

Mobile Multimedia Middleware for Seamless Service
Ken Ohta, Takashi Yoshikawa, Tomohiro Nakagawa, Yoshinori Isoda, Toshiaki Sugimura
Multimedia Laboratories, NTT DoCoMo, Inc.

Mobile phones with limited input/output capabilities can provide high-quality multimedia services that take advantage of next-generation mobile broadband by using information-related home electronics to perform media input/output. Mobile phone services such as phone calls have to be uninterrupted while moving, and services using external devices must also support mobility. This study proposes the READY alternate service finding and reserving method that reduces interruptions and improves service continuity in media transfers when switching to a new device or when a device fails during use, and that is designed as mobile multimedia middleware (MMM). We installed a streaming music playback application on MMM and measured the continuity in media playback during device switching and device failure to evaluate the effectiveness of READY.

1. Introduction

[01] IMT-2000 [1], or the so-called third generation, will provide mobile multimedia services such as video calls and music distribution, and mobile phones will evolve as multimedia terminals that handle video and music in addition to text, voice, and images. Fourth generation mobile communication systems are expected to handle even faster mobile broadband (around 100 Mbps)¹.

[02] However, even though mobile phone terminals ARE moving toward handling more multimedia, the input/output devices such as screens, speakers, microphones, and cameras built into mobile phones are limited in terms of size and quality compared to information-related home electronics and PCs in order to ensure portability. As a result, it is difficult to provide high-quality multimedia over mobile broadband using mobile phones alone, and information-related home electronics with higher-quality external monitors and external speakers will have to be used. For example, use of nearby large-screen televisions or speakers to output high-quality content obtained from a video/music distribution service via Bluetooth [2] and use of nearby and remote cameras and microphones to provide video telephone service are being considered.

¹ Prospects for a New Generation of Mobile Communication Systems:
www.joho.soumu.go.jp/pressrelease/japanese/sogo_tsusin/010625_6.html

[03] It is essential that mobile phone services such as phone calls remain uninterrupted while moving, and supporting mobility is an issue when providing services using external devices. It will be necessary to enable device switching (known as device handoff) that does not interrupt voice and video transfers when a high-quality external camera and speaker are found at the destination during a video call using just a mobile phone. Meanwhile, when the user moves away from the camera and speaker or connectivity with these devices deteriorates, any failure has to be detected and service switched immediately to the camera and headset, etc. built into the mobile phone to ensure uninterrupted video telephony service.

[04] This study proposes the READY alternate service finding and reserving method to solve the continuity problem in mobile multimedia services using external devices. READY improves continuity by managing connections for both active and new devices to smoothly switch devices, and by detecting device failures and switching to alternate devices. READY is designed as mobile multimedia middleware (MMM), and MMM is used to input and output media so that applications can seamlessly provide service without device switching concerns.

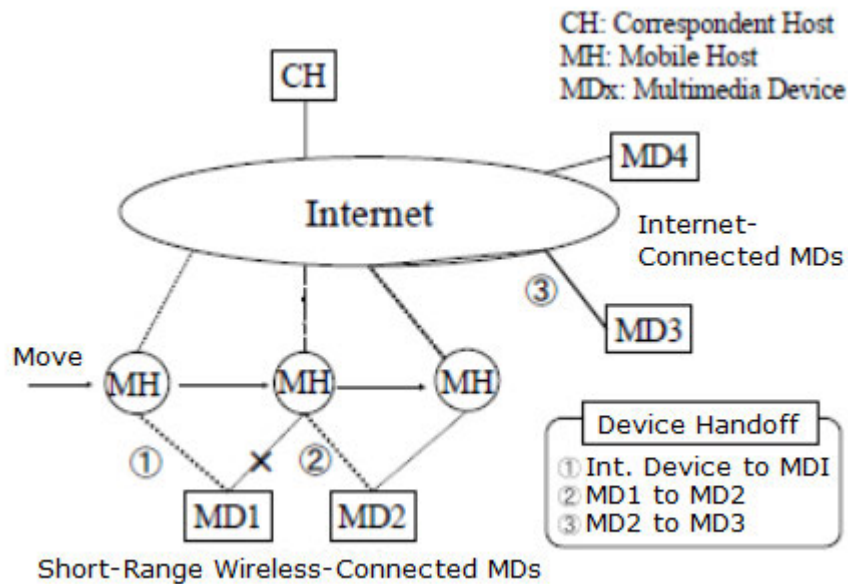


Figure 1: Study Model

[05] In Section 2, we describe the study model and review related research. In Section 3, we present our READY proposal and our MMM design. In Section 4, we install a streaming music playback application on the MMM and measure the continuity of media playback during device switching and failure in order to evaluate the effectiveness of READY. Finally, we provide a conclusion in Section 5.

2. Background

2.1 Study Model

[06] The study model consists of three elements: a mobile host, a multimedia device, and a companion host (Fig. 1).

[07] Mobile Host (MH): This is a mobile terminal such as a mobile phone or PDA that is equipped with several wireless interfaces such as Bluetooth, IrDA, wireless LAN, and cellular, etc., and that can connect to the Internet. Also, we assume that the multimedia device being used supports service finding and control protocols.

[08] Multimedia Device (MD): This is information-related home electronics such as a camera or speaker that is equipped with media input/output and a network interface, and that can be connected to the mobile host (MH) indirectly via the Internet or directly via Bluetooth, etc. A multimedia device (MD) that connects directly via short-range wireless is limited to the range reachable by wireless, whereas an indirect connection can be used as long as

there is an Internet connection. Here, we assume that the multimedia devices (MD) also support service finding and control protocols.

[09] Companion Host (CH): This is a host on the Internet that is used to realize services in conjunction with a mobile host (MH), such as a remote terminal for video telephony service or a content distribution server for a music distribution service.

[10] Services are provided by an application running on a mobile host (MH) and an application running on a servers on a companion host (CH) cooperating with each other. This paper focuses on real-time multimedia services such as video telephony and music distribution, and a form is considered in which services are provided by switching the media input/output destination of an application in a mobile host (MH) from an internal device to an external device and then to another external device. There are two triggers for device switching: when a device more suitable than the current device is found (device finding), and when the current device becomes unusable (device failure). The switching is performed when an instruction is received from the user or when the system makes a certain determination. Figure 1 shows three examples of device switching.

1. An external device MD1 that can be connected to via short-range wireless is found, and a handoff is performed from the internal device to MD1. In other words, the media input/output destination is switched from the built-in device to MD1. In one example, when the user gets into a car, the mobile host (MH) connects to the Bluetooth car speakers in the car and changes the music output from the built-in speaker to the car speakers.

2. MD1 has become unavailable (device failure) and a handoff is performed to an alternate external device MD2. In the same example, when the user gets out of the car and the car speakers become unusable, the user switches to a portable Bluetooth headset to continue listening to music.

3. A device MD3 on the Internet that is of higher quality than the current device MD2 is found and a handoff is performed. In one example, the mobile host (MH) accesses the stereo speakers in a living room via a home wireless LAN and switches the music output destination from the headset to the stereo speakers.

2.2 Related Research

[11] One of the features of this study is that it deals with the continuity of real-time services over devices. Position-oriented distributed multimedia architecture [3] deals with mobility such as switching input to a nearby camera or outputting music to a nearby speaker as the user moves, but there is no real-time or continuous media transfer. Service architecture [4] that deals with discovery and interface issues when heterogeneous mobile terminals use devices on the move and function-oriented object models VNA [5] that create a new virtual device by combining surrounding information-related home electronics also experience service interruption.

[12] Another feature of this study is that it targets handoffs between mobile terminals and external devices in a personal area network (PAN). Research on protocols such as Mobile IP [6], HAWAII [7], and Cellular IP [8] for supporting macro/micro-mobility of the latest terminals, and on a handoff method between heterogeneous wireless systems known as Vertical Handoff [9] also addresses handoffs between terminals and networks. In this study, we deal with the diversity of links used to connect each device, and examine a smooth device handoff method that takes the device finding and connection process into consideration. Various techniques such as the Service Location Protocol (SLP) [10], HAVi²,

Jini [11], UPnP³, Salutation⁴, and Bluetooth have been proposed as device (service) finding and access techniques, and this research uses these as a starting point.

3. Mobile Multimedia Middleware

[13] The following three are believed to be system requirements for seamlessly providing services using external devices, which is the purpose of this study. Continuity, adaptability, and versatility are requirements for devices, media, and networks, respectively. As this paper focuses on devices, we propose the READY alternate service finding and reserving method to solve continuity. Adaptability and versatility are dealt with separately in the environment-adaptive mobile terminal architecture described below.

- Continuity: Uninterrupted media transfer is provided to the user even when devices are switched or a device failure occurs.
- Adaptability: Content or transfer media is provided that is compatible with device capabilities and communication speeds. Because the screen size, number of speakers, and other capabilities differ from device to device, content quality and transfer media quality have to be adjusted when switching between devices.

² www.havi.org; HAVi-Home Audio/Video Interoperability

³ www.upnp.org; Universal Plug and Play Forum

⁴ www.salutation.org; Salutation Consortium

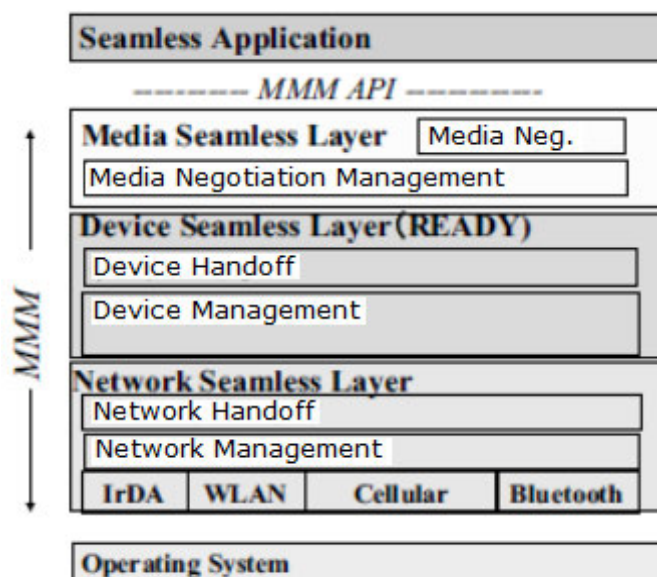


Figure 2: Environment-Adaptive Mobile Terminal Architecture

- Versatility: Supporting different links used to connect with devices and different handoff mechanisms for heterogeneous wireless access, such as switching from Bluetooth to cellular.

[14] READY manages connections to both current and new devices, provides a mechanism for smooth device switching, detects device failures, and improves continuity by switching devices when a device failure occurs. An architecture is also adopted that provides the READY mechanism as mobile multimedia middleware (MMM). Incorporating the READY code

[15] Figure 2 shows the environment-adaptive mobile terminal architecture. The MMM consists of three layers, and READY corresponds to the device seamless layer. The media seamless layer resolves compatibility issues by negotiating content quality with, for example, a content delivery server and performing media conversion based on device capabilities. The network seamless layer resolves versatility issues by enabling device connection and network handoff using various wireless interfaces included in mobile terminals. The architecture and modules for each seamless layer are described in detail in [12].



[16] READY solves the problem of interrupted media transfers during device switching and device failure.

Exhibit 1004
Hewlett Packard Enterprise Company v. Sovereign Peak Ventures, LLC

[17] Device handoff has to take into consideration the external device finding and connection process. If media I/O is switched to a device that is not ready during the connection process, media I/O will be interrupted. While connecting with a new device, it is necessary to continue using the original device and to switch the media transfer destination once the new device is ready. This means that even if the user gives a device switching instruction, the device cannot be used immediately. The external device finding and connection time depends on the number of exchanges between a mobile host (MH) and a multimedia device (MD) required to establish a connection, communication delays between the mobile host (MH) and the multimedia device (MD), and the multimedia device (MD) preparation time. If the delay is long, it may take several seconds. If the device takes a long time to recover from the standby state, such as a projector or printer, it may take anywhere from several dozen seconds to several minutes.

[18] To solve these problems, READY performs a periodic device search, media transfer destination switching (smooth handoff) after device connection is completed, and pre-connection with a reservation option. Figure 3 shows the procedure followed when an application that has been outputting video to an external device MD1 switches output to a newly found device MD2.

1. Periodic searching: READY periodically (for example, every 30 seconds) activates the service search protocol and notifies the user of information on MD2 which has been found. The device search is performed automatically in the background to reduce the user's perceived waiting time for discovery of a new device.
2. Pre-connection (optional): If the reservation option is set, the connection process with MD2 is started immediately after MD2 has been found. The purpose is to reduce the waiting time after the user gives an instruction to switch devices.
3. Device switching instruction: The user instructs the device to handoff to MD2 via the user interface used to control device selection and switching provided by the MMM.
4. Smooth handoff: With connections established to both MD1 and MD2, READY switches the media destination from MD1 to MD2. If the reservation option is not set, handoff is executed after connection processing with MD2 is completed, so there is a possibility that a delay will occur after the user issues an instruction to switch. If the connection has been completed by pre-connection, handoff is executed immediately after the user's device switching instruction.

3.1.2 When a Device Fails

[19] In order to shorten the service interruption time due to an external device failure and improve service continuity, switching to an alternate device must occur immediately after the failure is detected. READY solves this problem by periodically monitoring for device failure and preparing alternate devices.

- Periodic monitoring: The status of the device in use is periodically queried. If an abnormality notification occurs or no response is returned, it determines that a device failure has occurred and executes a handoff to an alternate device.
- Alternate device preparation: An option is provided to maintain a connection with an alternate device as a backup for the device currently being used. When the device currently being used fails, processing does not have to be performed to connect with an alternative device, so immediate handoff occurs and the service interruption time can be reduced.

3.2 Middleware Design

[20] The reason for designing READY as middleware is to hide device switching from the application and simplify application development. If the address or interface (API) used by the application to access a device has to be changed each time a device is switched, the burden on the application is significant. For example, similar video input devices, such as webcams and built-in video capture devices, may require different behavior from applications if they use different APIs.

[21] In order to address interface issues, the MMM is based on object orientation and treats devices of the same type, such as built-in speakers and headsets, as wrapped objects (real device objects) that provide a common API. In order to address access issues, we introduce the concept of virtual device objects that are independent of physical devices. While access to real device objects is dynamically created and eliminated through device discovery and failure, virtual device objects logically guarantee permanent access. A real device object is bound to a virtual device object in READY, and the media input/output from the application is inputted and outputted to a real device object via a virtual device object. By binding a new real device object to the virtual device, a device can be switched without the application having to be aware of it.

[22] The MMM provides applications with virtual device objects such as audio output and video input. Below is a C++ based class design for an audio output virtual device object. A virtual device object provides device status (normal/abnormal, in use by another user, etc.) and attributes (quality, location) as data, and device control as a method. It also has a pointer for binding a real device object. When the application calls the playback method AudioOut to output audio data, the input audio data is passed to the bound real device object.

```
// Define Audio Output Virtual Device Object
class CVAudioOutDevObj
{
public:
    // --- Data Section ---
    int    m_status;    // normal, fault, occupied
    char*   m_quality;  // ex. "stereo headphone"
    char*   m_location; // ex. "portable", "S105room"
    // Bind Actual Device Object
    CRAudioOutDevObj* m_devobj;
    ....
    // --- Control Section ---
    int Bind( char* specification ); // binding
    int AudioOut( BYTE* sample ); // output
    int SetVolume( int volume ); // volume control
    ....
};
```



Figure 4: Streaming Music Playback Application

[23] An example of an application program is shown below. First, a virtual device is generated and bound to a real device, and then input/output is performed to the device via the virtual device. If they use virtual devices for media I/O, applications do not have to be aware of READY device switching. In the future, we will have to define how to specify parameters such as the location (nearby or remote specific location), quality (screen size, etc.), and features (portable, stationary, etc.) during the binding process. In the current implementation, the MMM automatically selects a device based on the quality attributes of real device objects.

```
// Declare Virtual Object
CVAudioOutDevObj* m_audiout;
// Generate Interface
m_audiout = new CVAudioOutDevObj();
// Bind to Actual Device
m_audiout->Bind( "best quality");

// Output Audio Data
char* buf;    // Audio Data Buffer
int len;      // Buffer Length
m_audiout->AudioOut( buf, len );
```

4. Installation and Evaluation

[24] To evaluate the effectiveness of READY and the MMM, we implemented a streaming music playback application in a MMM prototype and measured media playback continuity during device switching and device failure. Figure 4 shows the implemented streaming music playback application. Icons indicating external devices and built-in devices found by READY and their device status are displayed on the right side of the screen. The user can

then switch the music output destination device (issue a device handoff instruction) by clicking on an icon.

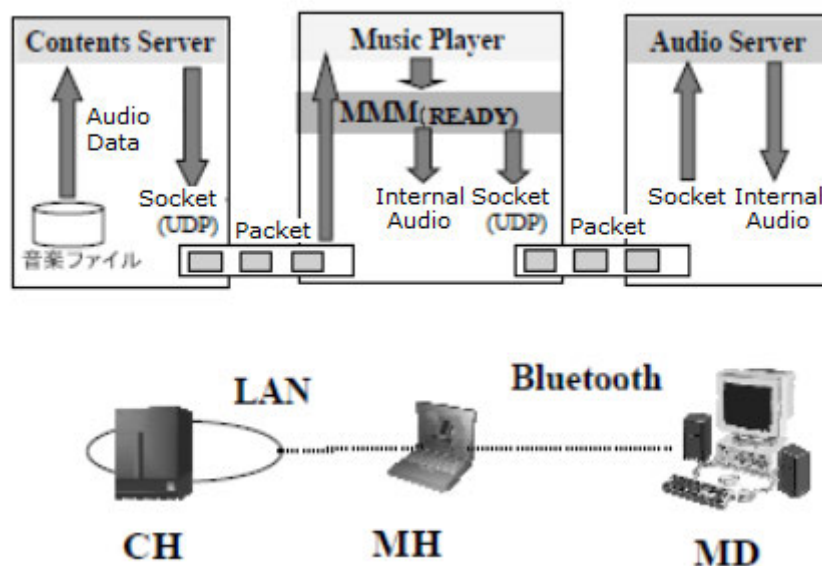


Figure 5: Test Environment

[25] Fig. 5 shows the test environment. Two PCs equipped with Bluetooth are arranged within communication range of each other that are a mobile terminal MH and an external device MD. The MH is also connected to a 100 Mbps Ethernet LAN and communicates with a content distribution server CH over the LAN. Since this test aims to measure the effects of device switching and device failure on media transfers between the MH and MD, the MH and CH are connected via a LAN with very little delay or stutter.

[26] The streaming music playback application described above is executed on MH, and when the user presses the play button, starts receiving audio packets from CH and writes audio data to the virtual device for audio output. The CH reads audio files (8 bit, 8 KHz, stereo) at intervals of about 100 milliseconds, converts the audio data into UDP packets, and sends them to the MH. In order to absorb the stutter from packet transfers between the MH and CH, the application starts playing after audio data has been stored in a 2-second buffer after reception from the CH has started. Audio data received by the MH is outputted to the built-in device by the MMM or sent to the MD via a socket.

[27] The audio device program in the MD receives audio UDP packets from the MH, and starts playing after a 500 millisecond buffer for stutter absorption has built up, as in the case of the application. Because the longer the buffer takes, the later the playback starts after handoff to the MD, a smaller buffer is used than in the application. Taking the buffer buildup time into consideration, we added a process for continuous playback from the built-in device for 500 milliseconds when implementing the MMM. We will consider incorporating buffer adjustments and management as a function of READY in the future.

Table 1: Average Processing Time (Seconds)

Find Device	Determine Name	Link	Connect Socket
11.0	2.0	2.7	0.3

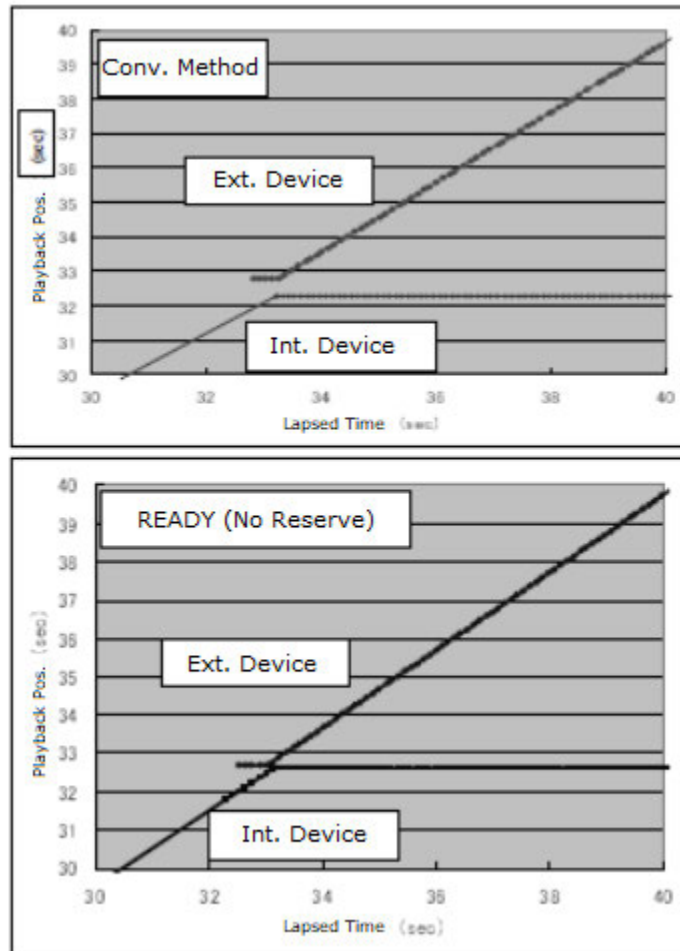


Figure 6: Conventional Method vs. READY

[28] After a Bluetooth device search and device name resolution, the MH establishes a Bluetooth link to the MD using a network profile that emulates Ethernet, and performs a socket connection on that link. Table 1 shows the average processing time for each procedure. For easier implementation, the MD Bluetooth device name includes information on the socket connection such as an IP address and port number, and device quality information (e.g. "AudioDev1 : 192:168:38:12 : 8017 : 16 bit; 8 KHz; Stereo").

4.1 Continuity When Switching Devices

[29] The stuttering of media playback was measured when the MH hands off from an internal device to a newly discovered external MD. Three cases were compared: a case in which READY is not used and cases in which READY is used with and without the reservation option.

1. Conventional method: A device connection is started when the user issues a device switching option, and the device is switched without waiting for a device response from the socket connection (before completion of preparations).

2. Without reservation: The connection is started when the user issues a device switching instruction, and switching is performed after the connection has been completed.

3. With reservation: A device connection is established immediately after the device is found, and the switch occurs immediately after a device switching instruction is received from the user.

[30] In the test, the device search process was executed in the background five seconds after playback starts, and a device switching instruction was issued after 30 seconds had elapsed. Figure 6 shows a comparison of the playback status between READY and the conventional method. The horizontal axis represents the elapsed time, and the playback positions in the audio data played back by the built-in device (MH) and the external device (MD) are plotted. Both completed the connection with the external device after about 33 seconds of elapsed time, and the external device started playing after 500 msec of buffer buildup time. While the conventional method had a playback gap of 0.5 seconds from the playback positions at 32.3 seconds to 32.8 seconds, READY ensured continuity without any gaps, which confirms the effectiveness of READY. The gaps in the conventional method were 0.5 seconds long on average and were caused by a loss of audio packets sent to the MD before the connection had been completed.

[31] Fig. 7 shows a comparison between the use and non-use of the reservation option. Playback started from the external device at an elapsed time of 33.0 seconds when the reservation option was not used, and playback started at an elapsed time of 30.6 seconds when the reservation option was used. This confirms that the reservation option improves responsiveness to user handoff requests. When the reservation option was used, external device playback started an average of 0.5 seconds after a user handoff instruction. This delay is due to the MD buffer buildup time of 0.5 seconds. Meanwhile, when the reservation option was not used, it took an average of 3.1 seconds to complete the handoff. Link and socket connection times are overhead for handoffs.

4.2 Continuity When a Device Fails

[32] Next, we set the READY device monitoring cycle for 1 second and 3 seconds and set the timeout to twice that, 2 seconds and 6 seconds, before comparing the media playback (Fig. 8). READY queries the status of the MD at this cycle interval, and determines that a device failure has occurred if no response is received within the timeout period. After handing off from the built-in device to the MD as in the previous test, the audio device program in the MD was forcibly terminated to cause a failure. As a result, we were able to confirm that both handed off to the built-in device 2.3 seconds and 6.2 seconds after a status response was last received. This corresponds approximately to the timeout time.

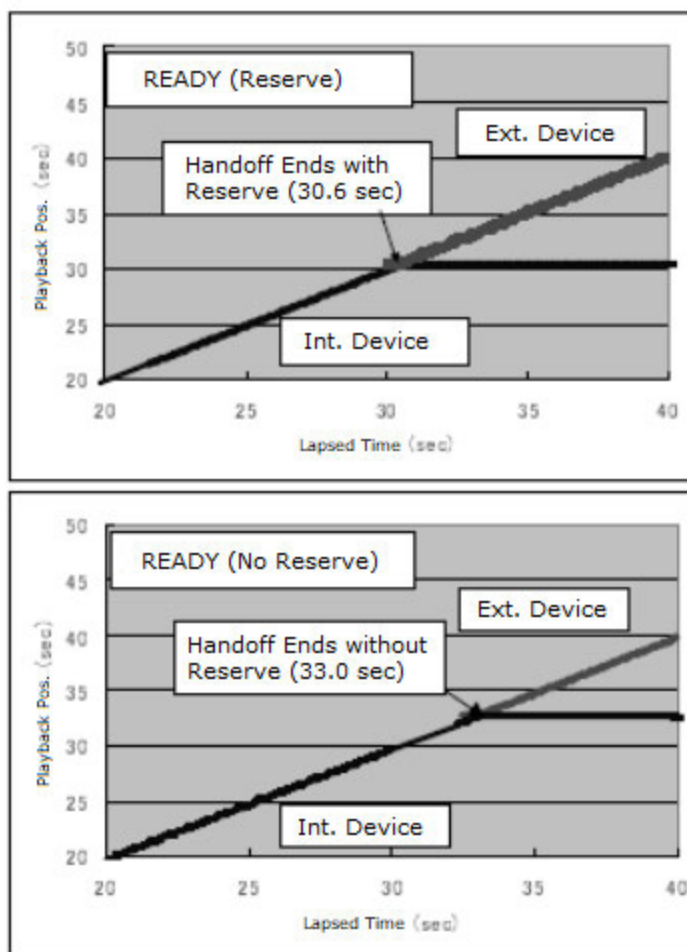


Figure 7: Reserve Option Comparison in READY

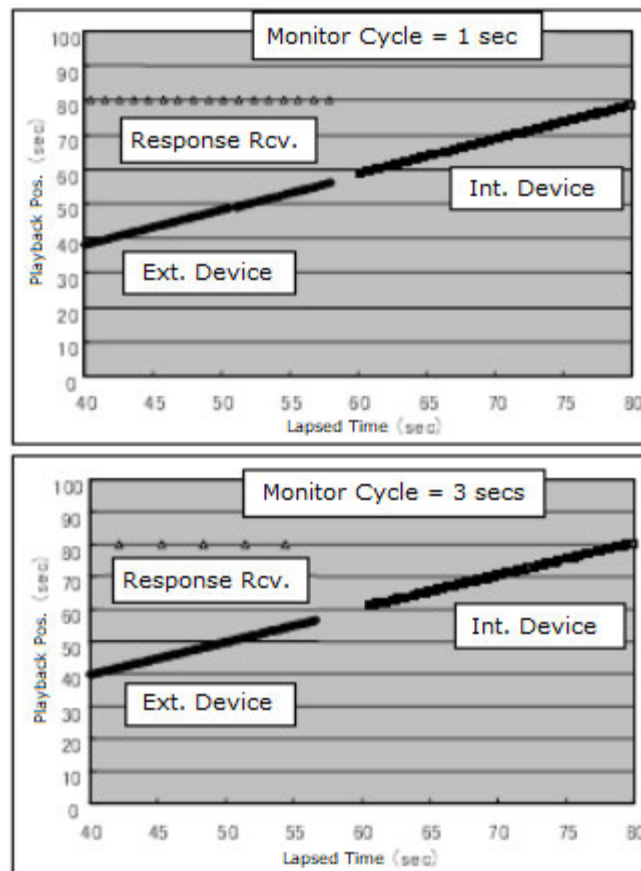


Figure 8: Media Playback During Device Failure

[33] However, regarding gaps in playback, a 1.9-second interruption was observed for a monitoring period of 1 second, and a 3.7-second interruption was observed for a 3-second monitoring period. This confirms that a shorter monitoring period improves continuity. If the time of the failure (forced termination) is immediately after the last status response has been returned, the interruption will be significant, while if it is just before the response, the interruption will be less significant. In both cases, however, testing confirmed that the interruptions were limited to being within the timeout period. In applications that do not allow for service interruptions, continuity of service can be improved by setting a short monitoring cycle, although communication fees and battery power are consumed.

5. Conclusion

[34] In this paper, we proposed the READY alternate service finding and reserving method in order to solve the problem of continuity in mobile multimedia services using external devices, and described its design and implementation as middleware. In addition, we performed streaming music playback testing and confirmed that READY can guarantee media transfer continuity when switching devices and can detect failures by periodically querying devices and switching to an alternate device.

[35] The information-related home electronics around us are expected in the future to be connected to the Internet and to have wireless communication functions such as Bluetooth.

In such a ubiquitous computing environment, we believe that device handoff mechanisms such as READY will become even more important.

[36] A future task is to develop a media adaptation mechanism that guarantees continuity. Media quality adjustments by device capability notification using CC/PP[13] etc. will be introduced to middleware in a form that takes continuity into consideration. In this paper, we proposed an architecture in which mobile terminals control media flow as a gateway to external devices, but we would like to make a platform for seamless mobile multimedia services by integrating with an architecture in which gateways in the network control media flows.

Cited Literature

- [1] Special Issue on IMT-2000: Standard Efforts of the ITU, 1997.
- [2] J. Haartsen. Bluetooth - the universal radio interface for ad hoc, wireless connectivity. Ericsson Review, No. 3, pp. 110-117, 1998.
- [3] J. Bacon, J. Bates, and D. Halls. Location-oriented multimedia. IEEE Personal Communications, Vol. 4, No. 5, pp. 48-57, 1997.
- [4] T. D. Hodes, R. H. Katz, E. Servan-Schreiber, and L. A. Rowe. Composable ad hoc location-based services for heterogeneous mobile clients. ACM Wireless Networks, 1998. Special issue on Mobile Computing, 1998.
- [5] J. Nakazawa, T. Okoshi, M. Mochizuki, Y. Tobe, and H. Tokuda. VNA: An object model for virtual network appliances. In IEEE International Conference on Consumer Electronics (ICCE 2000), 2000.
- [6] C.E. Perkins. IP Mobility Support, October 1996. RFC2002.
- [7] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and L. Salgarelli. IP micro-mobility support using HAWAII, July 2000. Internet Draft, draft-ietf-mobileip-hawaii-01.txt.
- [8] A. G. Valko, A. T. Campbell, and J. Gomez. Cellular IP, November 1998. Internet Draft, draft-valko-cellularip-00.txt.
- [9] Mark Stemm and Randy H. Katz. Vertical handoffs in wireless overlay networks. Mobile Networks and Applications, Vol. 3, No. 4, pp. 335-350, 1998.
- [10] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service location protocol, version 2, June 1999. Request for Comments: 2608.
- [11] W. Keith Edwards. Core Jini. Prentice Hall, Boston, 1999.
- [12] Ken Ohta, Tomohiro Nakagawa, Yoshinori Isoda, and Toshiaki Sugimura. Environment-adaptive mobile terminal architecture for realizing seamless service. IPSJ Multimedia, Distributed, Cooperative, and Mobile (DICOMO 2000) Symposium Proceedings, pp. 301-306. Information Processing Society of Japan, 2001.
- [13] Graham Klyne, Franklin Reynolds, Chris Woodrow, and Hidetaka Ohto. Composite capability/preference profiles (cc/pp): Structure and vocabularies, www.w3.org/tr, March 2001.

VERIFICATION OF TRANSLATION

The undersigned hereby declares the following:

That I am knowledgeable in Japanese and English. That I have reviewed the following:

I verify that the attached documents are an accurate translation the below referenced document:

Ken Ohta, et al.

Mobile Multimedia Middleware for Seamless Service

Mobile Computing and Wireless Communication No. 18-35

Intelligent Handoff System No. 6-35

IPSJ SIG Technical Report, Information Processing Society of Japan (2001.9.7)

All statements made of my own knowledge are true and all statements made on information and belief are believed to be true.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on: July 24, 2023



Frank McGee