysis by considering guard time effect. 29

4.7 Collision analysis examples: B_j falling in (a) CS1, (b) CS2, and (c) CS3. 30

4.8 Packet error probabilities under (a) 100% traffic load and (b) 70% traffic load. 31

4.9 (a) Aggregate network throughput under 70% traffic load, and (b) per-piconet throughput under 70% traffic load. 32

4.10 (a) Aggregate network throughput vs. traffic load, and (b) per-piconet throughput vs. traffic load. 33

4.11 Guard time effect on: (a) packet error probability and (b) network throughput. 34

List of Tables

3.1 Service types and corresponding receiver IDs in VS. 12

Chapter 1

Introduction

Peer-to-peer messaging has become a common way of communications in people's daily lives. In mobile phone systems, such as GSM and NTT DoCoMo i-mode, short message services are gaining more revenues year by year according to the ISP statistics [13, 14]. For IP-based data networks, instant messaging software, such as ICQ and MSN [8, 10], is also popular for nowadays Internet users. In addition, an interesting project called Hubbub [15], which aims at providing mobile users with impromptu information exchange services, is recently reported. However, this kind of services typically operates in their own respective networks and is triggered by fairly simple events (such as pushing a "send" button or pre-scheduling the transmission at later time).

On the other hand, with the rapid development of wireless data networks, the potential service range of instant messages can be further extended, if cross-network solutions can be provided. In this work, we adopt Bluetooth [12], which is one of the dominating short-range wireless radio systems, as our platform. Bluetooth is characterized by small size and low cost, and has drawn much attention from both academia and business worlds [3, 7, 9, 17, 21]. It is also available immediately, given the fact that many laptops, PDAs, and handsets are already Bluetooth-enabled. Main applications of Bluetooths nowadays include cable replacement and simple voice/data exchange.

Motivated by the above observations, this thesis proposes an *event-driven messaging service* over an integrated GSM-IP-Bluetooth network. We use Bluetooth as a sensor network for surveillance and location-tracking purposes in an indoor environment. GSM is used to support instant messaging services. These networks are integrated under IP networks. Targeting at providing visitor-friendly services in an office environment, our system is referred to as the *Visitor System (VS)*. VS offers several attractive features/services. First, messaging services in VS are driven by pre-configured events which can be collected from the Bluetooth sensor network. Simple events might be a person entering/leaving a space, while complicated events might be a compound logic statement involving multiple users and multiple locations in VS. Through logic operators such as "AND", "OR", and "INVERSE", we can provide new types of event-driven instant messaging services with novel applications. Second, we believe that the proposed system justifies the potential of cross-network applications and services. In addition, integrated services across different networks release people from a used-to-be-confined service area. Migration between networks is transparent to users in VS. Third, the proposed architecture takes a modular approach by dividing the system into several subsystems according to their functionality. Logically dispatching jobs is the key to future extensions and further value-added services. This feature provides application programmers with the flexibility to customize their own visitor systems.

Since there may exist multiple Bluetooth piconets in a physical environment, we also analyze the packet collision effect in a multi-piconet environment. In several earlier works [5, 6, 18], this problem is studied, but the results are still very limited in that packets are usually assumed to be uniform in lengths and in that time slots of each piconet are assumed to be fully occupied by packets. These assumptions have been successfully removed in the analytical results proposed in [19]. In this thesis, we further improve the analytical results in [19] by taking into account the frequency-hopping guard time effect in Bluetooth baseband. The result would offer a way to better estimate the network performance in a multi-piconet environment.

The rest of this thesis is organized as follows. In Chapter 2, we review several existing messaging services and sensing systems. Chapter 3 reports our VS in more details. Chapter 4 addresses the performance aspects. Finally, Chapter 5 concludes this work and discusses possible extensions of VS.

Chapter 2

Preliminaries

This chapter reviews several messaging systems and location-aware sensor networks.

2.1 Instant Messaging Services

We present messaging services in telecommunication systems first, followed by those in IP-based networks.

- Short Message Service (SMS) in GSM [13]: As part of the GSM Phase 1 standard, SMS is a store-and-forward service providing text-based messages originated from and sent to mobile phones. Currently, the SMS center is in its second generation, which is capable of handling more messages in a more efficient and reliable way. Furthermore, national SMS inter-networking and international SMS roaming allow mobile users to communicate short messages freely between different operators. As GSM is evolving to encompass high-speed packet data services (e.g., GPRS), SMS is expected to support longer messages. In addition, potential multimedia applications are under development. For example, the emerging Multimedia Messaging Service (MMS) supports combinations of text, audio, picture, and video.

- NTT DoCoMo i-mode/FOMA [14]: The i-mode platform developed by Japan NTT DoCoMo provides a convenient interface for content providers over mobile phone telecommunication systems. This mobile service allows users to access more than 75,000 Internet sites, as well as specialized services such as e-mail, online shopping and banking, ticket reservations, and restaurant advice. Charges in i-mode are based on the volume of data transmitted, rather than the amount of time remaining connected. With the success of i-mode, in Oct. 2001, NTT DoCoMo further launched the FOMA services based on Wideband-CDMA 3G networks. FOMA supports high-speed data transmissions from 64 kbps to 384 kbps, thus enabling more attractive high-speed services.

- ICQ [10]: Originated from the pronunciation of "I seek you", ICQ enables Internet users with capabilities of sending instant messages, online chatting, as well as files exchanges. Besides peer-to-peer instant messaging, ICQ also allows users to conduct group conferences or participate in online games. Furthermore, another ICQ-based software, ICQphone, allows users to enjoy voice communications over PCs/IP phones via IP telephony technologies.

- MSN Messenger [8]: Similar to ICQ, the MSN Messenger (or .NET Messenger) delivered by Microsoft also creates an environment for users to pleasantly enjoy instant message, chat, file transfer, and video conference. An MSN Messenger client program will connect to an MSN Messenger server over the Internet. The client then exchanges data with other clients through the server, which keeps track of users' current points of Internet attachment. The instant messages are forwarded by the server, and processed by the destined client itself. MSN Messenger was not bundled with Windows operating systems until Windows XP, in which the messaging service is incorporated in a program known as Windows Messenger.

2.2 Wireless Sensor Networks

The *Wireless Sensor Network* (WSN) has many applications in disaster rescue and environment monitoring applications. Reference [1] discusses important design issues of WSN. Advances in micro-sensor and communication technologies have made it possible to manufacture small and cost-effective WSNs [16, 23]. Several attractive applications of WSNs have been reported [2, 20]. Below, we review several important systems.

- Active Badge [27]: The Active Badge system provides a means for locating individuals within a building by determining the locations of Active Badges worn by them. An Active Badge is a small device that can transmit a unique infrared signal periodically. Important places within a building such as offices and corridors can be equipped with networked sensors (i.e., badge readers) to detect these badges. The locations of badges (and hence their wearers) are then sent from the sensors to a server on the backbone.

- RFID [11]: Radio frequency identification (RFID) technology has been used for years in transportation applications (rail car tracking, toll collection, and vehicular access control) and inventory management and monitoring. A typical RFID system consists of some tags, readers, and processing servers on the backbone network. A reader can transmit radio waves to trigger antennas of tags. A tag then can convert the signal into DC voltage to charge itself and in turn emit radio signals with identification information for readers to interpret. As a result, the sizes of tags can remain very small due to this batteryless design. Readers then can pass the collected data on to the backbone servers to exploit further applications.

- Berkeley Smart Dust [28]: The Smart Dust mote developed by U.C. Berkely contains a microcontroller. Periodically the microcontroller gets readings from sensors, which can measure different physical or chemical stimuli such as temperature, ambient light, vibration, acceleration, and air pressure, and processes the data. It also occasionally turns on its optical receiver to see if anyone is trying to communicate with it. In response to a message or upon its own initiative, the microcontroller can use its corner cube retroreflector or laser to communication with other motes or base stations. A smart dust is designed to be only a few cubic-millimeters in volume, with sensing, communicating, and self-powering capabilities. Given the limited available storage, one major challenge is to incorporate power-saving designs in all functional parts.

- MIT Cricket [24]: The Cricket indoor location system is developed by MIT. The system is based on TinyOS and supports continuous object tracking. Following the TDoA (time difference of arrival) model, the Cricket Compass uses a combination of RF and ultrasound technologies to determine the position and orientation of a device. A program called listener is used to analyze beacons from objects. A maximum-likelihood estimator is used to correlate a sequence of location information.

Chapter 3

Event-Driven Messaging Services: a Visitor System

In Chapter 2, we have reviewed several instant messaging and sensor networking technologies. We observe that combining these two technologies would be very powerful to provide many novel applications. The sensor networks can detect and report environment-related information and events. Then through instant messaging systems, these events can be sent to the outside world for processing immediately. These events may trigger human or application programs to take actions, which may be further fed back into the sensor networks.

We have prototyped a *Visitor System (VS)* by combining a Bluetooth-based sensor network and the GSM SMS instant messaging service. Our VS targets at providing novel and friendly services for visitors coming to a building, such as a company headquarters. The devices at visitors' hands can be a Bluetooth-enabled laptop/palmtop or a Bluetooth-enabled GSM WAP-compatible handset. We deploy Bluetooths in important locations within a building, which are connected to a backbone wireline network, to form a sensor network. We use the Bluetooth-based sensor network to identify the locations of visitors/employees in the building, as well as to provide data services. Messaging services are delivered to visitors/employees via either Bluetooth or GSM SMS.

GSM SMS has become a pervasive telecommunication data service, and has triggered many interesting applications. On the other hand, Bluetooth is characterized by low price, low power consumption, and short-range communication, making it a potential choice for supporting positioning in an indoor, office environment. Nowa-

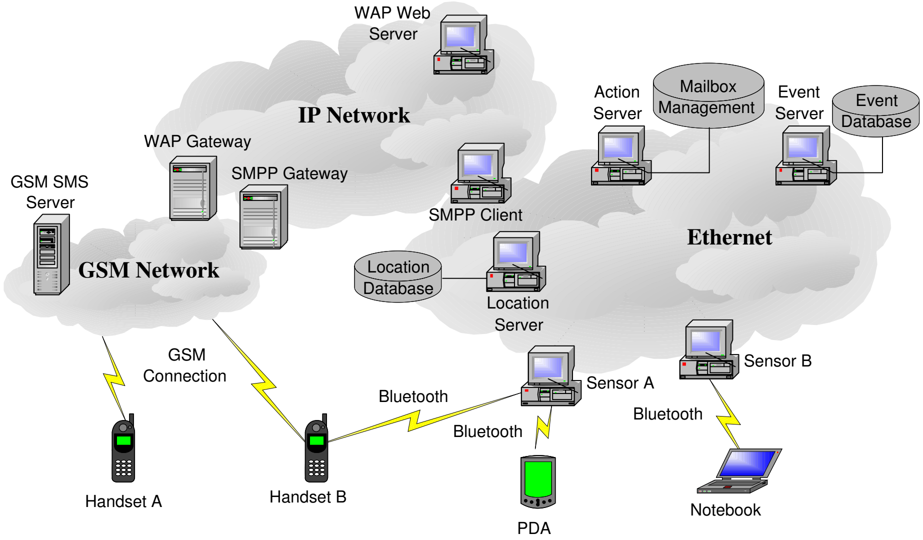


Figure 3.1: Visitor System architecture.

days, many mobile devices are already equipped with Bluetooths. As a result, we adopt Bluetooth and GSM as our prototyping platform. Through the implementation, we have developed several interesting location-based "event-driven" services.

3.1 System Architecture

The architecture of our VS is illustrated in Fig. 3.1, which includes user equipments, sensors, backbone networks, and several servers. Below, we detail each component separately.

In our VS, user equipment could be laptops, palmtops, or GSM handsets with WAP capability. Since we adopt Bluetooth as the sensing technology, all participant mobile terminals should be Bluetooth-enabled. One exception is a single GSM handset that is only WAP-enabled, in which case the user can only receive our instant messageing services, but can not be detected while traveling in our VS coverage area. Every mobile terminal registered in VS is assigned a unique logical ID. We classify user equipment into two categories: *program-loadable*, such as laptops and PDAs, and *program-unloadable*, such as mobile phones. For program-loadable terminals, a client program will be installed such that users can configure their own events and query databases. For program-unloadable terminals, a WAP interface is provided in VS so that users can still enjoy most functionalities provided to program-loadable devices.

A Bluetooth-based sensor network is deployed at important venues in an office building. These sensors are responsible for location sensing. At every pre-defined 30 seconds, each sensor will perform the "inquiry" procedure to discover newly arriving devices. The inquiry interval is in fact adjustable (we will further discuss this parameter in Chapter 4). Once a new device is discovered, the sensor will obtain its Bluetooth MAC address and logical ID, and notify the Location Server to establish a MAC-ID-location mapping in its database. Sensors also keep MAC-ID mapping at their local databases. Users' subsequent communications are done via logical IDs. Note that in addition to user location tracking, sensors are also responsible for relaying information between users and servers, which includes event configuration with the Event Server and instant message exchange with the Action Server.

The backbone network of VS is an integrated IP data network over the Bluetooth sensor network and GSM cellular network. The information exchange from network to network is transparent to end-users. In this way, communications are no longer confined to the limited office area. In particular, with the convenient and robust SMS service supported by GSM, we successfully extend the service area of VS. Two gateways, WAP and SMPP, are used to bridge these networks, which will be elaborated later on.

Our VS consists of a number of servers to support event-driven messaging services and a variety of functions. A modular system architecture is adopted in our design, so as to facilitate future extensions and improve flexibilities for third-party application developments. Note that those logically separated components can actually be executed on the same machine through multiple process threads. Below we describe servers in VS in more details.

- **Location Server**

  The Location Server maintains the user-location mapping in a database. Whenever a sensor detects a newly joining device, it will register the device ID with the Location Server. There is a two-way handshaking between the Location Server and sensors. Only after the registration is completed can the corresponding mapping be committed to its local database. Once a device is detected absent by a sensor in a pre-specified interval, the sensor will notify the Location Server to de-register that device. Other servers, such as Event Server or Action Server, will query the Location Server to obtain such user-location mapping information.

- **Event Server**

  In VS, actions are triggered by events. An event could be an instant message request, time event, location event, or their combinations. An instant message request is simply a message to be delivered to a terminal and it will be processed immediately. In VS, there are two basic events: *time event* and *location event* (detailed definitions to be presented in the next subsection). A user can configure a basic event or a compound event with the Event Server. The Event Server will parse users' requests into basic events and store them into its event database. Time events will be stored locally, while location events will be delivered to the Location Server. Time events are triggered by a local clock at the Event Server. Location events are stored at the Location Server and triggered by users' movements. Once a location event becomes true, a signal will be sent from the Location Server to the Event Server. The Event Server will then check its event database and, on a user event becoming true, trigger the Action Server to take proper actions. Once the action request is sent to the Action Server, the corresponding event record will be removed from the event database.

- **Action Server**

  The responsibility of the Action Server is to really deliver the services. Basically, the Action Server is triggered by commands from the Event Server. Upon receiving an action instruction, the Action Server will first issue a query to the Location Server to retrieve the receiver's location information. According to the receiver's profile, if it is recognized as a GSM handset, the service will be forwarded onto the GSM network via several assistant servers and delivered to the destined mobile phone. On the other hand, for services destined for a laptop/palmtop currently located in the VS coverage area, the Action Server acts as a gateway to relay the services to the receiver.

  To increase reliability, we associate a mailbox with each device ID. According to the classification of *mailbox-based schemes* in [4], our mailbox is stationary with "push" message delivery. However, we do not synchronize between mobile terminal migration and mailbox forwarding. Instead, we adopt a retransmission strategy. Before a message is successfully delivered, it is buffered locally at the Action Server. Each buffered message will be retransmitted at predefined time intervals. To avoid wasting too much resource on retransmissions, the retransmission interval is doubled each time a failure is experienced. We set up a lifetime of 2 hours for each message. Once the lifetime expires, the corresponding message will be deleted from the buffer.

- **SMPP Client**

  The Short Message Peer to Peer (SMPP) protocol is an open industry standard messaging protocol designed to simplify integration of data applications with wireless mobile networks such as GSM, TDMA, ,and CDMA. The current system that we are using is SMPP v3.3 (the newest version is v5.0, dated February 2003). In order to provide messaging services to GSM handsets, we develop a SMPP v3.3 client program. To transmit, we need to specify the SMPP Gateway Host Name, Port Number, User Name, Password and Telephone Number in the URL text. The SMPP Client is responsible for creating a SMPP connection to SMPP Gateway. The message content is attached at the end of the URL text, which will be formatted by the SMS protocol and forwarded to the GSM SMS Server for delivery.

- **WAP Web Server**

  WAP provides an interface for mobile handsets to access the Internet. GSM handset users can configure their requests/services through our WAP server. As a result, we set up a WAP web page to accept users' configuration request. This web page is developed using Wireless Markup Language (WML) and WML script. Any WAP-compatible GSM handsets can connect to the WAP Web Server and enjoy available VS services. On receipt of a request, the WAP server will contact the Event Server, which is responsible for processing the request.

- **Other Assistant Servers**

  In the prototyped system, we utilized several existing interfaces available in GSM telecommunication system. Those assistant servers include SMPP Gateway, WAP Gateway, and GSM SMS Server.

3.2 Events and Actions

One of the main features that distinguishes our VS from other messaging systems is its event-driven and location-aware service capability. The concept of events is useful when determining whether to take an action or not. Furthermore, a combination of multiple basic events may support more complicated novel applications. For example, a general manager may want to be notified that a meeting is ready only when the scheduled time is up and all other meeting members have shown up in the meeting room. A staff may want to be notified if he/she is not in the meeting room within 3 minutes before the meeting time.

In light of this, services in VS are accomplished by event-driven actions, which can be formulated by a number of rules, each with the following format:

$$< EvntService > \quad = \quad On \ < EvntVal > \ Do \ < Action >$$

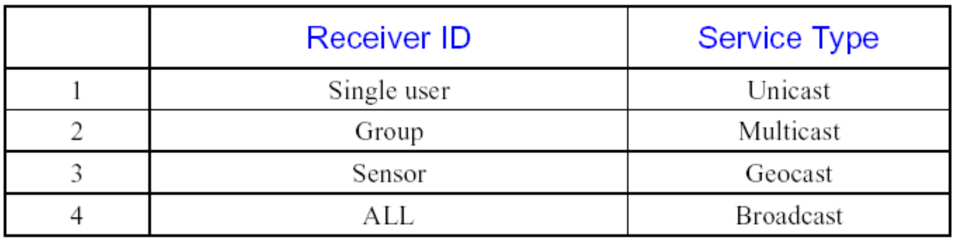
In VS, there are two types of basic events:

- **L**ocation Event: This simply includes someone entering or leaving an area.
- **T**ime Event: Four types of temporal concept can be specified: absolute time, relative time, time interval, and periodical time. For example, we can specify "10am on 7/9/2003", "20 min. after ID leaves sensor X", "from 10:00AM to 10:15AM", and "every 5 min."

More complicated events can be combined from basic events through logical operators, such as "AND", "OR", and "NOT".

We summarize the definition of events in the following EBNF-like recursive grammar.

Table 3.1: Service types and corresponding receiver IDs in VS.



	Receiver ID	Service Type
1	Single user	Unicast
2	Group	Multicast
3	Sensor	Geocast
4	ALL	Broadcast

$$
\begin{aligned}
<EvntVal> &= <SubEvntVal> \;\; AND \;\; <EvntVal> \;| \\
& <SubEvntVal> \;\; OR \;\; <EvntVal> \;| \\
& <SubEvntVal> \\
<SubEvntVal> &= (<EvntVal>) \,|\, NOT <SubEvntVal> \;|\; <SnglEvntVal> \\
<SnglEvntVal> &= <LocEvnt> \;|\; <TimeEvnt> \\
<LocEvnt> &= <ID><Rel><Sensor\_X> \\
<Rel> &= ENTER \,|\, LEAVE \\
<TimeEvnt> &= <min/hr/dat/mon/yr> | \\
& <TimeOfEvnt(LocEvnt)+min/hr> | \\
& <min/hr/dat/mon/yr, min/hr/dat/mon/yr> | \\
& <min/hr/dat/mon/yr, period>
\end{aligned}
$$

Note that *TimeOfEvnt(LocEvnt)* returns the actual time when a particular location event happens.

Actions in VS are all message transmission commands. The service types include unicast, multicast, geocast, and broadcast. The receiver ID (i.e., target of the transmission) can be a user, a group of users, or users in a particular space. The mapping between service type and receiver ID is illustrated in Table 3.1.

3.3 Service Operations and Message Flows

Based on the above definition of event-driven actions, we next discuss the operations of VS and some real message flows. We will explain through two examples in this subsection.

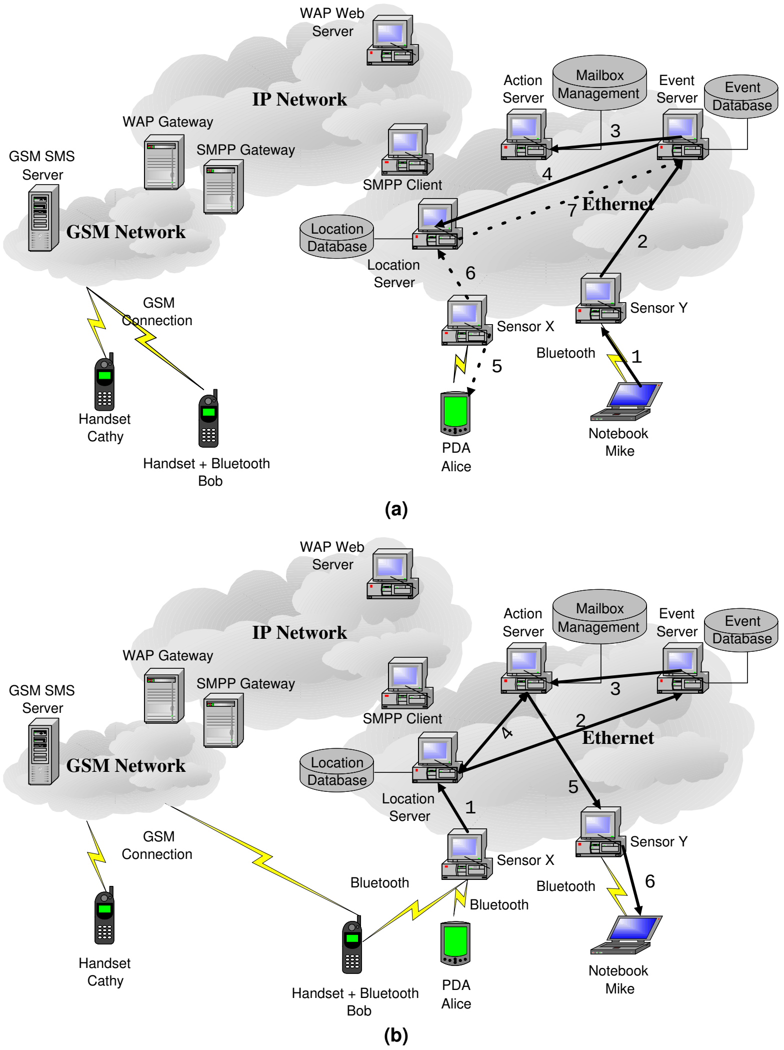


Figure 3.2: Message flows for a meeting-reminding instant message service.

The first example demonstrates how one is reminded by an instant message when a meeting is ready. Suppose that manager Mike calls for a meeting involving staffs Alice and Bob at 10:00AM. However, Mike does not want to be kept in waiting if anyone of Alice and Bob is late. So Mike uses a notebook installed with the VS Client program and sets up an event service: "On (time = <10:00, 10:10>) AND (Alice ENTER Sensor X) AND (Bob ENTER Sensor X) Do Unicast(Mike, "staff meeting is ready")".

Suppose that Mike is now in sensor Y. We illustrate the message flow in Fig. 3.2(a). At first, Mike configures the above event-driven action through sensor Y. In this case, since Mike uses a notebook with a Bluetooth, the Bluetooth is used as a data network to connect to the Event Server. Through our VS Client program, sensor Y will relay the request. On receiving the request, the Event Server parses the request, which includes a time event and two location events together with a unicast action. The time event will be stored locally at the Event Server, while the two location events will be forwarded to the Location Server. The Action Server will also be contacted to register the unicast action together with an index to the action. Note that the index will be stored at the Event Server, for which to trigger the corresponding action later on. Next, suppose that Alice, who has a PDA installed with VS, arrives at sensor X first. Then such an event will be captured by sensor X, which will conduct a location update with the Location Server. (Note that this update action is independent of the content of the Location Server; it is conducted whenever a sensor finds a device entering or leaving its coverage.) On receiving the update of Alice's location, the Location Server checks its local database and finds that such an event should be signaled to the the Event Server.

Later on, as shown in In Fig. 3.2(b), as Bob, who has a Bluetooth-enabled handset, arrives at the meeting room. Such an entering event will also be captured by sensor X and triggered to the Location Server and then the Event Server. Depending on whether the time event has become true or not, the Event Server may trigger the unicast action immediately or at later time. Once all events are satisfied, the Action Server will be triggered by the proper action index. The Action Server then queries Mike's current location and sends the reminder to Mike. On receiving an acknowledgement message from Mike, all related information at these servers will be cleaned.

Next, suppose that another staff Cathy needs Mike to sign a document but finds that Mike is still in the meeting. Suppose that Cathy does not want Mike to be disturbed, but wants to send herself a message three minutes after Mike leaves the meeting room. Therefore, Cathy may submit a request: "On (time = TimeOfEvnt(Mike LEAVE Sensor X) + 3) Do Unicast(Cathy, "Mike is available now")".

In Fig. 3.3(a), we show the message flow for Cathy's request through a WAP-enabled handset. In this case, the handset does not need to be Bluetooth-enabled. The configuration procedure will be done through our WAP Server. Note that in our implementation, we regard the cellular network as a single special sensor in our VS network. So the WAP Server will go through our SMPP Client to submit

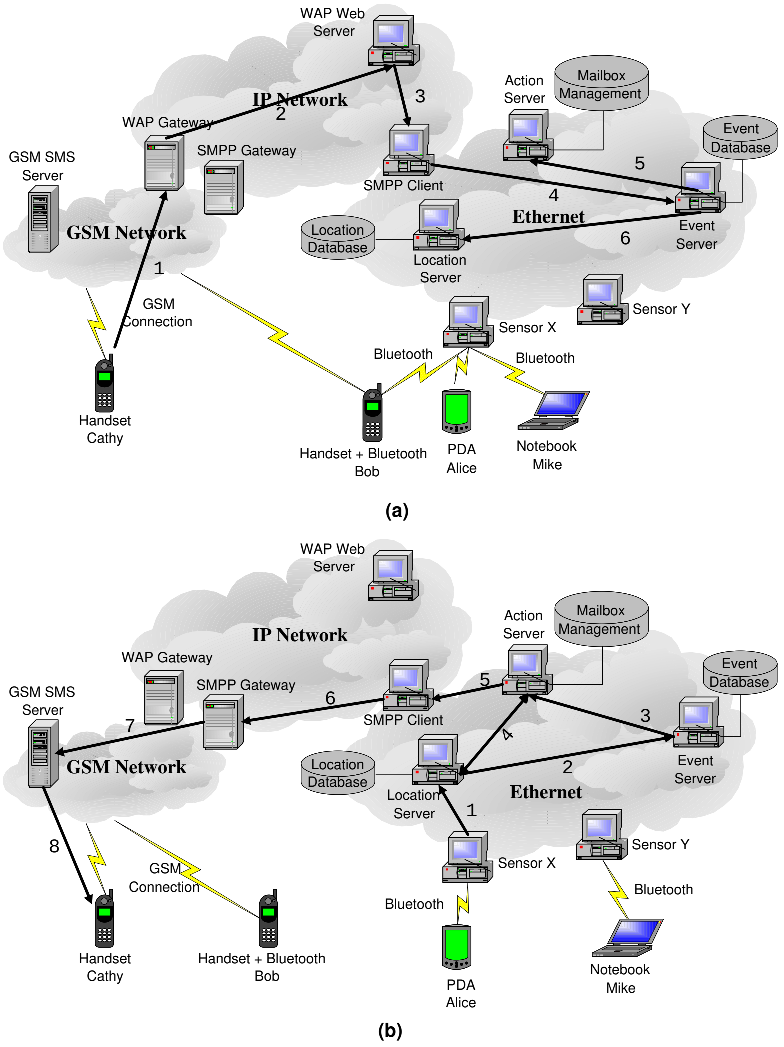


Figure 3.3: Message flows for a relative-time instant message service.

the event-driven action to the Event Server. In this way, the event server has a consistent view on users in VS, and all handset users are regarded as under a special "GSM sensor." Fig. 3.3(b) shows the rest of the message flow after Mike leaves the meeting room. Note that since Cathy is recognized as resident in the special "GSM sensor" network, the SMPP Client will be contacted to provide the instant message service. The SMPP Client will then contact the SMPP Gateway, which will connect to the GSM SMS Server, who actually delivers the short message.

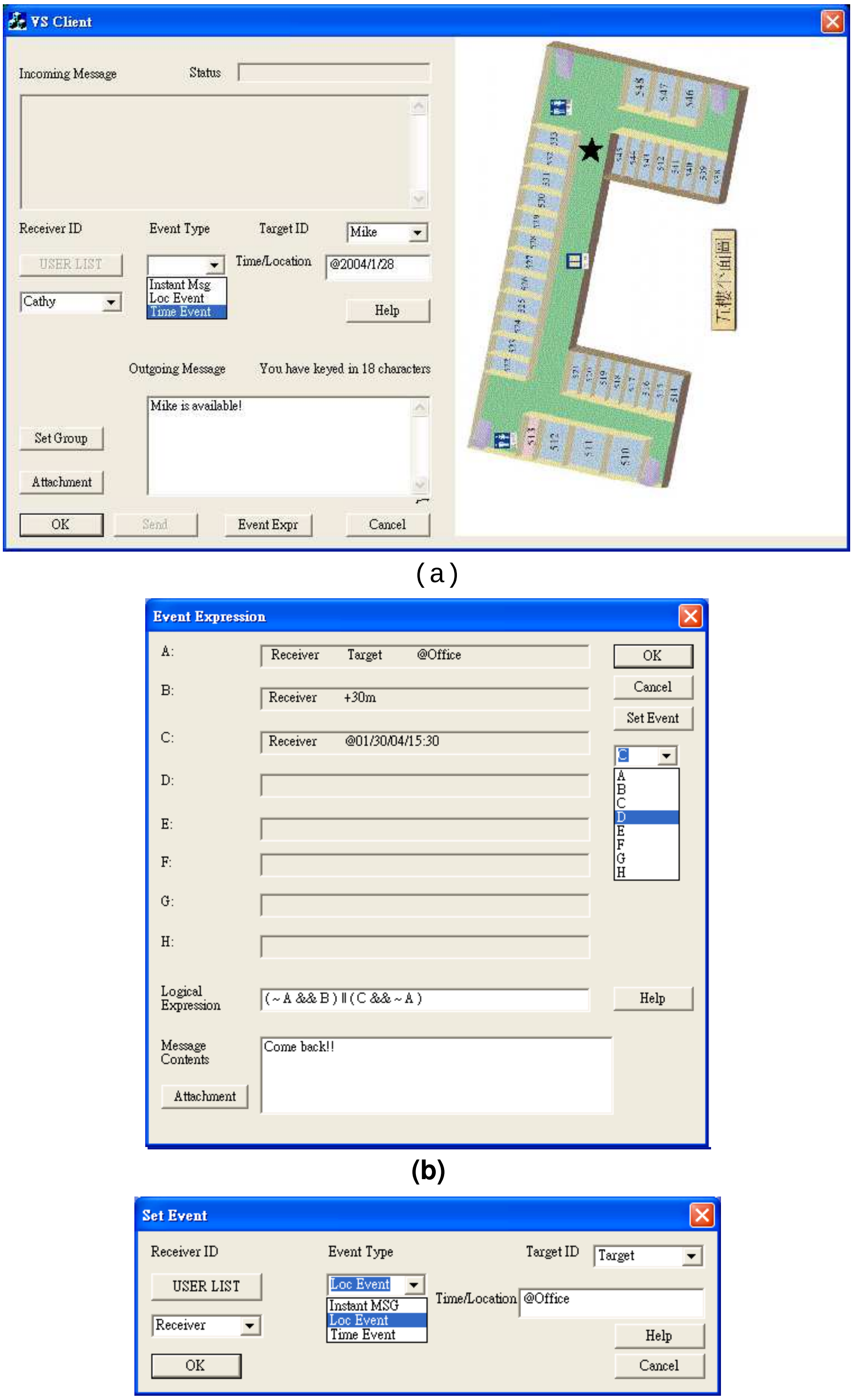
3.4 Implementation Details and User Interfaces

In this section, we demonstrate the configuration interfaces of VS. We first explain the interfaces for laptops/palmtops, followed by the interactive WAP web pages established for handsets.

Fig. 3.4(a) is the main user interface of the VS client software for program-loadable devices. On the right-hand side, the asterisk marks the current location of the user. The user can send an instant message by selecting "Instant Msg" as the event type. After filling in Receiver ID and Outgoing Message, clicking the OK button will submit the request. The user can configure a basic event as follows. To set up a basic time event, the Event Type field should be declared first, followed by filling a parameter in the Time/Location field with one of the following formats: @mm/dd/yy/hh:xx (for absolute time), +xx{m,s,h} (for relative time, xx minutes/seconds/hours after the current time), and #xx{m,s,h} $\sim$ yy{m,s,h} (for time interval, every xx time units, lasting for yy time units). To set up a basic location event, we first declare the Event Type field followed by entering desired sensor with the format: @SensorID. Then Target ID field should be declared by filling in either a single device or a group of devices.

To configure a compound event, one should click on the "Event Expr" in the main window. Then the Event Expression window as shown in Fig. 3.4(b) will pop up. Up to eight basic events (numbered A-H) can be defined. To define a basic event, the "Set Event" button should be clicked, which will bring up the interface as shown in Fig. 3.4(c). The setup procedure is similar to what is discussed earlier. After configuring all basic events, the user can enter a logic statement in the field Logical Expression in Fig. 3.4(b).

We also provide online Help through the main window. In addition, buttons



(a)

VS Client

Receiver ID: Cathy

Event Type: Time Event

Target ID: Mike

Time/Location: @2004/1/28

Outgoing Message: Mike is available!

You have keyed in 18 characters

(b)

Event Expression

A: Receiver Target @Office

B: Receiver +30m

C: Receiver @01/30/04/15:30

D:

E:

F:

G:

H:

Logical Expression: (~ A && B) || (C && ~ A)

Message Contents: Come back!!

(c)

Set Event

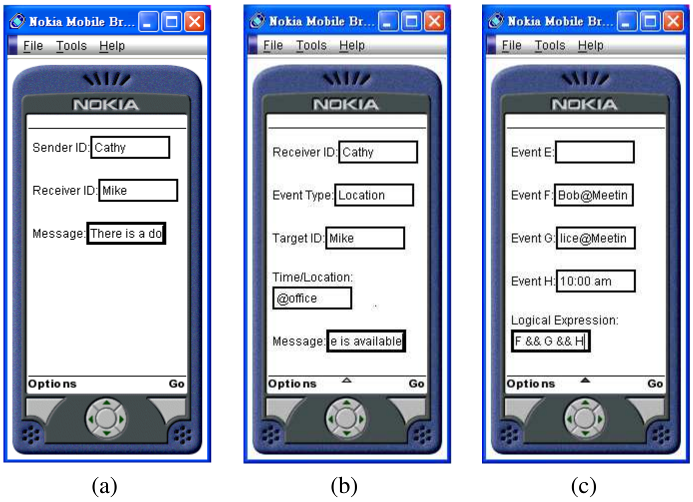
Receiver ID: Receiver

Event Type: Loc Event

Target ID: Target

Time/Location: @Office

Figure 3.4: Interfaces for laptop/palmtop terminals in VS.



Sender ID: Cathy

Receiver ID: Mike

Message: There is a do

(a)

Receiver ID: Cathy

Event Type: Location

Target ID: Mike

Time/Location:
@office

Message: e is available

(b)

Event E:

Event F: Bob@Meetin

Event G: lice@Meetin

Event H: 10:00 am

Logical Expression:
F && G && H

(c)

Figure 3.5: The interactive WAP web page interfaces for handsets in VS. (a) instant message, (b) single event configuration, and (c) hybrid event configuration.

USER LIST and Set Group facilitate users to check users currently visible by VS and to set up a group ID for multicast services. The Attachment option, though not discussed yet, supports simple file transfers between users. At the current stage, VS only allows text-based messaging services due to the limited bandwidth and MAC restrictions imposed by Bluetooth. Multimedia applications are being considered and will be directed to our future work.

The WAP interface is designed following the similar philosophy as discussed above, except that the setup procedure is decomposed into several small WAP pages. We are unable to provide all details due to space limitation. Fig. 3.5 shows some WAP web pages for handset configurations.

Chapter 4

Performance Issues

Our VS already performs well in terms of message delivery efficiency and reliability. Below, we discuss some performance issues, including sensing/detecting capability and collision problem.

4.1 Sensing and Detecting Capability

In VS, sensor nodes are required to perform Bluetooth inquiry periodically. We experiment on different inquiry intervals from 10 seconds to 1 minute. Intuitively, the larger the inquiry interval, the longer connection latency will be. However, a short inquiry interval means that sensors will spend significant time discovering devices. As a result, messages buffered at sensors may not be processed in time, leading to message drops. In addition, due to the inquiry frequency hopping rules of Bluetooth, it is suggested that each inquiry should last for at least 10 seconds. Otherwise, the “inquiry” from a sensor and the “inquiry scan” from a device are likely to miss each other, resulting in a failed discovery. Our results suggest that 25-30 seconds perform well, with average connection time no longer than 15 seconds. With the recent release of Bluetooth v1.2, the connection efficiency is claimed to be greatly improved. However, the firmware update is not available to us at this point of time. In fact, a lot of works have tried to reduce the inquiry efficiency for a single Bluetooth piconet [22, 25, 26, 29].

While most works in the literature only consider the inquiry delay in a single Bluetooth piconet, our work must rely on the sensing capability of multiple Bluetooth piconets deployed in a sensing field. How these piconets are arranged is an important issue. Below, we propose a model to evaluate the sensing capability of our VS. Suppose that we are given a sensing field A with a number of regularly or irregularly deployed sensors. Sensors are not synchronized in time (in terms of their timing to inquiry). Given any user appearing in any location in A, we would like to analyze the average latency L such that the user can be detected by any of the piconets after it appears. Let A_i be the area of A that is covered by exactly i sensors, $1 \leq i \leq n$, where n is the total number of sensors. (We assume that every point in A is covered by at least one sensor; otherwise, analyzing the average detection latency is meaningless.) Also, let L_i be the latency such that a user is detected by any piconet after it appears in a point in A_i. It is easy to see that

$$L = \frac{A_1}{A} \times L_1 + \frac{A_2}{A} \times L_2 + \cdots + \frac{A_n}{A} \times L_n.$$

Now let T be the inquiry interval and t be the duration of inquiry in each inquiry interval. It is easy to see that for a user appearing in any point in A_1, the detection latency is

$$L_1 = \frac{T-t}{T} \times \frac{T-t}{2}.$$

For a user appearing in any point in A_2, the latency is the time to the nearest inquiry from the two nearby sensors. The latency can be obtained by integrating the latency for each combination of the two inquiry durations of the two piconets. So we have

$$L_2 = 2 \times \int_0^t \frac{1}{T} \times \frac{(T-(x+t))^2}{2T} + \int_t^{T-t} \frac{1}{T} \times \frac{(x-t)^2 + (T-(x+t))^2}{2T}.$$

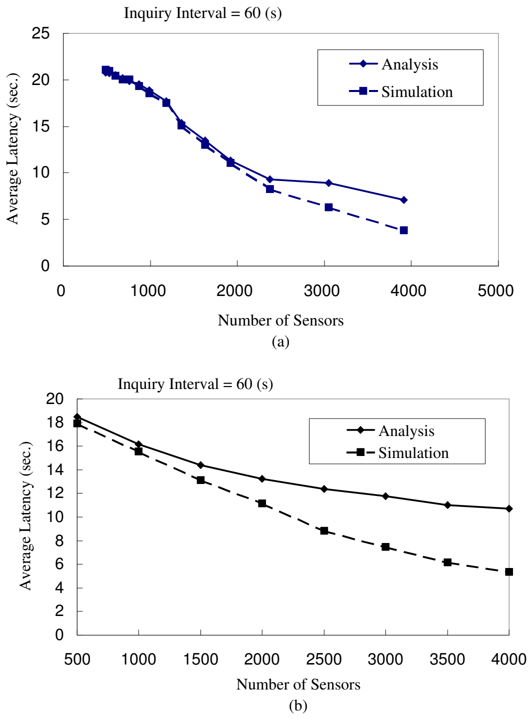
Similarly, we can derive the latency for a user appearing in any point in A_3 as follows:

$$L_3 = 6 \times \int_0^t \frac{1}{T} \times \int_x^{t+x} \frac{1}{T} \times \frac{(T-(y+t))^2}{2T} dydx + 6 \times \int_0^t \frac{1}{T} \times \int_{t+x}^{T-t} \frac{(y-(x+t))^2 + (T-(y+t))^2}{2T} dydx$$

$$2 \times \int_t^{T-2t} \frac{1}{T} \times \int_{x+t}^{T-t} \frac{1}{T} \times \frac{(x-t)^2 + (y-(x+t))^2 + (T-(y+t))^2}{2T} dydx.$$

We can derive $L_i, i \geq 4$ in a similar way, but the derivation will be more tedious.

Next, we have also conducted some simulations. A sensing field of size 500×500 is simulated. The radius of each Bluetooth coverage area is 10. We vary the number of sensors, n, and observe the detection latency. Fig. 4.1 (a) shows our result when sensors are deployed in a regular manner. A typical triangular deployment is



(a)

(b)

Figure 4.1: Average detection latency Vs. number of sensors with $T = 60$ sec. and $t = 10$ sec. (a) Regular deployment, and (b) random deployment

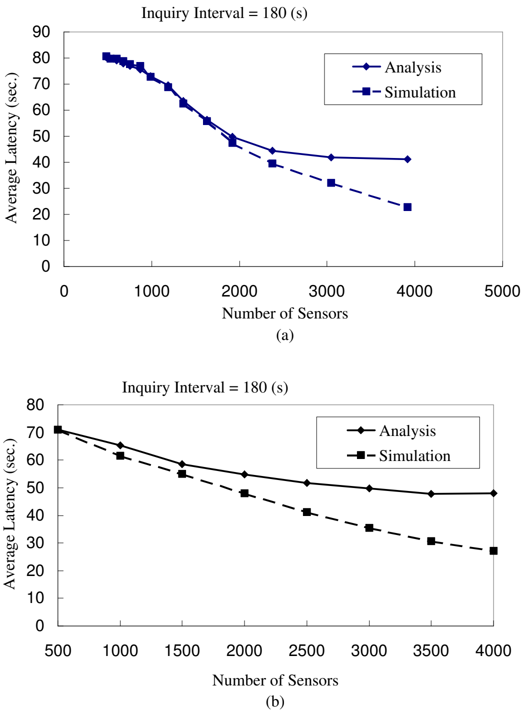
adopted, where each three nearby sensors form a triangle of the same side length. We adjust the side length to adjust the number of sensors. We set $T = 60sec$ and $t = 10sec$. For the analysis part, we approximate L by $\sum_{i=1}^{3} \frac{A_i}{A} \times L_i$. As can be seen, when $n \leq 2500$, the analysis result and the simulation result are very close. As n increases, the values of A_i for $i \geq 4$ are non-negligible, thus leading to more error. Fig. 4.1 (b) shows our result when sensors are randomly deployed. The gap between the analysis and the simulation results increases because with random deployment, sensors are more like to overlap, leading to a higher value of $A_i, i \geq 4$.

Fig. 4.2 shows a similar result with a larger value of inquiry interval $T = 180sec$.

4.2 Packet Collision Analysis

4.2.1 Problem Statement

In each Bluetooth piconet, the master and slaves take turns to exchange packets. While the master only transmits in even-numbered slots, slaves must reply in odd-



(a)

(b)

Figure 4.2: Average detection latency Vs. number of sensors with $T = 180$ sec. and $t = 10$ sec. (a) Regular deployment, and (b) random deployment

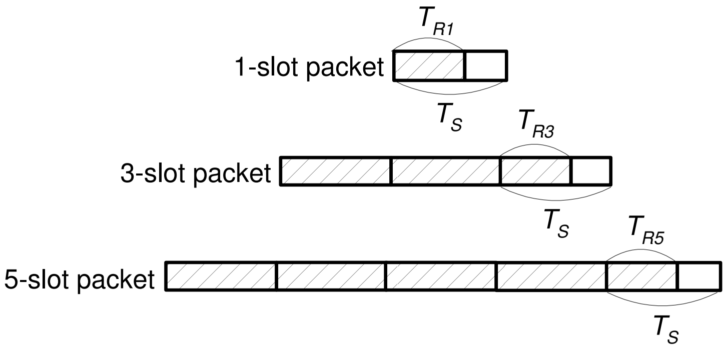


Figure 4.3: Frequency-hopping guard time in Bluetooth.

numbered slots. Three packet sizes are available: 1-slot, 3-slot, and 5-slot. For a multi-slot packet, its frequency is fully determined by the first slot and remains unchanged throughout.

Since Bluetooth takes a frequency-hopping channel model, each packet has a guard time at the packet end. As Fig. 4.3 illustrates, an i-slot packet actually does not fully occupy all the i slot(s). Let T_S be the length of one time slot. For a 1-slot packet, the data duration is T_{R1}. For 3-slot and 5-slot packets, the data durations at the last time slot are T_{R3} and T_{R5}, respectively. Those vacant periods without data transmission activities are designed mainly for radio transceiver turnover, preparing for stabilizing at the next frequency hop. Define $r_i, i = 1, 3, 5$, to be the corresponding *guard time occupancy ratio* for an i-slot packet,

$$r_1 = T_{R1}/T_S, \quad r_3 = T_{R3}/T_S, \quad r_5 = T_{R5}/T_S. \tag{4.1}$$

Since the differences between r_i's are very small in Bluetooth, below we approximate them by a single value r for simplicity, i.e., $r = r_1 = r_3 = r_5$.

We consider N piconets coexisting in a physically closed environment. Since no coordination is possible between piconets, each piconet has $N - 1$ potential competitors. In any time instance, if two piconets transmit with the same frequency, the corresponding two packets are considered damaged (note that during the guard time periods, a host is not considered transmitting). Our goal is to derive an analytic model to evaluate the impact of collisions in such a multi-piconet environment.

We assume a uniform traffic in each piconet, and let λ_1, λ_3, and λ_5 be the arrival rates of 1-, 3-, and 5-slot packets per slot, respectively, to a piconet. Note that for a multi-slot packet, only the header slot counts as arrival. It is easy to see that $\lambda_1 + 3\lambda_3 + 5\lambda_5 \leq 1$. Further, we regard the remaining vacant slots as "dummy" single-slot packets. Thus, the arrival rate of such dummy (1-slot) packets is $\lambda_0 = 1 - (\lambda_1 + 3\lambda_3 + 5\lambda_5)$.

4.2.2 Collision Analysis Without Considering Guard Time Effect

In this section, we first analyze collision effect without considering the guard time effect. Specifically, when this effect is not considered, two packets are considered damaged if they are transmitted using the same frequency and they have non-empty overlapping in their slot time (including both transmission period and guard time period). The review would facilitate presenting our result in the next section, which considers guard time effect.

Let's consider a piconet X and another competitor piconet Y, which is regarded as the unique source of interference to X. With the interference from Y, we first derive the success probability $P_S(i)$ of i-slot packets in X, where $i = 1, 3, 5$. We start by introducing the concept of "slot delimiter." Consider any slot in X. One or two slot delimiters in Y may cross X's slot. However, since we are considering continuous probability, the possibility of two crossing slot delimiters can be ignored, and thus we will deal with one crossing delimiter in the rest of the discussion. For example, for a 1-slot packet in X, it succeeds only if there is no interference from the two slots before and after the delimiter, so the success probability of X's packet could be 1, $\frac{78}{79}$, or $(\frac{78}{79})^2$, depending on whether Y transmits or not. Below, we denote the constant factor 78/79 by P_0.

Depending on what packet(s) is divided by it, a delimiter is classified into ten types (refer to Fig. 4.4):

- B_1, B_2, B_5: the beginning of a 1-, 3-, and 5-slot packet, respectively.
- B_3, B_4: the beginnings of the second and third slots of a 3-slot packet, respectively.
- B_6, B_7, B_8, B_9: the beginnings of the second, third, fourth, and fifth slots of a 5-slot packet, respectively.
- B_{10}: the beginning of a dummy slot.

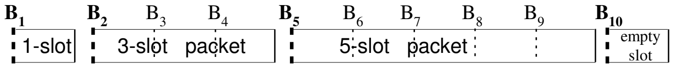


Figure 4.4: Classification of slot delimiters.

The rate of B_1 is λ_1 per slot; the rate of each of B_2, B_3, and B_4 is λ_3; the rate of each of B_5, B_6, B_7, B_8, and B_9 is λ_5; and the rate of B_{10} is λ_0. We denote the arrival rate of B_j by $\lambda(B_j)$, $j = 1..10$. Given any B_j, we also define $g(j)$ to be the number of slots that follows delimiter B_j and belong to the same packet. For example, $g(1) = 1$, $g(3) = 2$, $g(7) = 3$, and $g(10) = 1$.

Intuitively, when a packet in X is crossed by a delimiter of type $B_1/B_2/B_5$ in Y, there may exist two packets (of different frequencies) in both sides of the delimiter in Y which are potential sources of interference to X's packet. On the other hand, when the delimiter is of the other types, the interference source reduces to one.

Definition 1 Given any i-slot packet in piconet X and any interference source piconet Y, define $L(k), k < i$, to be the probability that the packet of X experiences no interference from Y starting from the delimiter of Y crossing the $(i-k+1)$-th slot of the packet to the end of the packet, under the condition that the aforementioned delimiter is of type $B_1/B_2/B_5/B_{10}$. For $k \leq 0$ (in which case the above definition is not applicable), $L(k) = 1$.

The above probability function is introduced in [19]. Intuitively, $L(k)$ is the success probability of the last k slots of X's packet excluding the part before the first delimiter of Y crossing these k slots, given the above delimiter type constraint. With this definition, we can find $P_S(i)$ by repeatedly cutting off some slots from the head of X's packet, until there is no remaining slot:

$$P_S(i) = \sum_{j=1}^{10} \lambda(B_j) \cdot f(j) \cdot L(i - g(j)), \tag{4.2}$$

where

$$f(j) = \begin{cases} (1-\lambda_0) \cdot P_0^2 + \lambda_0 \cdot P_0 & \text{if } j = 1, 2, 5 \\ (1-\lambda_0) \cdot P_0 + \lambda_0 & \text{if } j = 10 \\ P_0 & \text{otherwise} \end{cases} .$$

In the equation, we consider each type $B_j, j = 1..10$, of the first delimiter in Y crossing X's packet. The corresponding probability is $\lambda(B_j)$. Function $f(j)$ gives

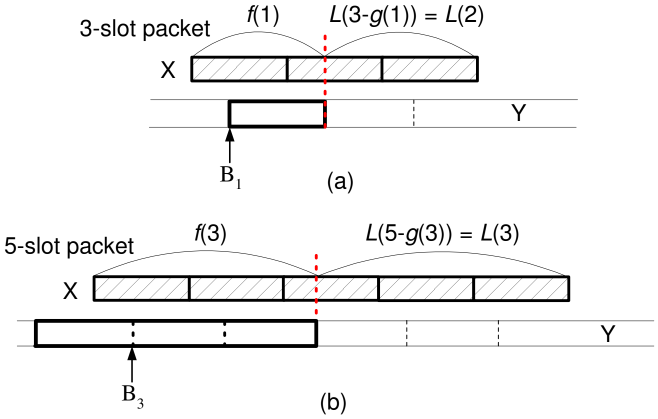


Figure 4.5: Analysis of success probabilities for (a) 3-slot and (b) 5-slot packets.

the probability that the packet(s) of Y on both sides of the first delimiter B_j does (do) not interfere with X's packet. It remains to consider the success probability of the last $i - g(j)$ slots of X's packet, excluding the part before the first delimiter of Y crossing these $i - g(j)$ slots (which must be of delimiter type $B_1/B_2/B_5/B_{10}$). This is reflected by the last factor $L(i - g(j))$.

For example, Fig. 4.5(a) illustrates a 3-slot packet in X. The first delimiter in Y crossing the 3-slot packet is of type B_1. The success probability of the first part in X is $f(1) = (1 - \lambda_0) \cdot P_0^2 + \lambda_0 \cdot P_0$. Intuitively, if the packet of Y before the delimiter B_1 is a dummy packet (of probability λ_0), the success probability is simply P_0; otherwise, there are two packets which are potential interference sources, and the success probability is P_0^2. Then we can move on to consider the success probability of the remaining part of X after the second delimiter in Y, which is given by $L(2)$. Another example of a 5-slot packet is shown in Fig. 4.5(b). The first delimiter in Y crossing the 5-slot packet is of type B_3. So the success probability from the beginning of the packet up to the third delimiter in Y crossing the packet is $f(3)$. For the remaining part, the success probability is $L(3)$. So the success probability of the 5-slot packet is $f(3) \cdot L(3)$.

The remaining part of X's packet covered by $L(k)$ must start with a delimiter in Y of a restricted type of $B_1/B_2/B_5/B_{10}$. Since the packet in Y after the delimiter must be a complete packet, it can be solved recursively as follows ($k > 0$):

$$\begin{aligned} L(k) &= \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_3 + \lambda_5} \cdot L(k - g(10)) + \\ & \frac{\lambda_1}{\lambda_0 + \lambda_1 + \lambda_3 + \lambda_5} \cdot P_0 \cdot L(k - g(1)) + \\ & \frac{\lambda_3}{\lambda_0 + \lambda_1 + \lambda_3 + \lambda_5} \cdot P_0 \cdot L(k - g(2)) + \\ & \frac{\lambda_5}{\lambda_0 + \lambda_1 + \lambda_3 + \lambda_5} \cdot P_0 \cdot L(k - g(5)). \end{aligned} \tag{4.3}$$

In each term, the first part is the probability of the corresponding packet type in Y. As to the boundary conditions, $L(k) = 1$, for $k \leq 0$.

Next, we consider an N-piconet environment. For each piconet X, there are $N - 1$ piconets each serving as an interference source. Since these interferences are uncoordinated and independent, the success probability of an i-slot packet in X can be written as $P_S(i)^{N-1}$. So the network throughput of X is:

$$\begin{aligned} T &= \lambda_1 \cdot P_S(1)^{N-1} \cdot R_1 + 3 \cdot \lambda_3 \cdot P_S(3)^{N-1} \cdot R_3 + \\ & 5 \cdot \lambda_5 \cdot P_S(5)^{N-1} \cdot R_5, \end{aligned} \tag{4.4}$$

where R_1, R_3, and R_5 are the data rates (bits/slot) of 1-, 3-, and 5-slot packets, respectively (for example, if DH1/DH3/DH5 are used, $R_1 = 216$, $R_3 = 488$, and $R_5 = 542.4$). The aggregate network throughput of N piconets is $N \times T$.

4.2.3 Enhanced Collision Analysis Considering Guard Time Effect

Next, we improve the collision analysis reported in [19] by considering the guard time effect. During guard time periods, hosts are considered not transmitting. So collisions may only occur in real transmission periods. Still, we consider a piconet X and another competitor piconet Y, which is the source interference of X. With guard time effect, the probability $P_S(i)$ should be reformulated as follows:

$$\begin{aligned} P_S(i) &= (1 - r) \cdot \sum_{j=1}^{10} \lambda(B_j) \cdot \tilde{f}(j) \cdot L(i - g(j)) + \\ & (2r - 1) \cdot \sum_{j=1}^{10} \lambda(B_j) \cdot f(j) \cdot L(i - g(j)) + \\ & (1 - r) \cdot \sum_{j=1}^{10} \lambda(B_j) \cdot f(i, j) \cdot \tilde{L}(i - g(j)), \end{aligned} \tag{4.5}$$

where

$$\tilde{f}(j) = \begin{cases} 1 & \text{if } j = 10 \\ P_0 & \text{otherwise} \end{cases}$$

and

$$f(i,j) = \begin{cases} (1-\lambda_0)\cdot P_0 + \lambda_0 & \text{if } i = 1 \text{ and } j = 1,2,5 \\ (1-\lambda_0)\cdot P_0^2 + \lambda_0 \cdot P_0 & \text{if } i = 3,5 \text{ and } j = 1,2,5 \\ (1-\lambda_0)\cdot P_0 + \lambda_0 & \text{if } j = 10 \\ P_0 & \text{otherwise} \end{cases}.$$

$\tilde{L}(k)$ can be derived in two cases. When $k > 1$,

$$\begin{aligned} \tilde{L}(k) \quad = \quad & \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_3+\lambda_5}\cdot L(k-g(10)) + \\ & \frac{\lambda_1}{\lambda_0+\lambda_1+\lambda_3+\lambda_5}\cdot P_0 \cdot L(k-g(1)) + \\ & \frac{\lambda_3}{\lambda_0+\lambda_1+\lambda_3+\lambda_5}\cdot P_0 \cdot L(k-g(2)) + \\ & \frac{\lambda_5}{\lambda_0+\lambda_1+\lambda_3+\lambda_5}\cdot P_0 \cdot L(k-g(5)), \end{aligned} \tag{4.6}$$

and when $k \leq 1$, $\tilde{L}(k) = 1$.

In Eq. (4.5), the definitions of $\lambda(B_j)$, $f(j)$, and $g(j)$ remain the same as those in Eq. (4.2). Eq. (4.6) differs from Eq. (4.3) in its boundary conditions. To explain the above formulations, we introduce the concept of *critical section* (CS) within a single time slot. Since guard time periods are interleaving real data packets, the position (or offset) of the first slot delimiter B_j in Y crossing X's packet does affect the packet success probability $P_S()$. So we partition the first slot of any X's packet into three critical sections, CS1, CS2, and CS3, which occupy $1-r$, $2r-1$, and $1-r$ of the packet, respectively (recall that r is the approximated guard time occupancy ratio). An illustration is in Fig. 4.6.

Fig. 4.6 shows how the slot delimiter of Y may cross the above critical sections. In the first case, the delimiter B_j falls in X's CS1. If fortunately B_j equals B_1, B_2, B_5, or B_{10} (i.e., the beginning of a new packet), the darkened area before B_j in Y belongs to guard time. So the CS1 of X would experience no interference from Y. In this case, the packet success probability $P_S()$ will increase, and this effect is reflected in the new function $f(i,j)$ (compared to the function $f(j)$ in Eq. (4.2)).

In the second case of Fig. 4.6, the delimiter B_j falls in X's CS2. The aforementioned benefit would disappear because certain part in the beginning of CS1 will fall

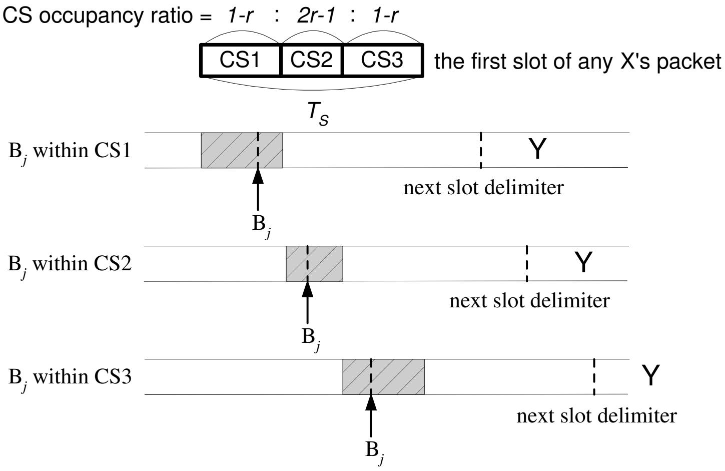


Figure 4.6: Critical sections and collision analysis by considering guard time effect.

out of the range of Y's guard time period. So $P_S()$ remains the same in this case (compared to Eq. (4.2)).

In the third case of Fig. 4.6, the delimiter B_j resides in X's CS3. If fortunately the packet in X is a single-slot packet, then CS3 of X would be guard time. If so, the darkened area in Y would not pose any interference to X's packet. Even if the packet in X is not a single-slot packet, this also means that the last slot delimiter of Y crossing X's packet would fall in a CS3 of X, a guard time period. If this delimiter is the beginning of a packet, $P_S()$ can also benefit in this case. This is reflected by ending the recursive formula $\tilde{L}()$ at earlier time.

Below we give some examples for our analysis.

- **Case I: B_j within CS1 (with probability $1 - r$)**

  Consider the example in Fig. 4.7(a). Benefiting from the guard time of Y's packet before B_1, the packet success probability of X's 1-slot packet is P_0. In comparison, without considering the guard time effect, the formulation in [19] would suggest a lower success probability of ${P_0}^2$.

- **Case II: B_j within CS2 (with probability $2r - 1$)**

  Consider the example in Fig. 4.7(b). The packet success probability of X's 1-slot packet is ${P_0}^2$, which remains the same as that in the analysis of [19]. Guard time does not reduce the probability of interference from Y's packet(s).

- **Case III: B_j within CS3 (with probability $1 - r$)**

  In Fig. 4.7(c), the packet in X under consideration is a 3-slot packet. The CS3

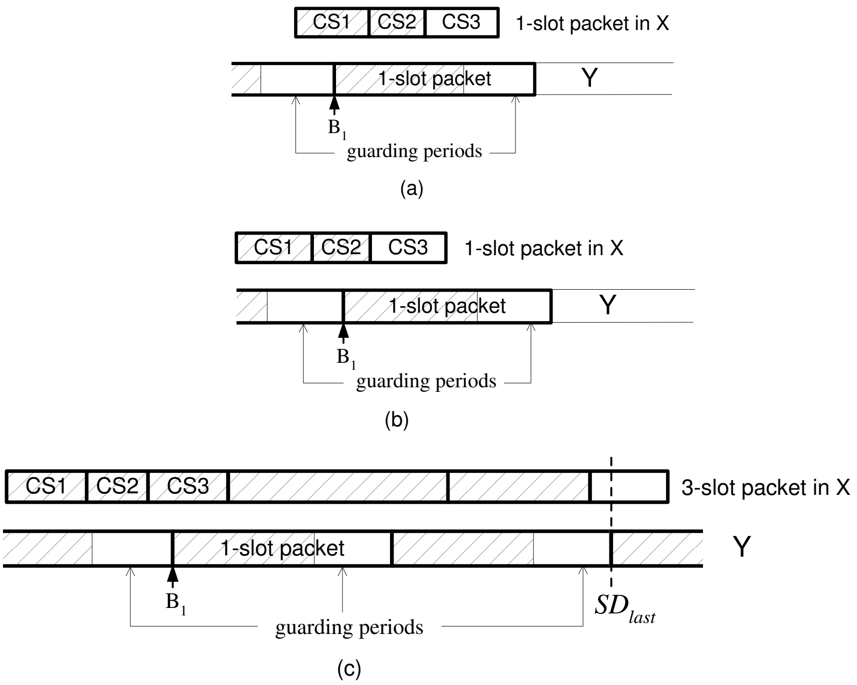


Figure 4.7: Collision analysis examples: B_j falling in (a) CS1, (b) CS2, and (c) CS3.

in X's first slot does not represent a guarding period in this example. Hence the two packets of Y right before and after the first delimiter B_1 still pose as a potential interference source to X's packet. However, since B_1 falls in CS3, the last slot delimiter of Y crossing X's packet (denoted by SD_{last} in the figure) must reside in a guarding period. So no interference needs to be taken into account after SD_{last}. In this example, the success probability of X's packet is ${P_0}^3$ (compared to a success probability of ${P_0}^4$ in the analysis of [19] without considering guard time effect). These are reflected in the function $f(i,j)$ and in the boundary conditions for $\tilde{L}(k)$.

Finally, for an N-piconet environment, the network throughput T of piconet X can be derived by the same Eq. (4.4). The aggregate network throughput of N piconets is therefore $N \times T$.

We remark that we use the uniform guard time occupancy ratio r to approximate r_1, r_3, and r_5. One concern is that the proposed analysis may have a certain level of bias. We examine this concern through simulation experiments in the following section.

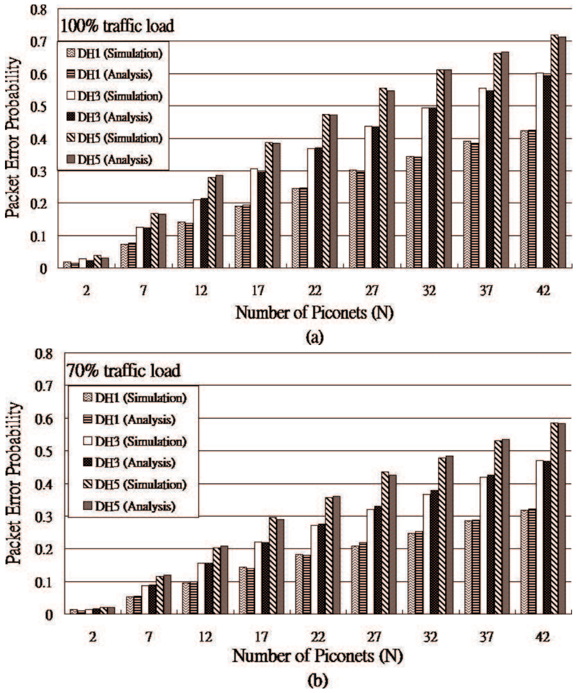


Figure 4.8: Packet error probabilities under (a) 100% traffic load and (b) 70% traffic load.

4.2.4 Simulation and Experimental Results

This section presents our simulation and experimental results based C++ programs. We test different numbers of piconets. Each simulation run lasts for 10000 time slots. Frequency hopping sequences are simulated by random sequences. For simulation results, we use the exact value of r_1, r_2, and r_3. For analytical results, we use the approximated r. Packets being simulated are DH1/3/5.

Assuming $\lambda_1 = \lambda_3 = \lambda_5$, we inject traffic loads of 100% and 70% (the percentage of busy slots) to each piconet (i.e., $\lambda_1 + 3\lambda_3 + 5\lambda_5 = 1$ and 0.7). Fig. 4.8 plots the error probabilities of DH1/3/5 packets under different Ns. According to the figure, the differences between simulation and analytical results are very small. The packet error probability increases as the traffic load or the number of piconets grows. Smaller packets suffer less collisions than larger ones due to the formers' shorter transmission durations. However, larger packets are much more bandwidth-efficient than smaller ones (e.g., a DH5 carries 542.4/216 times more bits per slot than a DH1 does). This observation leads us to conduct the next experiment by using network throughput as the metric.

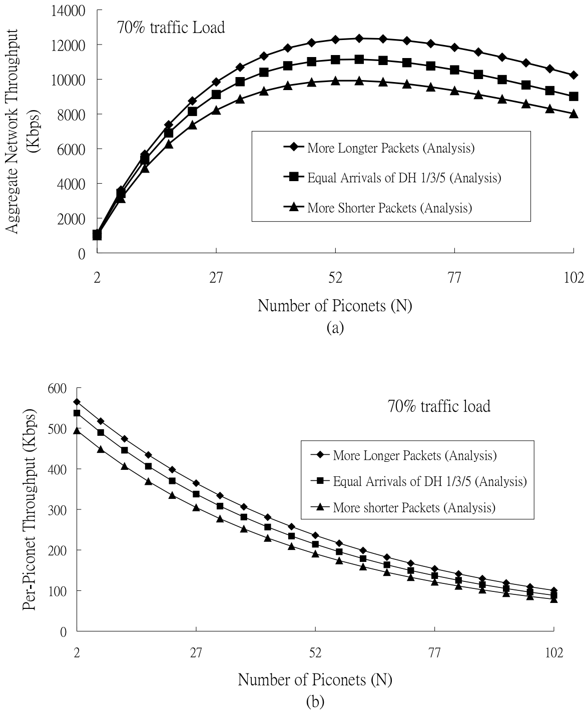


Figure 4.9: (a) Aggregate network throughput under 70% traffic load, and (b) per-piconet throughput under 70% traffic load.

Next we evaluate the aggregate network throughput ($N \times T$) and per-piconet throughput. We show the case of 70% traffic load. We consider three arrival models: one with equal arrivals of DH1/3/5 packets ($\lambda_1 = \lambda_3 = \lambda_5$), one with more shorter packets ($\lambda_1 : \lambda_3 : \lambda_5 = 3 : 2 : 1$), and one with more longer packets ($\lambda_1 : \lambda_3 : \lambda_5 = 1 : 2 : 3$). The results are in Fig. 4.9. The aggregate throughput saturates at a certain point as the number of piconets increases, and then drops sharply. Different from the earlier observation, the result indicates that longer packets are more preferable in terms of throughput because the collision problem can be compensated by the benefit of higher bandwidth efficiency. Also, in terms of per-piconet throughput, the performance consistently degrades as N increases, which is reasonable due the impact of increased packet error probability.

Fig. 4.10 plots the network throughputs vs. traffic loads under fixed values of N. It indicates that the throughput goes up steadily as traffic load increases when $N \leq 42$. However, for larger N's, throughputs saturate at certain points, due to more serious collisions. The results can be used to determine the number of piconets

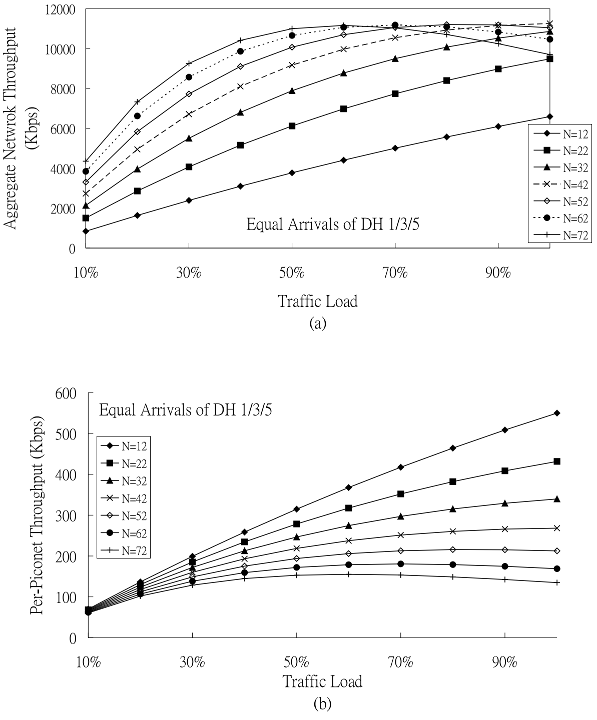


Figure 4.10: (a) Aggregate network throughput vs. traffic load, and (b) per-piconet throughput vs. traffic load.

to be deployed in a physical area.

Finally, we observe the guard time effect. Fig. 4.11 compares the packet error probability and network throughput when the effect is considered and when the effect is not considered (i.e., the case in [19]) under 70% traffic load with equal arrivals of DH1/3/5 packets. For example, due to less stronger conditions for packet collision, at $N = 42$, when considering the guard time effect, the packet error probabilities for DH1/3/5 packets are reduced by about 0.1, 0.09, and 0.075, respectively, and the aggregate network throughput is improved by about 1,000 Kbps. The error for DH1 is more significant because guard time takes larger part in such packets. As to throughput, the improvement when considering guard time effect tends to increase as N increases.

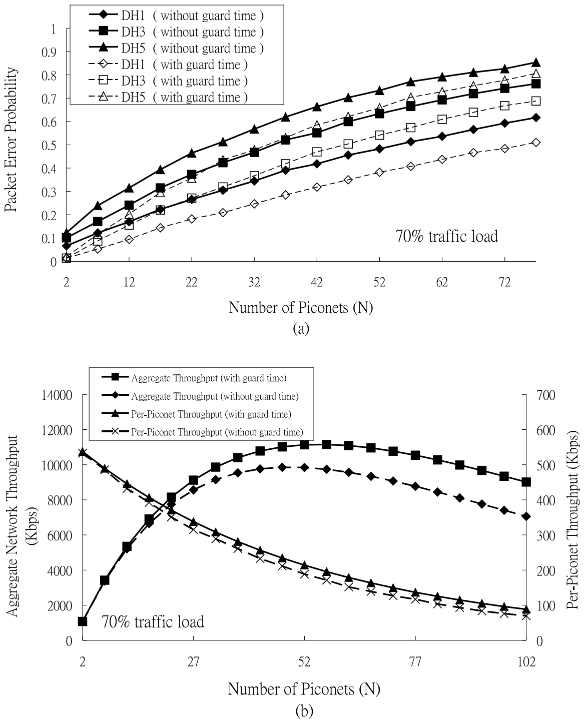


Figure 4.11: Guard time effect on: (a) packet error probability and (b) network throughput.

Chapter 5

Conclusions and Future Directions

In this thesis, we have reported how we prototyped an event-driven instant messaging application over integrated telecomm and datacomm networks. The proposed Visitor System in its current stage exercises both GSM and Bluetooth technologies. With the recent release of Bluetooth v1.2, which emphasizes on faster connection and better interference mitigation, we expect to improve the performance of our system.

Our VS has been put into operation. Many extensions and optimizations are continuously under development, aiming to provide human-centric services. Two extensions are expected to be realized in the near future, which are described below:

- **User Profile Management**

  Currently, VS recognizes senders/receivers by device IDs. However, one disadvantage is that device ID provides little information on the ownership of the user. It is also possible that a user owns multiple devices. The problem is that the system is unaware of which mobile device the user is carrying. One solution is to maintain a user profile, which registers all devices he/her owns. Once we have such information, multicast could be a solution to this problem. With user profiles, several user-centric applications are possible.

- **More Value-Added Services**

  Limited by Bluetooth and GSM wireless bandwidth, the current VS supports mainly text-based services. We have experimented some graphics files, such as maps, and the transmission performance is acceptable. However, supporting multimedia streaming services requires much more works. With the popularity of GSM MMS services, one extension step is to implement MMS transmissions in VS. This will require modifications in the Action Server.

As we have predicted, unifying novel applications across heterogeneous networks is an attractive trend. From the prototyping experiences, we have shown how to separate logical servers (such as event, location, and action servers) for event-driven messaging services, and the importance of an open interface for third-party application development. Our module-based system design allows software programmers to configure their own systems by just adding or modifying components without changing the base architecture. This flexibility is critical when implementing a real system. We have also demonstrated how to analyze the sensing/detecting capability of a sensor network at the deployment stage and the collision effect due to Bluetooth co-channel interference.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. *IEEE Communications Magazine*, pages 102–114, Aug. 2002.

[2] A. Baptista, T. Leen, Y. Zhang, A. Chawla, D. Maier, W. C. Feng, W. C. Feng, J. Walpole, C. Silva, and J. Freire. Environmental Observations and Forecasting Systems: Vision, Challenges and Successes of a Prototype. In *Proc. Int'l Annual Conf. Society for Environmental Information Sciences (ISEIS)*, 2003.

[3] J. Beutel, O. Kasten, M. Ringwald, F. Siegemund, and L. Thiele. Bluetooth Smart Nodes for Mobile Ad-hoc Networks. *TIK-Report No: 67*, 2001.

[4] J. Cao, X. Feng, and S. K. Das. Mailbox-Based Scheme for Mobile Agent Communications. *IEEE Computer*, pages 54–60, 2002.

[5] A. El-Hoiydi. Interference Between Bluetooth Networks - Upper Bound on the Packet Error Rate. *IEEE Communications Letters*, 5(6):245–247, June 2001.

[6] A. El-Hoiydi and J. D. Decotignie. Soft Deadline Bounds for Two-Way Transactions in Bluetooth Piconets under Co-Channel Interference. In *Proc. IEEE Int'l Conf. Emerging Technologies and Factory Automation*, pages 144–151, Oct. 2001.

[7] J. Haartsen, W. Allen, and J. Inouye. Bluetooth: Vision, Goals, and Architecture. *Mobile Computing and Communications Review*, 1(2), 1998.

[8] http://messenger.msn.com/. MSN Messenger.

[9] http://nms.lcs.mit.edu/projects/blueware. MIT Blueware Project. 2001.

[10] http://web.icq.com/. ICQ Homepage.

[11] http://www.aimglobal.org/technologies/rfid/. AIM RFID Technology.

[12] http://www.bluetooth.com. Bluetooth SIG Specification v1.2. Nov. 2003.

[13] http://www.gsmworld.com/technology/sms/index.shtml. GSM Short Message Service.

[14] http://www.nttdocomo.com/corebiz/imode/global/. NTT DoCoMo i-mode.

[15] E. Isaacs, A. Walendowski, and D. Ranganathan. Mobile Instant Messaging Through Hubbub. *Communications of the ACM*, 45(9):68–72, Sep. 2002.

[16] J. Kahn, R. Katz, and K. Pister. Mobile Networking for Smart Dust. *Mobile Computing and Networking*, 1999.

[17] O. Kasten and M. Langheinrich. First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network. In *Proc. First German Workshop on Mobile Ad Hoc Networks*, Mar. 2002.

[18] Y. Lim, J. Kim, S. L. Min, and J. S. Ma. Performance Evaluation of the Bluetooth-based Public Internet Access Point. In *Proc. IEEE Int'l Conf. Information Networking*, pages 643–648, 2001.

[19] T.-Y. Lin and Y.-C. Tseng. Collision Analysis for a Multi-Bluetooth Picocells Environment. *IEEE Communications Letters*, 7(10):475–477, Oct. 2003.

[20] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *Proc. First ACM Int'l Workshop on Wireless Sensor Networks and Applications*, 2002.

[21] N. Milanovic, A. Radovanovic, B. Cukanovic, A. Beric, N. Tesovic, and G. Marinkovic. Bluetooth Ad-hoc Sensor Network. *Univ. of Belgrade Technical Report*, 2002.

[22] P. Murphy, E. Welsh, and J. P. Frantz. Using Bluetooth for Short-Term Ad-Hoc Connections Between Moving Vehicles: A Fesability Study. *IEEE Vehicular Technology Conference*, 2002.

[23] G. Pottie and W. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43, May 2000.

[24] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom)*, Aug. 2000.

[25] T. Salonidis, P. Bhagwat, and L. Tassiulas. Proximity awareness and fast connection establishment in Bluetooth. *Proc. of Mobile and Ad Hoc Networking and Computing, 2000 (MobiHOC'00)*, 2000.

[26] F. Siegemund and M. Rohs. Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices. *Proc. 1st International Conference on Architecture of Computing Systems*, pages 256–273, 2002.

[27] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Trans. On Information Systems (TOIS)*, 10(1):91–102, Jan. 1992.

[28] B. Warneke, M. Last, B. Liebowitz, and K. S. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *IEEE Computer*, pages 2–9, Jan. 2001.

[29] R. Woodings, D. Joos, T. Clifton, and C. D. Knutson. Rapid heterogeneous connection establishment: Accelerating bluetooth inquiry using irda. *Proceedings of the Third Annual IEEE Wireless Communications and Networking Conference (WCNC '02)*, 2002.

Curriculum Vita

Yen-Ku Liu (yklau@csie.nctu.edu.tw) received his B.S. degree in Computer Science from the National Chiao-Tung University, Taiwan, in 2002. His research interests include wireless network, sensor network and Bluetooth.