

## Design and Implementation of Ad Hoc Communication and Applications on Mobile Phone Terminals

YUJIN NOISHIKI,<sup>†</sup> HIDETOSHI YOKOTA<sup>†</sup> and AKIRA IDOUE<sup>†</sup>

Recent advances in wireless communication technologies have attracted ad hoc networking, which enables mobile users to communicate with each other without any infrastructure. While many ad hoc routing protocols have been proposed, ad hoc communication and its applications are not widespread. The handiest wireless devices, namely, mobile phone terminals, are expected to expedite widespread use of ad hoc communications. More mobile phones have been equipped with short distance wireless interfaces such as wireless LAN and Bluetooth. When ad hoc communication is realized on such mobile phones with these wireless interfaces, ad hoc communications will be available anytime, anywhere, which we believe will become a driving force for ad hoc networking technology. However, although the performance of mobile phones has improved greatly over the past several years, their resources are still limited compared to laptop PCs. In this paper, we first present design principles for mobile ad hoc communications based on cases of assumed use. According to the design principles, we implement ad hoc networking protocols as well as their applications on mobile phones to verify their effectiveness. Based on the results of evaluations, we discuss the performance characteristics and potential application areas.

### 1. Introduction

Rapid deployment of wireless PAN technologies such as Bluetooth<sup>1)</sup> and ZigBee is expected to enable mobile users and devices in their vicinity to access and exchange information more easily. Communication ranges will become even more flexible and extensible by using mobile ad hoc networking technologies. To realize communications on mobile ad hoc networks, however, there are several issues to be solved such as the slow deployment of ad hoc networking capabilities on mobile terminals.

Many ad hoc routing protocols have been proposed under the initiative of the IETF Mobile Ad hoc Network Working Group (MANET WG). Ad hoc On-demand Distance Vector<sup>2)</sup> (AODV) routing and Optimized Link State Routing<sup>3)</sup> (OLSR) have been standardized as RFCs in the IETF. While many studies examine their method by computer simulation, there are few reports on the evaluation of practical implementations. Evaluations in a real device environment are important when considering actual user operations.

It is becoming common for mobile terminals to have multiple wireless interfaces. Besides laptop PCs and PDAs, mobile phone terminals also have short-range wireless interfaces, such as wireless LAN and Bluetooth. When ad hoc

network communication is realized in mobile phone terminals, the provision of additional services becomes possible. As mobile phones can connect to core networks through cellular networks, they enable applications that co-operate with core networks. In addition, the portability of mobile phone terminals is becoming the driving force for ad hoc networking technology.

Attractive applications are required for ad hoc communication to become widespread. In ad hoc networks, where there is no infrastructure, Peer-to-Peer (P2P) applications may be suitable because P2P communication is conducted without any central servers. Furthermore, concurrent connection support to core networks makes conventional server-oriented applications, e.g., Web services, easy to use on ad hoc networks. As each application has unique characteristics, the evaluation of applications in practical implementations is as important as that of routing protocols from the viewpoint of users.

This paper presents the design principles of ad hoc communication and applications on practical mobile devices from the viewpoint of mobile communication users. We focus on three design principles, terminal device selection, discovery of communication targets and protocol efficiency. According to these design principles, we implemented ad hoc networking protocols as well as their applications on mobile phone terminals. Our implementation used Bluetooth

---

<sup>†</sup> KDDI R&D Laboratories, Inc.

as the short-range wireless interface. Ad hoc communications and applications were implemented on the BREW<sup>4)</sup> platform. The implementation architecture was composed of a radio control unit that manages Bluetooth communication, a network control unit that establishes a route on ad hoc networks and a user application control unit. In the user application control unit, P2P chat communication and file-sharing applications were implemented. We evaluated the performance of our implementation and verified its effectiveness.

The rest of this paper is organized as follows. In Section 2, we review related work, focusing on the implementation of ad hoc networks. Section 3 presents the design principles for ad hoc communication and its applications. Section 4 presents the implementation of ad hoc networks on mobile phone terminals and Section 5 presents the evaluation of our implementation. Finally, Section 6 concludes the paper.

## 2. Related Work

Other research groups have implemented ad hoc routing protocols. AODV, an on-demand routing protocol, is implemented on practical machines<sup>5),6)</sup>. Both implementations support the Linux-based platform. AODV-UU<sup>5)</sup> is also implemented on ARM-based architecture PDAs. OLSR is a proactive routing protocol and is implemented on Linux, Windows and mobile devices with the ARM architecture<sup>7),8)</sup>. These implementations use wireless LAN as the short-range wireless interface.

Bluetooth is a short-range wireless interface that enables ad hoc networking. Bluetooth specification defines a Piconet as a network where up to eight terminals connect with each other. In the Piconet, a master terminal controls multiple simultaneous connections to slave terminals<sup>9)</sup>. Scatternet is defined as a group of independent and non-synchronized Piconets. Two Piconets communicate via gateway terminals that belong to both Piconets. The Piconet and the Scatternet are special types of ad hoc networks. Haartsen<sup>9)</sup> reports that Piconets and Scatternets increase the efficiency of channel use by the Frequency Hop spread spectrum as compared to wireless LAN, which shares frequency in the same radio propagation area.

The cluster-based ad hoc routing protocol is proposed for Piconets and Scatternets<sup>10)</sup>. In cluster-based ad hoc routing protocols, a cluster head manages routing and traffic forwarding in

the cluster. As Scatternet consists of multiple Piconets managed by master terminals, cluster-based ad hoc routing protocols are suitable for Piconets and Scatternets. However, in cluster-based ad hoc routing protocols, data transmission tends to concentrate on only some specified terminals, such as the cluster head or the master terminal in a Piconet. When terminals have limited wireless resources and battery power, it is necessary to examine routing protocols in which all terminals consume available resources equally.

In the Bluetooth specification, the Inquiry process is defined to find adjacent terminals. It requires several seconds to finish this process. Ronai, et al.<sup>11)</sup> proposed a new adjacent terminal discovery process, which improves the discovery time by simplifying the Inquiry process. As there are some differences between wireless LAN ad hoc networks and Bluetooth ad hoc networks, e.g., the Inquiry process, we need to take these differences into account. While many studies have evaluated Bluetooth ad hoc networks using computer simulation, some research groups have reported implementations on laptop PCs. Formo, et al.<sup>12)</sup> reported the design and implementation of indoor positioning over Bluetooth ad hoc networks. The proposed system measures the positioning of devices and transmits positioning results via Bluetooth ad hoc networks.

While some studies report implementation on PCs or PDAs, implementation on mobile phone terminals has not been reported. As mobile phone terminals have limited memory and power resources compared to laptop PCs or PDAs, we need to consider these limitations for design and implementation.

It is important to consider applications suitable for ad hoc networks. P2P type applications are attractive candidates because their distributed behavior fits ad hoc networks.

Many P2P file-sharing applications manage the key and location of a communication target in a distributed manner by using Distributed Hash Table (DHT) mechanisms<sup>13)</sup>. For example, target key is the desired content name, and target location is the IP address of the content holder. Since DHT mechanisms provide only a logical relationship between a content key and its holder's location, they require physical reachability between the source and its destination. Generally, routing protocols provide physical reachability separately from DHT

mechanisms. Thus, in ad hoc networks, to discover communication targets, applications have to find the locations of content holders and routes to them separately. Our paper addresses this issue when deciding design principles.

Caesar, et al.<sup>14)</sup> proposed a DHT-inspired routing protocol. In this protocol, each node uses a location independent identifier and maintains routes to logically adjacent nodes, that is, nodes where the identifiers are close to a source node. By using hash keys as location independent addresses, applications can find locations of content holders and routes to them at the same time. However, while logically adjacent nodes are not necessarily close to a source node, this might cause message overhead in resource-limited platforms such as mobile phone terminals. To prevent this message overhead, our implementation applies an on-demand type approach. This approach achieves the physically shortest routes to content holders.

Ratnasamy, et al.<sup>15)</sup> also proposed a DHT-inspired routing protocol for a data centric storage method in sensor networks. Under this method, the particular node that stores a given content is determined by the content's name. To realize this data centric storage, they proposed a Geographic Hash Table, which uses hashing a key into geographic coordinates and stores a key-value pair at the sensor node geographically nearest the hash of its key. To establish a route to the sensor node, they use geographical ad hoc routing protocols. However, to obtain and use geographical information, mobile nodes have to implement some sort of positioning system or database.

JXTA<sup>16)</sup> is an implementation framework for P2P applications and provides a set of open P2P protocols. A JAVA based JXTA implementation can be executed on mobile devices and realize P2P applications on those devices. In the JXTA framework, peers create a virtual, ad hoc network on top of existing networks. Besides our implementation, such P2P implementation framework on mobile terminal devices will contribute to the development of P2P applications suitable for ad hoc networks.

### 3. Design Principles

We investigate practical use cases for ad hoc communication and applications. From the viewpoint of the above use cases, we present three design principles; terminal device selection, discovery of communication targets and

protocol efficiency.

#### 3.1 Terminal Device Selection

We start by selecting a platform that is most suitable for mobile users to run ad hoc applications. In our design principles, terminal devices are required to meet the following criteria:

(1) Portability: To make the best use of mobile ad hoc applications, the terminal device has to be portable so that the user can run ad hoc applications anytime, anywhere.

(2) Availability: To establish ad hoc networks, there must be enough devices in the vicinity of the user.

(3) Security awareness: In ad hoc networks, there may be unidentified devices that are malicious to the user. Therefore, each device should be able to identify itself in a secure manner when requested to do so.

As for (1), when considering cases in which ad hoc applications are used on the road or on a train, a laptop PC is not portable enough. Handheld devices are required; such as a cellular phone or at the most a PDA. As for (2), to use ad hoc applications anytime, anywhere, PDAs are not widely used. Cellular phones or at least regular PCs including laptops are required. As for (3), a secure device authentication mechanism is required. To this end, a common PKI system or a trusted service provider as well as an access method is required.

To meet these criteria, a mobile phone terminal is one of the most suitable candidates. As ad hoc networks consist of mobile terminals without any servers, ensuring security and authentication is not taken for granted, which is raising concerns among mobile users. As mobile phone terminals originally have the access method to cellular networks, co-operation with cellular networks enables ad hoc networks to acquire security and authentication through the servers of trusted cellular providers. Co-operation with cellular networks also enables the use of existing Web applications. In addition, in regions where cellular networks are out of service, mobile terminals can connect to cellular networks by multi-hop links. Implementation of ad hoc networking capability on mobile phone terminals will therefore become a real driving force for ad hoc communications.

In contrast, the disadvantages of ad hoc networks on mobile phone terminals are resource limitations and implementation platforms.

(1) Resource limitations: Although mobile phone terminals have become smarter, they

have limited memory, CPU speed and power, as compared to laptop PCs or PDAs.

(2) Limitation of implementation platforms: While many mobile phone terminals provide programming platforms such as JAVA, only a few platforms can access devices such as wireless interfaces.

It is necessary to take into account these advantages and disadvantages in order to implement ad hoc networks on mobile phone terminals.

### 3.2 Discovery of Communication Targets

To communicate with a target node in ad hoc networks, each node establishes a route to the target node. While many route establishment methods are reported, they assume that the location of a communication target (e.g., IP address) is already known at the beginning of path discovery. However, in ad hoc networks, it is difficult to obtain the location of a communication target. Moreover, because it is difficult for many users to handle IP addresses directly, many Web applications use name information and a mapping mechanism between name and location. For example, Web browsers use a URI to connect to Web pages. While a URI is resolved to the IP address by the DNS server, no such server can be assumed to always be present in ad hoc networks. Therefore, it is still an open issue to discover the name and location of the communication target.

From the perspective of communication targets, there are two categories of applications. The first category includes applications that communicate with a specified peer. These applications need an identifier of the communication peer for communication. An example of these applications is Instant Messenger or Voice over IP (VoIP). Usually, servers manage the IDs and/or status of participants in such applications. However, because there is no management server in an ad hoc network, it is important to find communication targets in advance in order to execute applications.

The other category includes applications that communicate with unspecified peers. One example is a P2P file-sharing application. The goal of P2P file-sharing applications is not to communicate with specified peers, but to obtain desired information or files that are stored on unspecified peers. Therefore, it is important for such applications to discover target contents and establish routes to the content holders.

For these reasons, in our design principles, ad

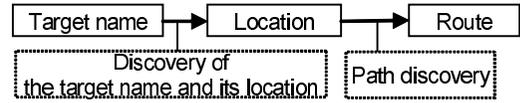


Fig. 1 Discovery of communication targets.

hoc communication implementation needs the means of discovering communication targets. Moreover, our implementation should take application discovery types into account.

### 3.3 Protocol Efficiency

While mobile terminal devices such as cellular phones have become smarter, they have fewer resources than the other terminals such as laptop or desktop PCs. Moreover, many ad hoc routing protocols use a flooding mechanism in the path discovery process. However, the flooding mechanism consumes wireless resources and is heavily dependent on link-layer technologies. Thus, in our design principles, more efficient protocols are required.

There are two categories of ad hoc routing protocols, on-demand routings and proactive routings. On-demand routing protocols start path discovery when communication requests occur and hold only active routes. In contrast, proactive routing protocols continuously search for routing information and maintain routes to all nodes. Generally, on-demand routing protocols need less memory resources than proactive routing protocols. Therefore, on-demand protocols are a suitable candidate when mobile terminal devices have limited memory resources.

As mentioned in Section 3.2, a means of discovering communication targets is required in our design principles. Discovery of the target name and its location is as important as path discovery. Therefore, protocol efficiency is required in these discoveries. As shown in **Fig. 1**, in many systems, discovery of routes is separately performed for discovery of the target name and its location. In wired networks where network bandwidth is abundant, separate discoveries do not influence communication performance. However, from the perspective of radio resource limitation on mobile terminal devices, separate discoveries would interfere with application performance. To solve this issue, joint discovery of the target name, its location and routes is an efficient approach on mobile terminal devices.

## 4. Implementation

This section shows our implementation of ad

**Table 1** Specifications of implementation platform.

|                                 |                                   |
|---------------------------------|-----------------------------------|
| Terminal type                   | W21T by Toshiba,<br>CDMA2000      |
| Bluetooth version               | 1.1                               |
| Supported<br>Bluetooth profiles | SPP, HSP, DUN,<br>BIP, OPPP, OBEX |
| BREW version                    | 2.1                               |
| Programming language            | C / C++                           |

hoc communication and ad hoc applications on practical device terminals according to the design principles proposed in Section 3.

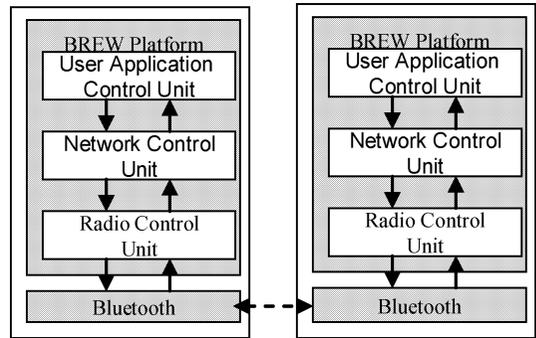
#### 4.1 Implementation Platform

According to the design principle of terminal device selection, mobile phone terminals were selected as implementation platforms. As for the link-layer protocol for ad hoc communications, several access control methods can be considered such as wireless LAN and Bluetooth. Among these technologies, we selected Bluetooth, which is relatively widely implemented on mobile phone terminals and stable in an infrastructureless network. The specifications of the mobile phone terminals are shown in **Table 1**. In our implementation, applicable profiles of Bluetooth are SPP, HSP, DUN, BIP, OPPP and OBEX. The Bluetooth profiles define the available functions. We used the Binary Runtime Environment for Wireless (BREW) for a programming platform. In the BREW platform, C and C++ are available as programming languages. One of the characteristics of BREW is the function of accessing devices such as address books and Bluetooth interfaces on mobile phone terminals.

On the BREW platform, we developed the new implementation architecture as shown in **Fig. 2**. The implementation architecture consists of three control units; a radio control unit that handles Bluetooth communication, a network control unit that establishes routes and forwards data packets and a user application control unit that includes a user interface and user applications. These control units are implemented as a single program file and the total program size is 315 Kbytes. The ratio of each control unit in the total program size is as follows: the radio control unit is 17%, the network control unit is 46% (route establishment is 26% and data forwarding is 20%), the user application unit is 26% and the other is 10%. The details of each control unit are shown in the following sections.

#### 4.2 Radio Control Unit

The radio control unit supports data trans-

**Fig. 2** Implementation architecture.

mission with adjacent terminals via the Bluetooth interface. The radio control unit carries out device discovery and broadcast/unicast of data transmissions.

The Bluetooth device has a 48-bit address that uniquely specifies each device. By using the Inquiry process, the device finds the device addresses of adjacent terminals. These device addresses are registered in a device address table, which is managed in each mobile phone terminal. The device address table manages only the active device addresses. If link failure or transmission failure occurs, the corresponding device address is deleted from the table.

The Bluetooth specification defines Piconets or Scatternets to realize ad hoc networks. However, our platform does not support Piconets and Scatternets. We therefore used the Serial Port Profile (SPP) for data transmission.

The SPP allows Bluetooth devices to perform serial cable emulation. The SPP has two transmission modes, a client mode and a server mode. In the SPP connection, data is transmitted from the client mode terminal to the server mode terminal. In our implementation, unicast uses this SPP transmission. When two terminals connect to one terminal simultaneously, connection collision occurs. In this case, one terminal succeeds in connection and the other fails. The failed terminal retries the connection. The Bluetooth interface caches information about the device address and radio frequency pairs when SPP connection is established. At the second time or after, the Bluetooth interface establishes the connection more quickly by using the information.

In contrast, SPP supports only point-to-point connection and does not support broadcast transmission. From this limitation of platform available function, a pseudo broadcast mech-

anism was implemented. A pseudo broadcast transmission is realized by repeating a unicast SPP transmission. When a broadcast transmission is required, a terminal selects one entry from its device address table in the registered order and then it unicasts to the selected entry. Until the terminal finishes transmissions to all entries in its device address table, it repeats unicast transmissions.

**4.3 Network Control Unit**

The network control unit realizes packet forwarding and route establishment.

(1) Packet forwarding

IP and transport layer protocol stacks were used to transmit messages. While these protocol stacks are already prepared as part of the Application Programming Interface (API) in the BREW platform, this API is not available for Bluetooth communication. Therefore, we implemented the new IP and transport layer stacks required for Bluetooth communication, route establishment and applications. The new IP stack supports IP packet format construction and packet forwarding. UDP was implemented as the new transport layer stacks. This UDP stack transmits control packets for route establishment and data packets of applications.

(2) Route establishment

To establish routes, we implemented the on-demand ad hoc routing protocol, AODV, based on the design principle of protocol efficiency taking memory resource limitation on mobile phone terminals into account. AODV carries out the following path discovery process.

A source terminal begins flooding a Route Request (RREQ) when a communication request occurs and the terminal does not have the route to the destination. As described in Section 4.2, flooding is realized by the pseudo broadcast mechanism. Terminals that receive the RREQ register a reverse route to the source terminal. If the terminal that receives the RREQ is not the destination, the RREQ is re-broadcast. When the destination terminal receives the RREQ, it unicasts a Route Reply (RREP) to the source terminal. When terminals receive the RREP, they make the forward route to the destination. After the source terminal receives the RREP, it begins data transmission of the user application.

(3) Extension of path discovery

The path discovery process of AODV achieves only route establishment. According to the design principle shown in Section 3.2, discovery

of the target name and its location is required. Moreover, the design principle explained in Section 3.3 needs protocol efficiency when discovering the target name, its location and its route. We extended the path discovery process that uses a content ID as a target name and realized joint discovery of the target name, its location and its route.

This extension uses a content ID, not the IP addresses, as the identifiers of communication targets. The meaning of content ID is different in each application. In applications where the target is a specified peer, a content ID presents a target node ID. For example, VoIP applications use a target user name for VoIP service or an E-mail address. When communication targets of applications are unspecified peers, e.g., P2P file-sharing applications, a content ID means a target file name. That is, content IDs should represent unique target names from the viewpoint of application service. In this extension, each node maintains available content and their content IDs in its content ID table. For simplicity and efficiency of protocol implementation, each terminal maintains only content IDs that it holds by itself.

The following extended path discovery process was implemented. A source terminal broadcasts an extended RREQ when a communication request occurs. When requesting communication, applications specify the desired content ID. The extended RREQ has the content ID and the IP address of the source terminal. Note that the extended RREQ does not include the IP address of the destination terminal because the source terminal does not know the location (IP address) of the destination terminal. The extended RREQ has an option format field as shown in **Fig. 3**. The option data format is composed as a Type-Length-Value (TLV) format and loads the desired content ID.

When adjacent terminals receive the extended RREQ, the reverse route to the source terminal is created. When the receiving terminals do not have the desired content ID in their content ID tables, they re-broadcast it. Otherwise, the terminal unicasts an extended RREP to the source terminal. The extended RREP has the same

|                       |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |
|-----------------------|--|--|--|--|--|--|--|--|--|---------------------|--|--|--|--|--|--|--|--|--|---------------------|--|--|--|--|--|--|--|--|--|---------------------|--|--|--|--|--|--|--|--|--|
| 0                     |  |  |  |  |  |  |  |  |  | 1                   |  |  |  |  |  |  |  |  |  | 2                   |  |  |  |  |  |  |  |  |  | 3                   |  |  |  |  |  |  |  |  |  |
| 0 1 2 3 4 5 6 7 8 9 0 |  |  |  |  |  |  |  |  |  | 1 2 3 4 5 6 7 8 9 0 |  |  |  |  |  |  |  |  |  | 1 2 3 4 5 6 7 8 9 0 |  |  |  |  |  |  |  |  |  | 1 2 3 4 5 6 7 8 9 0 |  |  |  |  |  |  |  |  |  |
| Type                  |  |  |  |  |  |  |  |  |  | Length              |  |  |  |  |  |  |  |  |  | Value...            |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |
| Value... (cont.)      |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |

**Fig. 3** Option format for content ID.

option format as the extended RREQ. The extended RREP loads the desired content and the IP address of the content holder terminal. When the extended RREP is received, a terminal makes the forward route to the content holder terminal. When receiving the extended RREP, the source terminal obtains the routes to the content holder terminal and the desired content. Sample sequences of the extended path discovery process are shown in **Fig. 4** (request phase) and **Fig. 5** (reply phase). Terminals detect path discovery failure when they do not receive an RREP message in a certain period. When path discovery failure occurs, ter-

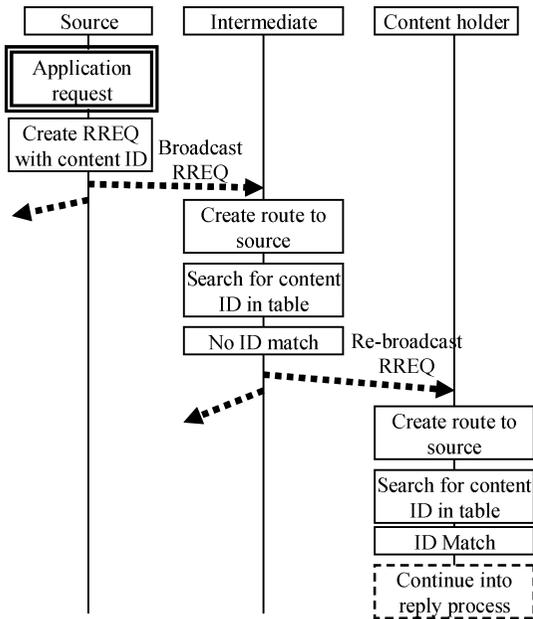
minals retry the path discovery. In our implementation, the period is set at 60 seconds.

**4.4 User Application Unit**

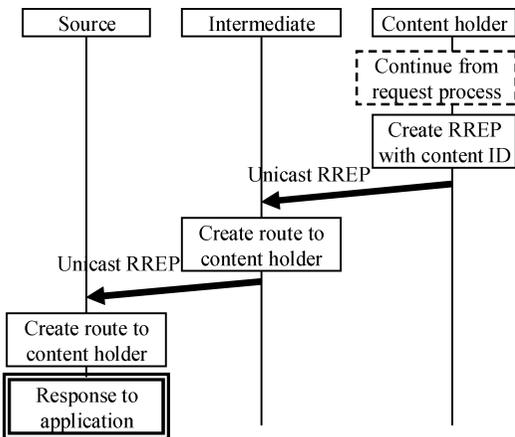
Based on the consideration in Section 3.2, we implemented two applications that have different types of communication targets, respectively. One includes applications whose communication targets are specified peers. We implemented a chat application because it does not require real-time communications. The other includes applications that communicate with unspecified peers. We implemented a file-sharing application. These two applications are P2P type applications, that is, they do not need a management server and communicate with each other directly. **Figures 6 and 7** show screen images of these two applications.

(1) Chat application

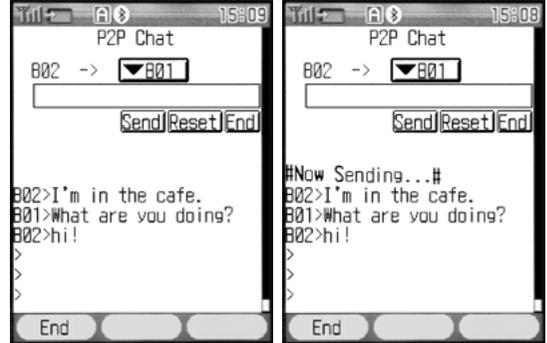
A chat application exchanges short text messages in a peer-to-peer manner. In the infrastructureless situation, the user can send a short message to another user such as friends. We assume that users know lists of target user names in advance, e.g., they obtain the list from address books on mobile phone terminals. However, even when target user names are known,



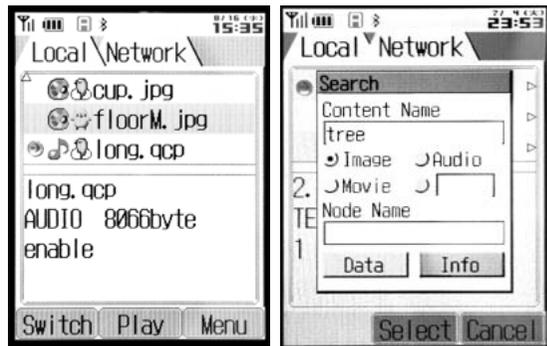
**Fig. 4** Extended path discovery (request phase).



**Fig. 5** Extended path discovery (reply phase).



**Fig. 6** Screen images of the chat application.



**Fig. 7** Screen images of the file-sharing application.

it is not easy to obtain the IP addresses of target users in ad hoc networks. To find targets and make a route to them, we used the extended path discovery shown in Section 4.3. Here, the content ID is the target user name. The extended path discovery achieves discovery of the target user and route establishment jointly. After the extended path discovery, a chat data message is transmitted via the established routes. We assume that the data size of chat messages is about 100 Bytes.

(2) File-sharing application

A file-sharing application exchanges content files such as pictures or movies with other users. This application loads pictures and movies from a data folder of a mobile phone. Each content file has a unique content ID by using the hash function. When a user inputs the content file name that he wants to obtain, this application searches for it on the ad hoc network and retrieves it. To find a content file and establish a route to the content file holder, the extended path discovery is used. From the extended path discovery, the user can obtain content files from content holders. For simplicity of implementation and evaluation, this application does not use file cache, that is, only the content holder can reply to a request for file-sharing.

5. Performance Evaluation

We evaluated the performance of our implementations. Through the performance evaluations, we discuss performance characteristics of our implementation and potential application areas. In the following experiments, each result is averaged over 20 trials.

5.1 Preliminary Evaluation

We present the preliminary evaluation that focuses on the Bluetooth performance. We show two evaluations; Bluetooth device discovery time and 1-hop data transmission time.

(1) Bluetooth device discovery

Table 2 shows the average time of device discovery with the number of adjacent terminals. The average time is linear to the number of adjacent nodes. We confirmed from this result that about 1 second is necessary to find one adjacent terminal.

(2) 1-hop data transmission time

Figure 8 shows the average 1-hop data transmission time. When data size is smaller than or equal to 10 Kbytes, 1-hop data transmission time is about 2 seconds. The specification of Bluetooth version 1.1 originally supports up to

Table 2 Device discovery time.

| No. of adjacent terminals | 1   | 2   | 3   | 4   |
|---------------------------|-----|-----|-----|-----|
| Average time (second)     | 1.1 | 2.1 | 2.9 | 4.3 |

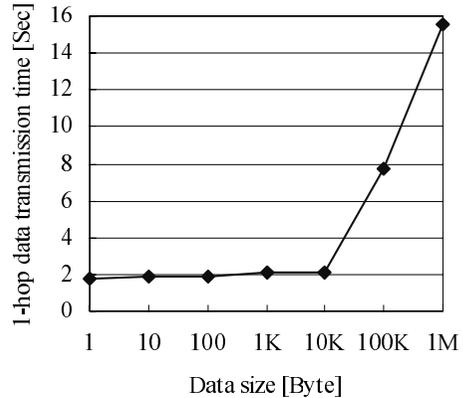


Fig. 8 1-hop data transmission time.

721 Kbps effective data rate transmission. However, the result is significantly different from such data rate. We examined this difference and confirmed that this transmission time is mainly dependent on SPP connection establishment. Moreover, the data transmission time increases when the data size is over 10 Kbytes. This is caused by two factors. One is data fragmentation in SPP data transmission. While small data is forwarded by one transmission, large data requires several transmissions after being segmented to a certain length. The other is the increase in the file access speed. When handled data size is large, more access time is required to execute file input and output because of low memory and processing speed.

5.2 Path Discovery Evaluation

This section evaluates the path discovery performance. To examine the effect of multi-hop transmission and the pseudo broadcast, stability of network topologies is required. To construct the experimental network topology, the device address table presented in Section 4.2 was used. The terminals registered in that table mean to be adjacent to the target terminal and only the links to those terminals are validated.

(1) Multi-hop transmission evaluation

We measured the path discovery period in the line network topology shown in Fig. 9. This experiment used ten terminals and changed the hop count to a destination terminal from 1 to 9. Figure 10 shows the path discovery period in the line network topology. As the hop count to a destination increases, the path discovery pe-

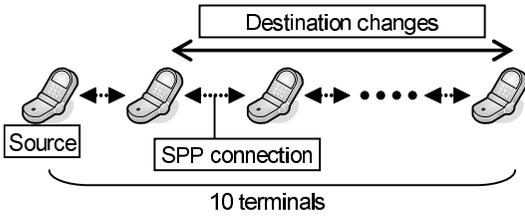


Fig. 9 The single neighbor multi-hop network (the line network topology).

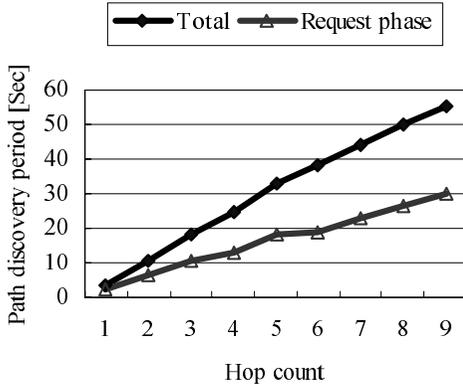


Fig. 10 Path discovery period in the line network topology.

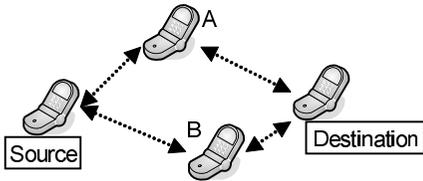


Fig. 11 The multiple neighbor multi-hop network (2 neighbors).

riod increases linearly. The transmission time per hop is 2 or 3 seconds. This is slightly longer than that of 1-hop data transmission shown in Fig. 8. This is because SPP connection overhead is longer in multi-hop transmission than in 1-hop transmission. From Fig. 10, the time in the request phase is slightly more than that in the reply phase. In the reply phase, terminals establish SPP connections quickly by using the cache information of the device address and radio frequency pairs as presented in Section 4.2.

(2) Multiple neighbors evaluation  
 Here, we examine the other network topology where there are multiple neighbor terminals. We experiment with the two network topologies shown in Fig. 11 (2 neighbors) and Fig. 12 (3 neighbors). Figure 13 shows the path discovery period in Fig. 11 and Fig. 14 shows that in Fig. 12, respectively. The average path discov-

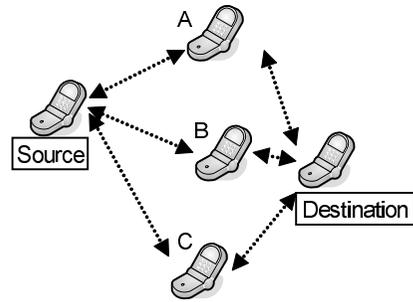


Fig. 12 The multiple neighbor multi-hop network (3 neighbors).

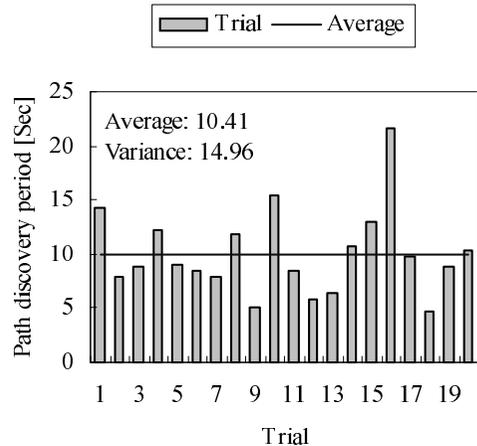


Fig. 13 Path discovery period in the network topology shown in Fig. 11.

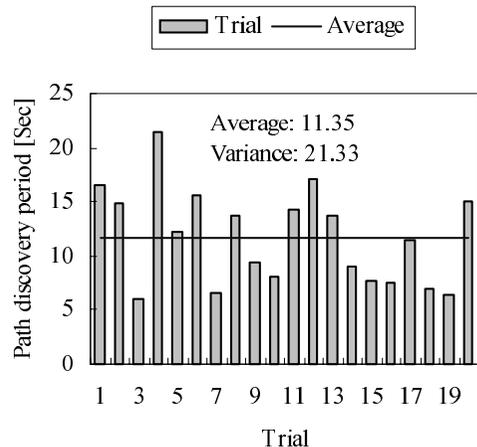


Fig. 14 Path discovery period in the network topology shown in Fig. 12.

ery period in these network topologies is nearly equal to that in Fig. 10 with 2-hop count. This result shows that the path discovery period is mainly dependent on the hop count. On each trial in Figs. 13 and 14, the path dis-

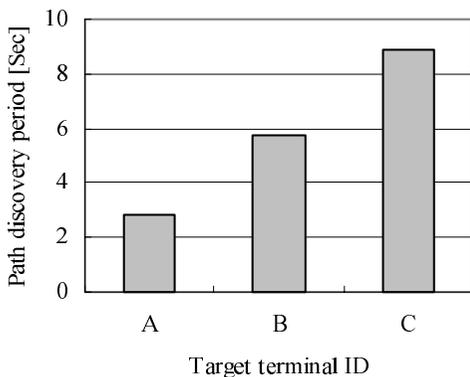
covery period varies widely. Moreover, when comparing between Fig. 13 and Fig. 14, the average and variation of path discovery period in Fig. 14 are larger than those in Fig. 13. The wide variation on each trial and the difference between two network topologies are caused by the same factor, collision occurrence. When there are multiple neighbor terminals, collision sometimes occurs. As the number of adjacent terminals increases, collision occurs more frequently.

(3) Pseudo broadcast evaluation

We examine the effect of the pseudo broadcast presented in Section 4.2. In this experiment, we set the destination terminals as Terminals A, B and C, respectively and we register the device address tables in the alphabetical order. **Figure 15** shows the path discovery period in this experiment. The path discovery period increases in the registered order, i.e., the connection order in the pseudo broadcast. From this result, the order in the pseudo broadcast affects the path discovery performance.

(4) Comparison between the baseline AODV and the extended path discovery

If the content discovery is conducted independently, since the route between the content seeker and content holder is not maintained in the intermediate nodes, both destination finding and returning processes are performed in a flooding way. After the content seeker (i.e., the source) receives the information on the content holder (i.e., the destination), the source establishes the route to the destination. The time required for this series of processes is roughly twice as long as that of our proposed method. Given that the information on the content holder is cached in the intermediate nodes or is stored in a certain centralized server, the



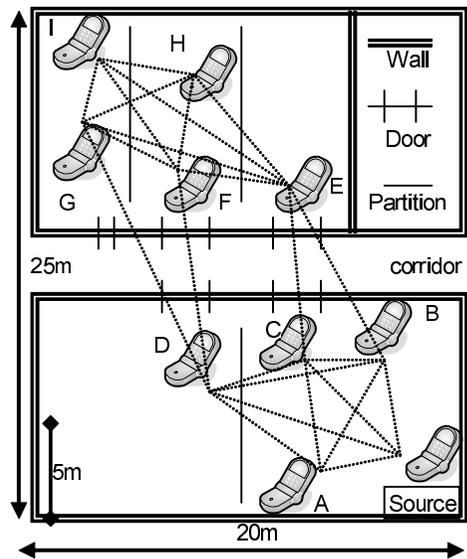
**Fig. 15** Path discovery period in the pseudo broadcast evaluation.

content seeker needs to reach such a node or server, which may be closer than the destination; however, the content seeker still needs to establish the route to the content holder. This process may be shorter than the aforementioned case, but would still take longer than our proposed method.

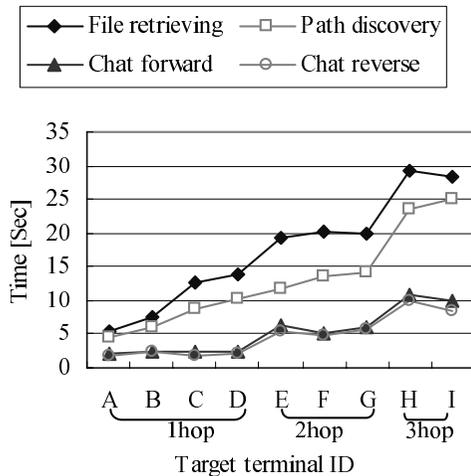
**5.3 Application Performance Evaluation**

We examine the performance of the chat and file-sharing applications. To evaluate the performance in a practical situation, terminals are located in the 25 m by 20 m experimental space shown in **Fig. 16**. The experimental space is composed of two rooms separated by a corridor and there are walls, partitions and doors between terminals. At the beginning of the experiment, each terminal executes device discovery and then the network topology shown in Fig. 16 is constructed. In this experiment, each terminal is adjacent to about four to six terminals. Because of the radio shielding effect of the walls, Terminals C and F are not topologically adjacent while they are physically close.

This experiment focuses on the communication between the source and the other terminals. Terminals are divided into three groups from the point of the shortest hop count from the source. Terminals A, B, C and D are 1-hop away, Terminals E, F and G are 2-hop away and Terminals H and I are 3-hop away from the source terminal.



**Fig. 16** The experimental space and the ten nodes network topology.



**Fig. 17** Performance in the network topology shown in Fig. 16.

In the chat application, the source begins the extended path discovery to find destination terminals. After path discovery, chat messages are transmitted to each other. We use the chat message where the data size is 100 bytes. **Figure 17** shows the path discovery period, data transmission time from the source to the destination (forward) and data transmission time from the destination to the source (reverse). There are differences in the path discovery period of Terminals A, B, C and D, which are 1-hop neighbors. In this experiment, we register the device addresses in alphabetical order. Therefore, these differences are caused by the transmission order in the pseudo broadcast. Looking at the results of the other terminals, the path discovery period increases when the number of hops is large. Therefore, we confirm that, in this practical situation, the path discovery performance depends on the hop count and the transmission order in the pseudo broadcast.

The time in chat message transmission is also linear to the hop count and is shorter than that in path discovery. This is because a chat message is transmitted by unicast along the established route. Moreover, the cache information presented in Section 4.2 contributes to this small transmission time. From the result, we confirm that there are few differences between forward and reverse transmission.

The file-sharing application discovers the target content and retrieves it in one transaction by the extended path discovery. Figure 17 shows the time for this process. The content file is a mobile phone screen (240 by 320 dots)

sized picture and 20 Kbyte data size. The total time of file retrieving is larger than the path discovery period. The difference between them is due to data size. Thus, if the data size becomes larger, the total time of file retrieving increases due to the file accessing speed as presented in Fig. 8.

The above evaluation revealed that networking performance is lower than that of other mobile environments such as laptop PCs with wireless LAN. This is mainly due to the resource limitations of mobile phones and its application platform. In high-mobility situations, performance degradation occurs due to the long delay in data transmission. However, our implementation will be more suitable in low-mobility situations, for example, ad hoc communication at meeting spots or event sites. Moreover, this long delay is not suitable for real-time applications such as VoIP or streaming applications. In contrast, we can execute non real-time applications with a small data size, e.g., chat or file-sharing applications. In addition, our extended path discovery, which achieves joint discovery of name, location and route, is effective by reducing the amount of costly message passing.

## 6. Conclusion

This paper investigated practical use cases for mobile ad hoc communications and clarifies three design principles; terminal device selection, discovery of communication targets and protocol efficiency. The requirement for terminal device selection was shown based on the above use cases and from this design principle, it was concluded that a mobile phone terminal is the best suitable candidate. From the second design principle of discovering communication targets, we showed that ad hoc communication requires a discovery method of the target name, its location and its route. Moreover, taking the resource limitations of mobile terminal devices into account, protocol efficiency should be required in the third design principle.

Based on the design principles, a mobile phone terminal was selected as the implementation platform and ad hoc communication and applications were implemented on those devices. The path discovery method was extended to jointly discover the target name, location and route, thereby reducing the procedure time. We also implemented applications, chat and file-sharing applications, which are suitable candidates for ad hoc communication.

We further measured the performance of our implementation. Through the performance evaluation, we revealed the characteristics of ad hoc communication and applications on our implementation. We clarified the potential application area for ad hoc networks.

### References

- 1) Bluetooth SIG: Specification of the Bluetooth system. <http://www.bluetooth.com/>
- 2) Perkins, C., Royer, E. and Das, S.: Ad hoc On-demand Distance Vector (AODV) Routing, RFC3561, IETF (2003).
- 3) Clausen, T. and Jacquet, P.: Optimized Link State Routing Protocol (OLSR), RFC3626, IETF (2003).
- 4) BREW. <http://brew.qualcomm.com/brew/>
- 5) AODV-UU. <http://core.it.uu.se/core/index.php/AODV-UU/>
- 6) Kernel AODV. [http://w3.antd.nist.gov/wctg/aodv\\_kernel/](http://w3.antd.nist.gov/wctg/aodv_kernel/)
- 7) OLSR. <http://menetou.inria.fr/OOLSR/>
- 8) Unik olsr. <http://www.olsr.org/>
- 9) Haartsen, J.: Bluetooth — Ad-hoc Networking in an Uncoordinated Environment, *ICASP Proceedings*, pp.2029–2032, IEEE (2001).
- 10) Jian, Z., Fei, Z. and Bin, W.: Cluster Based Routing in Bluetooth Ad-hoc Network, *WCNM Proceedings*, pp.742–752, IEEE (2005).
- 11) Ronai, M. and Kail, E.: A Simple Neighbor Discovery Procedure for Bluetooth Ad Hoc Networks, *GLOBECOM Proceedings*, pp.1028–1032, IEEE (2003).
- 12) Formo, F., Malnati, G. and Portelli, G.: Design and Implementation of a Bluetooth Ad Hoc Network for Indoor Positioning, *IP-SEN Proceedings*, pp.223–228, IEE (2005).
- 13) Stoica, I., Morris, R., Karger, D., Kaashoek, M. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for Internet applications, *SIGCOMM Proceedings*, pp.149–160, ACM (2001).
- 14) Caesar, M., Castro, M., Nightingale, E., O’Shea, G. and Rowstron, A.: Virtual Ring Routing: Network Routing Inspired by DHTs, *SIGCOMM Proceedings*, pp.351–362, ACM (2006).
- 15) Ratnasamy, S., Karp, B., Yin, L., Yu, F., Estrin, D., Govindan, R. and Shenker, S.: GHT: A Geographic Hash Table for Data-Centric Storage, *WSNA Proceedings*, pp.78–87, ACM (2002).
- 16) JXTA. <http://www.jxta.org/>

(Received November 2, 2006)

(Accepted April 6, 2007)

(Online version of this article can be found in the IPSJ Digital Courier, Vol.3, pp.480–491.)



**Yujin Noishiki** received his B.E. and M.Informatics degrees from Kyoto University, Japan, in 2000 and 2002, respectively. He joined KDDI R&D Laboratories, Japan, in 2002. His current research interests include mobile computing and ad hoc networks. He is a member of IEICE.



**Hidetoshi Yokota** received his B.E., M.E. and Ph.D. degrees from Waseda University, Tokyo, in 1990, 1992, and 2003, respectively. He joined KDDI R&D Laboratories, Japan, in 1992. From 1995 to 1996 he was with SRI International in Menlo Park, CA as an International Fellow. He received the IEICE Young Engineer Award in 1998 and the IPSJ Yamashita SIG Research Award in 2005. His current research interests include mobile communications and ad hoc networks. He is a member of IPSJ, IEICE and IEEE.



**Akira Idoue** received his B.E. and M.E. degrees from Kobe University in 1984 and 1986, and his Ph.D. degree from the University of Electro-Communications in 2007, respectively. Since joining KDD (now KDDI) in 1986, he has worked in the field of network architecture, communication protocols, protocol testing and mobile communications. He is currently a senior manager of the Ubiquitous Networking Lab. at KDDI R&D Laboratories. He received the IPSJ Convention Award in 1993 and the Best Paper Award of the IPSJ National Convention in 1998. He is a member of IPSJ and IEICE.