ProxNet: Secure Dynamic Wireless Connection by Proximity Sensing

Jun Rekimoto, Takashi Miyaki, and Michimune Kohno

Interaction Laboratory, Sony Computer Science Laboratories, Inc.
3-14-13 Higashigotanda, Shinagawa-ku, Tokyo 141-0022 Japan
http://www.csl.sony.co.jp/person/rekimoto.html

Abstract. This paper describes a method for establishing ad hoc and infrastructure-mode wireless network connections based on physical proximity. Users can easily establish secure wireless connections between two digital devices by putting them in close proximity to each other and pressing the connection button. The devices ”identify” each other by measuring each other’s signal strength. We designed a set of protocols to support secure connections between digital devices by using a proximity communication mode to exchange session keys. We also introduce a ”dummy point” that is analogous to a wireless access point but handles proximity-mode communication. The dummy point represents physical locations of digital devices and supports context-sensitive network communications.

1 Introduction

Digital appliances based on wireless networks are becoming increasingly popular, and the operation of such appliances differs from the operation of conventional devices based on wired networks. Users of wireless appliances frequently change network connections depending on their needs and typically should be able to perform the following operations:

– exchange digital pictures between two digital cameras by dynamically creating an ad hoc wireless connection between the two cameras. They should be able to see the screens of both devices to interactively decide which pictures to copy from one device to the other. They also need to transmit files securely so unauthorized users cannot intercept their files.
– connect a digital device to the Internet in a public place in order to share files with other users. This file sharing should be protected from other users at the same hotspot.
– use wireless mobile game devices to dynamically establish ad hoc wireless connections in order to play games on the network.
– dynamically connect a PDA to a projector (for example, during presentations) without knowing the IP address and login password of the projector.
– upload pictures from a cellular phone to a public wall display.

To do these things, users must be able to establish and change network connections quickly. We therefore need to

(1) develop a method to identify target devices Users should be able to identify devices they want to connect to without explicitly specifying them by IP addresses or other symbolic references.

(2) provide communication security Established wireless connections must be secure in the sense that users should be able to connect to the correct devices and that communication on these connections should be protected to prevent the interception of files by a third party.

While traditional networking largely relies on specifying destination devices by (IP) addresses and using passwords for authentication, the management of nearby wireless devices can be based on the physical relationship between them. The establishment of network connections, for example, can be based on the proximity of devices. Devices can also be connected via a wireless connection by pointing to a target device (e.g., a TV set) with a mobile device (e.g., a PDA). In both cases, once a wireless connection has been established, device location is no longer restricted (i.e., the devices do not have to remain in proximity and the user does not need to keep pointing at the target device).

The sensors and user operations required in some previously developed systems based on this idea are outlined in Figure 1 [7, 6]. These systems establish wireless connections by using communication devices with a limited range, such as an RFID tag/reader, or directional communication methods such as infrared beaming. These devices and methods, however, are not always suitable for all mobile devices. In this paper we describe a simpler method that uses standard wireless cards and establishes wireless connections by measuring the signal strength of received packets.

Sensor Type	User Operation	Required Sensors / Devices
RFID[7]	put devices in proximity	RFID tag / reader
Infrared[7]	pointing to device A with device B	infrared transmitter-receiver
SyncTap[6]	Synchronous user operations (e.g., synchronized pressing and releasing of buttons)	buttons etc.

Fig. 1. User operations and sensors for establishing wireless connections between devices.

2 ProxNet: establishing wireless connections based on proximity

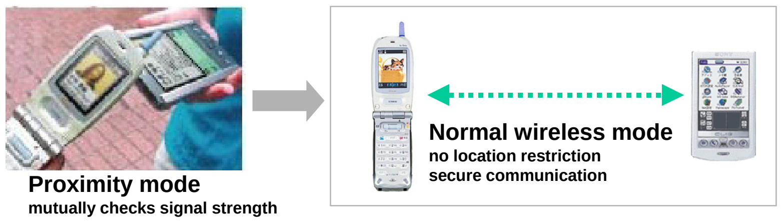


Fig. 2. Establishing a wireless connection by measuring the signal strength

Figure 2 shows the idea behind our proposed method. Unlike other location-sensing methods based on wireless signal strength [1], our method determines only the relative distance between devices. The proximity of two devices can be determined because the strength of a signal one receives from the other is inversely proportional of the square of the distance between them.

Our method has two communication modes: a ”proximity mode” in which both devices check the signal strength of incoming wireless packets, and a normal wireless mode. When users want to establish a network connection between two wireless devices, they put these devices close to each other and press the ”connection” button on one of these devices. The first device then transmits a wireless packet containing the initiator’s address and a dynamically generated public key. The second device receives this packet, and it acknowledges receipt of the packet only when the signal strength of this packet is greater than a predefined threshold. The acknowledging packet contains the responder’s address and a one-time session key that will be used to establish a wireless connection. Because this acknowledging packet is encrypted by the public key sent from the first device, only the first device can access the information in it. The first device then checks the signal strength of the acknowledging packet and establishes a normal wireless connection if it is greater than the threshold..

We implemented this method by using an Intersil PRISM2 802.11b wireless chip with our modified version of the ”Host AP” Linux wireless driver [4]. The Host AP device driver supports a ”monitor” mode that allows applications to monitor wireless packets transmitted by other devices. This mode also supports the monitoring of signal and noise levels (RSSI information) on each packet. While the original monitor mode is receive-only and does not support packet transmission, our modified driver allows the monitor mode to both receive and transmit packets. This modified driver makes it possible for devices listen to transmissions from other devices, recognize connection requests, check signal strength, and then use the normal wireless mode.

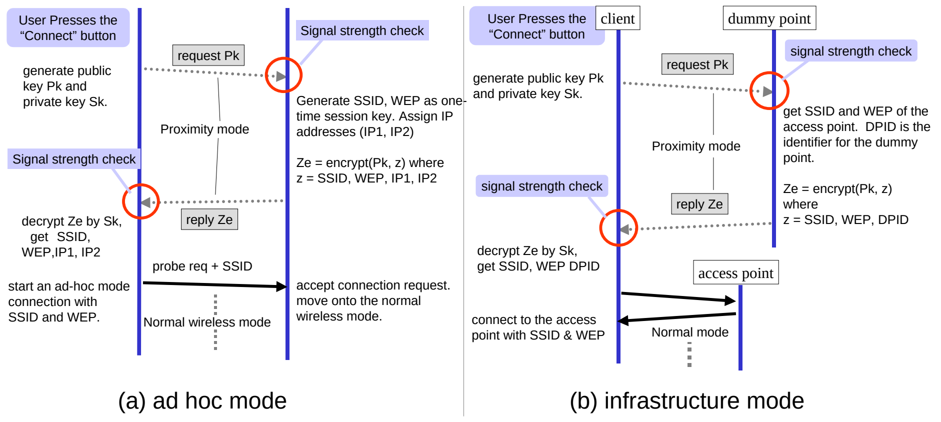
Because this device driver modification does not require any hardware changes, it can potentially be used with a variety of wireless devices without requiring special hardware or sensors. It should also be possible to apply this method to wireless networks other than 802.11, such as Bluetooth.

3 Communication Protocols

The actual protocols are shown in Figure 3. There are two modes for establishing network connections: the ad hoc mode, in which the devices establish wireless connections themselves, and the infrastructure mode, in which the devices communicate through an access point.

Ad hoc Mode To establish an ad hoc mode wireless connection, two devices must share the same network identifier (e.g., ESSID) and a session key (e.g., WEP key). This information is exchanged in the proximity mode (Figure 3-a). In this mode, the signal strength of the received packets is measured on both sides. Once the information needed for establishing an ad hoc connection has been exchanged, the two devices start normal wireless communication. The WEP key and ESSID are dynamically generated for each session, and users do not need to manually enter them.

Infrastructure Mode and the ”Dummy point” An operation similar to the one in the ad hoc mode could be used in the infrastructure mode: users could put mobile devices close to an access point and exchange the necessary information. This operation is not always practical, however, because access points are often located in physically inaccessible places (e.g., near the ceiling).



User Presses the "Connect" button
generate public key Pk and private key Sk.
request Pk
Signal strength check
Proximity mode
Generate SSID, WEP as one-time session key. Assign IP addresses (IP1, IP2)
Ze = encrypt(Pk, z) where z = SSID, WEP, IP1, IP2
Signal strength check
reply Ze
decrypt Ze by Sk, get SSID, WEP,IP1, IP2
probe req + SSID
start an ad-hoc mode connection with SSID and WEP.
accept connection request. move onto the normal wireless mode.
Normal wireless mode
(a) ad hoc mode
User Presses the "Connect" button
client
dummy point
request Pk
signal strength check
generate public key Pk and private key Sk.
Proximity mode
get SSID and WEP of the access point. DPID is the identifier for the dummy point.
signal strength check
reply Ze
Ze = encrypt(Pk, z) where z = SSID, WEP, DPID
decrypt Ze by Sk, get SSID, WEP DPID
access point
connect to the access point with SSID & WEP
Normal mode
(b) infrastructure mode

Fig. 3. ProxNet protocols for ad hoc and infrastructure-mode wireless connections.

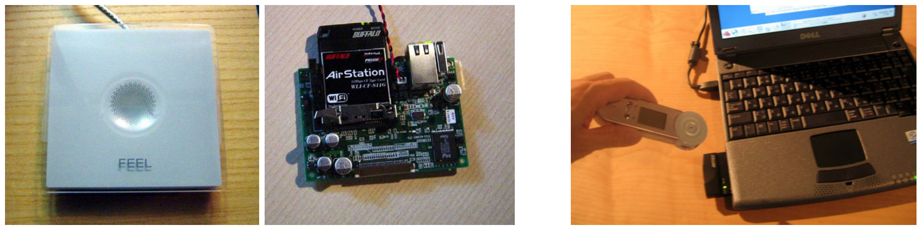


Fig. 4. Dummy point examples: (left: dummy-point modules based on a single board Linux. right: a handheld dummy point)

We therefore introduce a ”dummy point,” which is a wireless device that supports only proximity-based communication. For example, a dummy point can be placed on a meeting table to enable users to connect their mobile devices (e.g., wireless notebook PCs) to the access point in that room. Users do that by putting their mobile devices close to the dummy point, and the dummy point responds to the request packets from the mobile devices by sending information needed to establish an infrastructure-mode wireless connection (e.g., the ESSID and the WEP key for the access point). Then the mobile devices connect to the access point (Figure 3-b).

The physical form of the dummy point can vary, and a hand-held dummy point (Figure 4, right) can be used. At an Internet cafe, for example, a manager can put a dummy point like this close to a notebook PC in order to ”invite” it into the cafe’s wireless environment.

The dummy point can also support context-sensitive communication. We can place more than one dummy point in any environment, and each dummy point can represent a location or a corresponding object. When a user starts a wireless connection by using a dummy point placed next to a printer, for example, the default output device would be automatically set to that printer. Similarly, a wall-sized public display can have a corresponding dummy point. People can upload data by establishing a wireless connection with the corresponding dummy point.

4 Applications

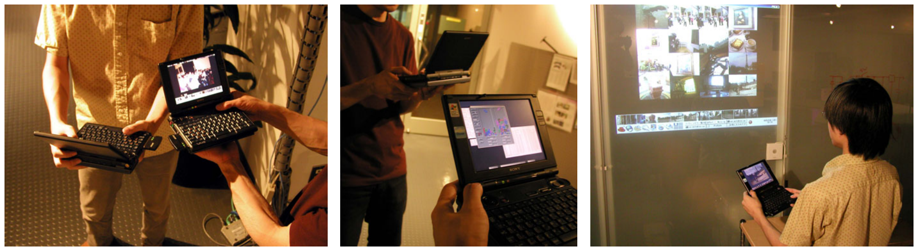


Fig. 5. Applications of ProxNet networking.

Sharing files by using an ad hoc wireless connection (Figure 5, left) Two mobile devices start an ad hoc- mode network connection and exchange files. Users first establish a connection by putting two devices close to each other and pressing the connection button on one of them. Once the connection has been established, users do not need to stay close to each other. By sharing the screens of their devices, users can interactively decide which files to transmit. This cannot be done in file transmission using physical media such as memory cards.

Setting up a network game session (Figure 5, middle) A network game can be set up by putting two devices close to each other. Users can also create connections dynamically during a game session. For example, a user can start a game in a single-user mode but then share this game with other players on the network.

Transferring information between mobile devices and public wall displays (Figure 5, right) People working in the same office can share information by uploading it to a public wall-mounted display. They approach this display and put their mobile devices close to the dummy point attached to it. Then they can upload digital picture files onto the public display or get files from the display. Similarly, users making a presentation can display presentation materials from a mobile device by first approaching a dummy point placed near the presentation screen. This method can also be used with digital devices without user interfaces. A file server or a network router, for example, can be controlled from mobile devices in this way.

5 Related Work

Pick-and-Drop [5] and mediaBlocks [10] are systems using physical manipulation to support digital data transfer among various devices. While these systems were designed primarily for ”one-shot” data transfer using physical objects, our method supports continuous wireless network connections between devices. Our method also supports secure communication.

Several other systems use range-limited communication channels, such as those provided by infrared beaming or used by radio-frequency identification (RFID) tag reader/writes, for associating wireless devices [9, 7, 8]. Our method uses wireless communication in two ways – one is for normal wireless communication and the other is for sensing signal strength (i.e., for measuring a range) .

The measuring of signal strength has been used for network security management in various applications, such as wireless keys for automobiles or login keys for computers. In these applications, however, only the signal strength of the key is measured. Our method supports a two-way measuring of signal strength for better security.

Another way to recognize spontaneous device associations is by using synchronicity [3, 6, 2]. For example, if two devices were tied together and moved together, the movements of both devices would be quite similar. Devices with acceleration sensors would be able to exchange acceleration information to find matching devices [3]. Similarly, if a user performs synchronized inputs, such as pressing buttons on two devices, this information can be used for finding device pairs [6]. These synchronicity approaches and our proximity-sensing approach are complementary and can be used in combination. For example, the selection of candidate devices to be associated can first be based on signal strength, and then an actual pair can be determined by synchronicity.

6 Conclusion

We developed a method for establishing secure wireless connections between mobile devices by measuring the strength of signals between them. The method supports ad hoc and infrastructure modes, and the same operations can be performed in both modes. This method uses normal 802.11 wireless chipsets without additional hardware and is applicable to a wide range of wireless devices.

References

1. P. Bahl and V. Padmanabhan. RADAR: An In-Building RFBasedUser Location and Tracking System. In *IEEE Computer and Communications Societies (INFOCOM 2000)*, pp. 775–784, 2000.
2. Ken Hinckley. Synchronous gestures for multiple users and computers. In *ACM User Interface Software and Technologies (UIST 2003)*, pp. 149–158, 2003.
3. L. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H. Gellersen. Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts. In *Ubicomp 2001*, pp. 116–122. Springer-Verlag, 2001.
4. Jouni Malinen. Host ap driver for intersil prism2/2.5/3. http://hostap.epitest.fi/.
5. Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'97*, pp. 31–39, October 1997.
6. Jun Rekimoto. Synctap: An interaction technique for mobile networking. In *Proc. of MOBILE HCI 2003*, 2003.
7. Jun Rekimoto, Yuji Ayatsuka, Michimune Kohno, and Haruo Oba. Proximal Interactions : A direct manipulation technique for wireless networking. In *Proc. of INTERACT 2003*, 2003.
8. Colin Swindells, Kori M. Inkpen, John C. Dill, and Melanie Tory. That one there! pointing to establish device identity. In *Symposium on User Interface Software and Technology (UIST'02)*, pp. 151–160, 2002.
9. Peter Tandler, Thorsten Prante, Christian Muller-Tomfelde, Norbert Streitz, and Ralf Steinmetz. Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In *Proceedings of the 14th annual ACM symposium on User interface software and technology (UIST 2001)*, pp. 11–20, 2001.
10. Brygg Ullmer, Hiroshi Ishii, and Dylan Glas. mediaBlocks: Physical containers, transports, and controls for online media. In *SIGGRAPH'98 Proceedings*, pp. 379–386, 1998.