Design and Implementation of Ad-hoc Communication and Application on Mobile Phone Terminals

Yujin Noishiki Hidetoshi Yokota Akira Idoue

KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino-Shi, Saitama, 356-0003, Japan, yujin@kddilabs.jp

ABSTRACT

Recent advancements 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. To expedite the widespread use of ad-hoc communications, we turn our attention to the handiest wireless device, namely, a mobile phone terminal. 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 almost anytime and anywhere, which we believe will become a driving force for ad-hoc networking technology. On the other hand, while the performance of mobile phones has improved greatly over the past several years, their resources are still limited compared to laptop PCs. Therefore, to realize ad-hoc communication in mobile phones, this limitation should be considered. In this paper, we first set the design principles for mobile ad-hoc communications based on the assumed use cases and implement ad-hoc networking protocols as well as their applications on mobile phone terminals to verify their effectiveness. Through the performance evaluations, we discuss performance characteristics and potential application areas.

Keywords: Mobile Ad-hoc Networks, Mobile Computing, Implementation, Bluetooth

1 INTRODUCTION

Rapid deployment of Wireless PAN technologies such as Bluetooth [1] or ZigBee is expected to enable mobile users and devices in their vicinity to more easily access and exchange information. Their 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. One such issue is the slow deployment of ad-hoc networking capabilities on mobile terminals. Many ad-hoc routing protocols have been proposed under the initiative of the Internet Engineering Task Force (IETF) Mobile Ad-hoc Network Working Group (MANET WG) [2]. Ad-hoc On-demand Distance Vector (AODV) routing [3] and Optimized Link State Routing (OLSR) [4] are standardized as experimental RFCs in IETF. While many studies examine their proposed method by computer simulation, there are few reports about the evaluation of actual machine implementations. Evaluations in a real device environment are important for consideration of the actual user operations. Applications for ad-hoc networks should also be examined on actual machine implementations. In particular, as each application has unique characteristics, the evaluation of applications on actual machine implementations is as important as that of routing protocols from the viewpoint of users.

Proposing attractive applications for ad-hoc communications is also a crucial issue in order to make ad-hoc communication widespread. In mobile ad-hoc networks, where there is no infrastructure, Peer-to-Peer (P2P) applications such as file sharing or messaging are suitable because P2P communication are conducted without any central servers. Furthermore, concurrent connection support to infrastructure networks makes conventional server-oriented applications, e.g. Web services, easy to use on ad-hoc networks. It is becoming common for mobile terminals to have multiple wireless interfaces. Besides laptop PCs and PDAs, several 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 on mobile phones. As mobile phones 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.

In this paper, we design and implement ad-hoc communication functions and ad-hoc communication applications on mobile phone terminals. From the perspective of design, resources for software programs and radio resource control are limited compared to PCs, even though the performance of mobile phone terminals has improved remarkably. Therefore, we need to implement ad-hoc communications and applications in consideration of this limitation. From the perspective of implementation, we use Bluetooth as the short-rage wireless interface. Ad-hoc communications and applications are implemented on the BREW platform [5]. The implementation stack is 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. The user application control unit manages P2P chat communication and file exchange applications.

The rest of this paper is organized as follows. In Section 2, we describe related work. We focus on the implementation of ad hoc networks. In Section 3, we show the design principles. Section 4 explains the implementation of ad-hoc networks on mobile phone terminals and Section 5 evaluates the implementation. Finally, Section 6 concludes this paper.

2 RELATED WORK

Implementation of ad-hoc routing protocols is reported from some research groups. AODV, one of the on-demand ad-hoc routing protocols, is implemented on actual machines [6][7]. Both implementations support the Linux-based platform and [6] implements AODV on PDAs that have ARM-based architecture. OLSR is a proactive type ad-hoc routing protocol and is also implemented on Linux, Windows and the ARM architecture [8][9]. The implementations of AODV and OLSR use IEEE 802.11 Wireless LAN as the short-range wireless interface. In the IEEE 802.11 standard, ad-hoc mode is used for peer-to-peer communication without infrastructure.

Bluetooth is another short-range wireless interface. Bluetooth uses the Industrial Scientific and Medical (ISM) band, which can be freely used in most countries. 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 [10]. Scatternet is defined as a group of independent and non-synchronized Piconets. Communication of two Piconets is via gateway terminals that belong to the two Piconets. Paper [10] reports that Piconets and Scatternets increase the efficiency of channel use by the Frequency Hop spread spectrum as compared to Wireless LAN that shares frequency in the same radio propagation area. The Piconet and the Scatternet are special types of ad-hoc networks.

The cluster-based ad-hoc routing protocol is proposed as the routing protocol in Piconets and Scatternets [11]. In cluster-based ad-hoc routing protocols, the terminal called the cluster head controls route establishment and traffic in the cluster. As Scatternet consists of multiple Piconets and data transmission in Piconets is controlled 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 is limited to only some of the terminals, such as the cluster head or the master terminal in a Piconet. When terminals have limited wireless resources and power, it is necessary to examine routing protocols in which all terminals consume available resources equally.

In Bluetooth networks, the Inquery process is defined for each terminal to find adjacent terminals. As the Inquery process requires several seconds to find adjacent terminals, there is a problem with applying the Inquery process to ad-hoc networks. Paper [12] proposes a simple adjacent terminal discovery process. The proposed method improves the discovery time by simplifying the Inquery process.

Thus, it differs from ad-hoc networks over Wireless LAN and contains many of the features of Bluetooth ad-hoc networks. We need this difference and the features to implement Bluetooth ad-hoc networks. Bluetooth ad-hoc networks are evaluated by computer simulation in many studies. However, some implementations on actual machines are reported from some research groups. Paper [13] reports the design and implementation of indoor positioning over Bluetooth ad-hoc networks. The proposed system measures the positioning of devices by Bluetooth and transmits positioning results via Bluetooth ad-hoc networks. The proposed system is implemented on laptop PCs.

While many studies report implementation on PCs or PDAs, the implementation of ad-hoc networks on mobile phone terminals has not been reported. As mobile phone terminals have limited memory resources and power compared to note PCs or PDAs, it is necessary to take into account of limitation for implementation.

3 DESIGN PRINCIPLES

3.1 Terminal Device Selection

In this section, we start by selecting a platform that is most suitable for mobile users to run ad-hoc applications. We consider 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 and anywhere.
(2) Availability: to establish ad-hoc networks, there must be enough devices in the vicinity of the user.
(3) Security awareness: in the ad-hoc network, unidentified devices can exist that may be malicious to the user. Therefore, each device should be able to identify itself when requested to do so in a secure manner.

As for (1), when considering cases where 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 and 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 can be considered as one of the most suitable candidates and it will therefore become a real driving force for ad-hoc communications. As ad-hoc networks consist of mobile terminals without any servers, ensuring the functions of security and authentication is not taken for granted, which is also raising concerns among mobile users. As mobile phone terminals originally communicate via cellular networks, co-operation with those cellular networks enables ad-hoc networks to have these functions through the servers of trusted cellular providers. Co-operation with cellular networks also enables the use of existing Web applications. Moreover, in regions where cellular networks are out of service, ad-hoc networks can connect to cellular networks by multi-hop links.

On the other hand, the disadvantages of ad-hoc networks on mobile phone terminals are (1) limitation of resources and (2) limitation of implementation platforms. (1) Limitation of resources: Although mobile phone terminals have become smarter, they have limited resources, 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 Target

Many studies that research ad-hoc routing assume the identifier of the communication target (e.g., IP address) is already known when the communication request occurs. Moreover, because it is difficult for many users to handle IP addresses directly, many Web applications use name information on the communication target. 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 obtain the identifier of the communication target.

Our design principle for ad-hoc applications defines two types of applications base on their discovery methods. One type is those applications which communicate with a specific target. An example of this type of application is an Instant Messenger. Usually, servers manage the IDs and/or status of participants and data transmission. In some P2P types of Instant messenger, servers manage only participants, while data transmission is handled via P2P links. 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 such applications. As mobile phone terminals acquire information on communication targets, such as phone numbers and E-mail addresses in most cases, use of this information is one approach. The other type of application is those that communicate with unspecified targets. One example of this is a file sharing application. The goal of file sharing applications is not to communicate with specified targets but to obtain desired information or files that are stored on unspecified targets. Therefore, what is important is not to specify communication targets, but to map between the target information and the terminals that hold it. P2P type file sharing applications often use the Distributed Hash Table (DHT) to construct a system that efficiently manages the terminals that maintain information and the information itself. However, while DHT improves the manners in which to locate the host holding the requested information, it does not guarantee that the route to the host has been established. Therefore, in ad-hoc networks on mobile phone terminals where resources are limited, management should operate jointly with the establishment of the route.

3.3 Protocol Efficiency

Ad-hoc networks are constructed only by mobile terminals without any infrastructure. Mobile terminals can move freely in ad-hoc networks. Thus, it is necessary to develop quick and efficient procedures to establish routes, find a communication target and transmit data. From this perspective, on-demand ad-hoc routing protocols are suitable for implementation on mobile phone terminals.

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, a more effective and suitable flooding mechanism is required for mobile phone terminals and their communication interfaces. Moreover, in standardized ad-hoc routing protocols, route establishment and data transmission are separately executed. However, as mobile phone terminals have limited resources, co-operation between route establishment and data transmission is desirable.

4 IMPLEMENTATION

According to the design principles denoted in Section 3, this section shows our implementation of ad-hoc communication and ad-hoc applications to mobile phone terminals..

4.1 Implementation Platform

As for the link-layer protocol for ad-hoc communications, several access control methods can be considered such as

Wireless LAN, Bluetooth or ZigBee. Among these technologies, we selected Bluetooth for our implementation. Bluetooth is relatively widely implemented on mobile phone terminals and stable in an infrastructureless network. The specifications of the mobile phone terminals that were used are shown in Table 1. Bluetooth defines profiles that provide the available functions. In our implementation, applicable profiles of Bluetooth are SPP, HSP, DUN, BIP, OPPP and OBEX. The implementation platform use was the Binary Runtime Environment for Wireless (BREW). BREW is a programming platform for mobile phone terminals, for which C and C++ are available as programming languages. One of the characteristics of the BREW platform is the function of accessing devices such as address books and Bluetooth interfaces on mobile phone terminals. The version of BREW used in our implementation was 2.1.

Table 1: Specifications of implementation platform

Mobile phone terminals	Toshiba W21T
Bluetooth version	1.1
Applicable profile	SPP, HSP, DUN, BIP, OPPP, OBEX
BREW version	2.1
BREW data folder size	5Mbytes
Programming language	C / C++

4.2 Implementation Stack

On the BREW platform, we developed a new implementation stack as shown in Figure 1. The implementation stack 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 the user interface and user applications.

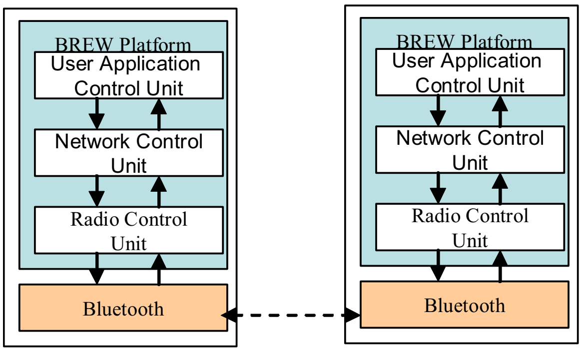

Figure 1: Implementation stack.

As shown in Figure 2, the control sequence has three phases, the initial phase, the route establishment phase and the user application execution phase. In the initial phase, the radio control unit carries out device discovery that finds adjacent terminals. In the route establishment phase, when a communication request occurs in the user application control unit, the network control unit begins the path discovery process. In the path discovery process, a control message is broadcast or unicast by the radio control unit. After establishing routes, in the user application execution phase, user data is forwarded via routes obtained in the route establishment phase. In the user application execution phase, the radio control unit transmits user data by unicasting. The details of each of the control units are shown in the following sections.

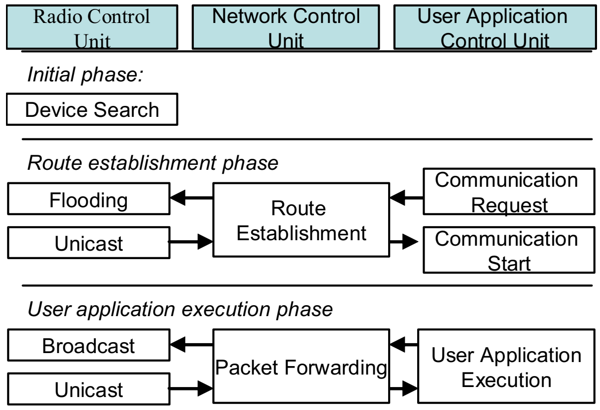

Figure 2: Control sequence.

4.3 Radio Control Unit

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

(1) Device discovery:

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

(2) Unicast and Broadcast:

Bluetooth can realize ad-hoc networks through Piconets or Scatternets. However, only the six Bluetooth profiles shown in Table 1 are available in our implementation. So, our platform does not support Piconets and Scatternets. Then, we use the Serial Port Profile (SPP) to transmit data to adjacent terminals.

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 unicast transmission, mobile phone terminals set the destination device address to the next hop terminal referenced from the device address table. On the other hand, SPP does not support broadcast transmission. Then, the broadcast transmission is realized in a pseudo manner by unicast SPP transmission. When the broadcast transmission is required, for example, at the beginning of the path discovery process, the mobile phone terminal connects to an adjacent terminal selected from the entries in the device address table. After this unicast transmission, the sending terminal connects to another terminal by SPP until it finishes transmission to all the terminals listed in the device address table. Thus, SPP connections are repeated for each of addresses registered in the device address table.

4.4 Network Control Unit

The network control unit realizes packet forwarding and route establishment.

(1) Packet forwarding:

To transmit control messages and data messages, we use IP and transport layer protocol stacks. While IP and transport layer 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. So, we implemented new stacks customized for Bluetooth communication. Each terminal has a routing table to determine the next hop terminal. The routing table is constructed by the route establishment method.

(2) Route establishment:

To establish routes, we implemented the on-demand ad-hoc routing protocol, AODV, according to the considerations in Section 3.3. AODV carries out the following path discovery process.

A source terminal begins flooding a Route Request (RREQ) when a communication request from a user application occurs and the terminal does not have the route to the destination. The RREQ includes the IP address of the destination terminal and the source terminal. Terminals that receive the RREQ register a reverse route to the source terminal in their routing table. When the destination terminal receives the RREQ, it unicasts a Route Reply (RREP) to the source terminal. The RREP provides the information on the destination terminal. If the terminal that receives the RREQ is not the destination, the RREQ is re-broadcast. 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:

According to the considerations in Section 3.2 and Section 3.3, we extended the path discovery process from the view of discovery of unspecified communication targets and protocol efficiency.

This extension uses an index of content or service, not the IP addresses, as the identifiers of communication targets. The index identifies content or services uniquely. Content and services that a terminal can provide are listed in the terminal's content information table. For simplicity and efficiency of protocol implementation, each terminal maintains the information on content that the terminal holds in our implementation.

The path discovery process with the above extension is as follows. A source terminal broadcasts an extended RREQ when a communication request occurs from a user application. The user application specifies the desired content index. The extended RREQ has the content index and the IP address of the source terminal. Note that the extended RREQ does not include the IP address of the destination terminal. The extended RREQ has an option format field shown in Figure 3. The option data format is in the Type-Length-Value (TLV) format and loads the desired content index. When the extended RREQ is received by adjacent terminals, the reverse route to the source terminal is made in their routing table. Then, if the content index in the RREQ is registered in the content information table of the receiver terminal, it unicasts an extended RREP to the source terminal. The extended RREP has the same option format as the extended RREQ. The extended RREP loads the information of the desired content and the IP address of the terminal that has the information. If the terminals do not have the entry in the content index in the RREQ, they re-broadcast the RREQ. When the extended RREP is received, the terminal makes the forward route to the terminal that sent the RREP. When the source terminal receives the extended RREP, the routes to the terminal that has the information on the desired content and the actual information on the content are obtained. Sample sequences of the extended path discovery process are shown in Figure 4 (request phase) and Figure 5 (reply phase).

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 3 1	
Type	Length	Value…	
Value… (cont.)			

Figure 3: Option format for content index

4.5 User Application Control Unit

We implemented two types of applications as shown in Section 3.2. One is a chat application that specifies the communication targets in advance. The other is a file sharing application that does not necessarily have to specify the communication targets explicitly. Figure 6 shows the screen images of our implementations of these two types of applications.

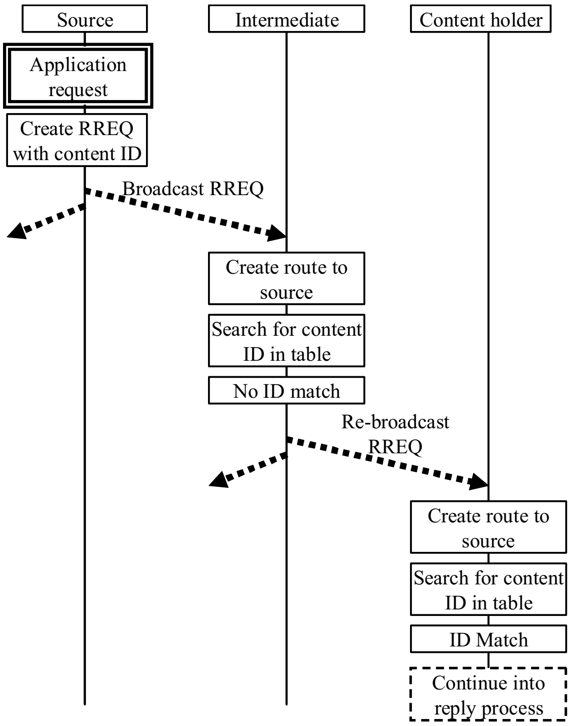

Figure 4: Extended path discovery (request phase).

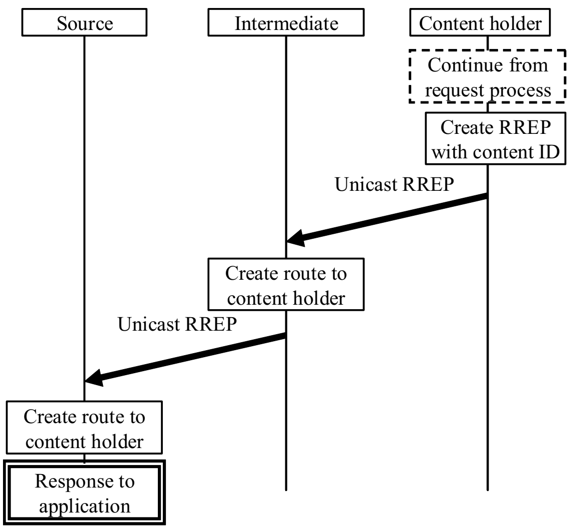

Figure 5: Extended path discovery (reply phase).

(1) Chat application:

We implemented a chat application that communicates peer to peer. Each terminal holds the information on the participants that join the ad-hoc network in the presence information table. The user can select communication targets from the presence information table. Presence information is advertised when the user joins the ad-hoc network. When chat communication begins, the path discovery process starts if a terminal does not have a route to the communication target. When the route is established, the chat data message is transmitted via multi-hop links.

(2) File sharing application:

The file sharing application was implemented as an application that communicates with unspecified targets. Each terminal registers the information on content that they have in the content information table. To find the content and make the routes to the terminal that has the content, we use the extended path discovery process shown in Section 4.4.

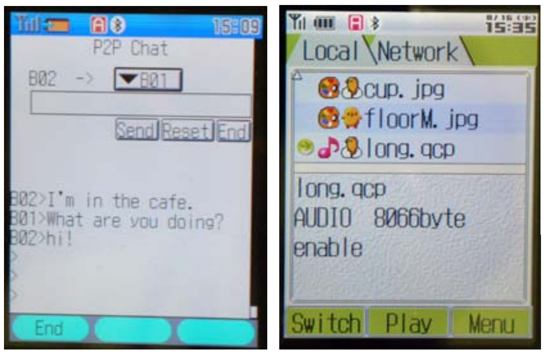

Figure 6: Application screen images of the chat application (left) and the file sharing application (right).

5 PERFORMANCE EVALUATION

We evaluated the fundamental performance of our implementations. In the following experiments, we use five mobile phone terminals and each result is averaged over 10 trials.

(1) Bluetooth device discovery:

We measured the time to successfully execute device discovery. Table 2 shows the average time of device discovery with the number of adjacent nodes. 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.

Table 2: Device discovery time.

Number of adjacent nodes	1	2	3	4
Time (seconds)	1.1	2.1	2.9	4.3

(2) Time to establish routes

We measured the time for route establishment in the experimental network shown in Figure 7. There are two routes between the source and the destination and we denote the upper one as Route #1 (2 hop) and the lower one as Route #2 (3 hop), respectively. The results of route establishment time are shown in Table 3. Total time via Route #1 is about 8.8 seconds and that via Route #2 is about 13.5 seconds. As shown in Table 4 of the Evaluation (3), the time to transmit data per hop is about 1 second. However, the data transmission time for route establishment is more than 1 second per hop. One reason is that broadcasting in the path discovery process interferes with data transmission.

(3) 1-hop data transmission time:

Table 4 denotes the average 1-hop transmission time with transmission data size. This result does not include time for route establishment. When the transmission data size is smaller than 10 Kbytes, the transmission time is about 1 second. The time increases with data size when data size is larger than 10 Kbytes. This is because data fragmentation occurs when the data size is larger than 10 Kbytes. When the number of hops increases, the transmission time increases linearly. Thus, applications where the data size is small enough (e.g., chat applications) are executable. However, to execute applications such as file sharing applications that handle large data sizes, effective mechanisms such as hop limitations are required.

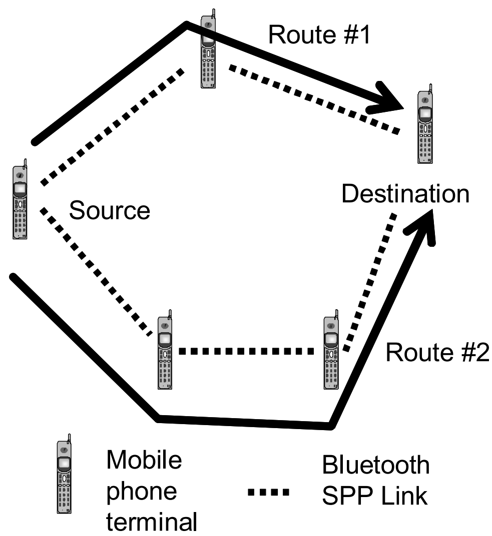

Figure 7: Network topology.

Table 3: Route establishment time.

	Route #1	Route #2
Time for route establishment (seconds)	8.77	13.5

Table 4: 1 hop data transmission time.

Transmission data size (Kbyte)	Time (second)	Transmission data size (Kbyte)	Time (seconds)
1	0.97	10	1.13
2	0.96	50	1.48
3	0.93	100	1.86
4	1.05	500	4.99
5	0.98	1000	10.73

6 CONCLUSION

This paper clarified the design principles for mobile ad-hoc communications and their applications and selected mobile phone terminals as the implementation platform. Based on the above criteria, we then implemented an ad-hoc communication protocol and applications with Bluetooth as the link-layer technology by taking into account the limitation of resources of mobile phone terminals. We further measured the time for each communication procedure and clarified the performance characteristics. In future work, we will further evaluate our implementation in more detail and add the function of co-operation with cellular networks.

ACKNOWLEDGEMENTS

The authors are grateful to Dr. Hirata, Dr. Akiba and Dr. Suzuki of KDDI R&D Laboratories, Inc. for their continuous support and encouragement.

REFERENCES

[1] Bluetooth SIG, http://www.bluetooth.com/

[2] IETF Mobile Ad-hoc network (MANET) WG (charter), http://www.ietf.org/html.charters/manet-charter.html.

[3] C. Perkins, E. Belding-Royer and S. Das, Ad hoc On-demand Distance Vector (AODV) Routing, IETF RFC 3561 (2003).

[4] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol (OLSR), IETF RFC 3626 (2003).

[5] BREW, http://brew.qualcomm.com/brew/en/

[6] AODV-UU, http://core.it.uu.se/AdHoc/AodvUUImpl

[7] Kernel AODV, http://w3.antd.nist.gov/wctg/aodv_kernel

[8] OOLSR, http://menetou.inria.fr/OOLSR/

[9] Unik olsr, http://www.olsr.org/

[10] J. Haartsen, "Bluetooth – Ad-hoc Networking in an Uncoordinated Environment," *Proc. IEEE ICASP*, pp. 2029-2032 (2001).

[11] Z. Jian, Z. Fei and W. Bin, "Cluster Based Routing in Bluetooth Ad-hoc Network," *Proc. IEEE WCNM*, pp. 742-752 (2005).

[12] M. Ronai and E. Kail, "A Simple Neighbor Discovery Procedure for Bluetooth Ad Hoc Networks," *Proc. IEEE GLOBECOM*, pp. 1028-1032 (2003).

[13] F. Formo, G. Malnati and G. Portelli, "Design and Implementation of a Bluetooth Ad Hoc Network for Indoor Positioning," *Proc. IEE IP-SEN*, pp. 223-228 (2005).