PAPER

The Influence of Segmentation Mismatch on Quality of Audio-Video Transmission by Bluetooth*

Hirotsugu OKURA[†a], **Masami KATO**[†], *Members*, *and* **Shuji TASAKA**[††], *Fellow*

SUMMARY This paper examines the effect of segmentation mismatch on audio-video transmission by Bluetooth. We focus on the segmentation mismatch caused by the difference between the RFCOMM Maximum Frame Size and the baseband packet payload size. By experiment, we assessed the maximum throughput and media synchronization quality for various types of ACL packets. In the experiment, a media server transferred stored video and audio streams to a single terminal with point-to-point communication; we supposed no fading environment and added white noise by which interference from DSSS systems is modeled. The experiment showed that the effect of segmentation mismatch is large especially when the total bit rate of the two streams is near the channel transmission rate. We also observed that the media synchronization control is effective in compensating for the disturbance by the segmentation mismatch in noisy environments.

key words: *Bluetooth, segmentation mismatch, QoS, media synchronization*

1. Introduction

Bluetooth is coming into wide use to interconnect a variety of mobile terminals in a small area and/or gain wireless access to wired networks. Bluetooth is based on a *time division duplex* (*TDD*) polling scheme with a *Frequency Hopping Spread Spectrum* (*FHSS*) technique and provides two types of link: *SCO* (*Synchronous Connection–Oriented*) and *ACL* (*AsynchronousConnectionless*) [1]. The latter is a packet–switched link; therefore, Bluetooth can be an efficient means to access the Internet. As multimedia communication over the Internet becomes common, audio-video transmission on Bluetooth can become a popular application, though it was developed mainly for transmission of voice and asynchronous data.

When we use the ACL link for the transfer of continuous media such as audio and video, we are faced with various kinds of technical problems associated with packet–switching. They include the overhead of *segmentation and reassembly* (*SAR*) of packets, and disturbance of temporal relations of the media due to packet delay and delay jitter; these factors degrade the *QoS* (*Quality of Service*), in particular, *media synchronization quality*, which is a kind of application-level QoS [2].

This paper investigates the effect of SAR on the media synchronization quality of audio–video transmission over a Bluetooth link. Since Bluetooth has adopted a layered network architecture with a specific protocol stack, SAR can be performed at several places of the protocol stack. Segmentation occurs owing to the difference in the maximum treatable data size between two adjacent layers. As such maximum data sizes, Bluetooth specifies, from the top to the bottom layers, the *Maximum Frame Size* in the *RFCOMM* layer, the *Maximum Transmission Unit* (*MTU*) size in the *Logical Link Control and Adaptation Protocol* (*L2CAP*) layer, the maximum packet size of the *Host Controller Interface* (*HCI*), and the payload size in the *Baseband* layer (i.e., ACL or SCO packet payload size). If these sizes do not fit in with each other, then the transmission efficiency decreases, and the media synchronization quality deteriorates. In this paper, we focus on the RFCOMM Maximum Frame Size and the ACL packet payload size and examine how they affect the media synchronization quality.

Media synchronization for continuous media streams means the preservation of the temporal structure of *media units* (*MUs*) such as video frames and voice packets [3]. It can be classified into *intra–stream synchronization* and *inter–stream synchronization*. The former refers to the temporal constraints within a single stream, e.g., the preservation of time intervals between two successive video frames of a video stream. The latter is synchronization among multiple media streams, e.g., between a voice stream and a video stream. In order to achieve media synchronization, we need some control mechanism. A variety of studies on this subject have already been reported [4]. Among them, the *Virtual–Time Rendering* (*VTR*) algorithm [5] and the *slide control* scheme [6], [7] are effective ones that are applicable to various network environments.

We can find many researches on Bluetooth network performance in the literature [8]–[11]. Regarding the SAR issue, Das *et al.* propose two SAR policies and evaluate the throughput, end–to–end delay and link utilization on an error–free channel in [9]; thus, this study does not deal with the media synchronization quality.

In the assessment of QoS in Bluetooth networks, interference from *Direct Sequence Spread Spectrum* (*DSSS*) wireless LANs typified by IEEE 802.11 LANs is also another important issue. This is because both systems share the ISM (Industrial, Scientific and Medical) band. This interference problem is widely noticed, and many research results have already been published, e.g., see [10] through

Manuscript received April 1, 2003.
Manuscript revised December 4, 2003.
[†]The authors are with Digital Systems Development Center, SANYO Electric Co., Ltd., Gifu-ken, 503-0195 Japan.
[††]The author is with Graduate School of Engineering, Nagoya Institute of Technology, Nagoya-shi, 466-8555 Japan.
a) E-mail: ookura@gf.hm.rd.sanyo.co.jp
*This paper was presented in part at PIMRC 2002, Lisbon, Portugal.

[15]. However, the researches used performance measures only at packet–level such as throughput and delay and did not assess the media synchronization quality.

In [16], the authors report an experimental result of the media synchronization quality with single–slot packets; multi–slot packets are not treated. In that study, the issue of segmentation mismatch, which is the subject of the present paper, was not so problematic, since the payload size of the DM1 packet is small.

The purpose of this paper is to examine the effect of the segmentation mismatch and interference from DSSS systems on audio-video transmission with various types of packets (single–slot and multi–slot) in Bluetooth. We conduct an experiment of audio–video transmission between a media server and a single terminal with point-to-point communication via a Bluetooth LAN access system. In the experiment, we assume no fading environment. We apply the slide control scheme for media synchronization control and confirm its effectiveness.

The rest of the paper is organized as follows. Section 2 presents the Bluetooth LAN access system and the media synchronization scheme as the system environment supposed in our study. Section 3 describes the problem of segmentation mismatch. Section 4 illustrates the experimental methodology including the experimental system configuration and a method of the experiment. Section 5 shows experimental results. Section 6 concludes the paper.

2. System Environment

2.1 LAN Access System

Figure 1 illustrates the configuration and protocol stack of the LAN access system adopted in this paper.

The advantage of adopting Bluetooth for LAN access is easy control of the bandwidth assignment since the master controls communication with slaves by polling. The details of the polling algorithm are not specified in the standard. This means that priority control algorithms can be developed to send continuous media with higher priority. Moreover, the speed of Bluetooth communication will become larger in the standard version 2.0.

In Fig. 1, a media server is connected to a LAP (LAN Access Point) through a LAN. A DT (Data Terminal) is also connected to the LAP with PPP over a Bluetooth link. Thus, the DT can communicate with the media server with IP. Video and audio streams stored at the media server are transferred to the DT.

Bluetooth provides L2CAP and RFCOMM as the data link protocols. L2CAP manages logical links, and segmentation and reassembly of packets of the higher layer. RFCOMM is a protocol that provides a serial communication interface to the higher layer.

PPP enables us to use several kinds of network layer protocols including IP; we adopt IP in this study. In the transport layer, we can use either TCP (Transmission Control Protocol) or UDP (User Datagram Protocol).

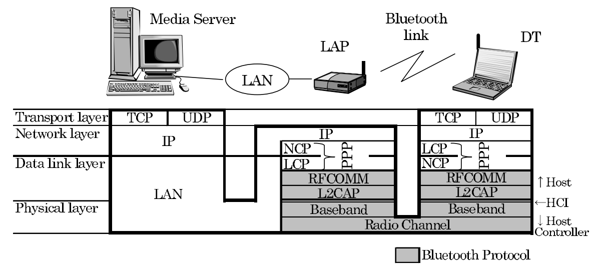

Fig. 1 Configuration and protocol stack of the LAN access system.

Table 1 Features of ACL packets.

Type	User payload size [bytes]	FEC	ARQ	Asymmetric maximum data transfer speed [kbps]
DM1	0-17	2/3rate	yes	108.8
DH1	0-27	no	yes	172.8
DM3	0-121	2/3rate	yes	387.2
DH3	0-183	no	yes	585.6
DM5	0-224	2/3rate	yes	477.8
DH5	0-339	no	yes	723.2
AUX1	0-29	no	no	185.6

As mentioned earlier, Bluetooth specifies the SCO and ACL links; we select the ACL link in this paper since it is used for LAN access. Seven kinds of packets in the ACL link are defined: DM1, DH1, DM3, DH3, DM5, DH5 and AUX1. DM and DH stand for Data–Medium rate and Data–High rate, respectively. Table 1 summarizes their features. They differ from each other in the kind of error control applied and the number of time slots occupied by a packet. In the experiment, we do not use AUX1 because it adopts neither FEC nor ARQ.

2.2 Media Synchronization Scheme

For media synchronization, we adopt the slide control scheme, which has been proposed for PHS (Personal Handy phone System) Internet access [6], [7]. There are two reasons why we adopt the slide control. One is that we can use the same control scheme for various networks with different lower layers, since media synchronization control is a kind of application-level control. The other is that we deal with stored media in this paper as well as in [6].

We define a video frame as a video MU, and a constant number of audio samples with constant bit rate constitutes an audio MU. Therefore, the size of a video MU is variable, while that of an audio MU is constant. Each MU has a timestamp in order to preserve the temporal relationship among these MUs.

Referring to Fig. 2, we define notations for specification of the slide control scheme. Let us consider a stream (say stream i), which can be either the video stream ($i = 1$) or the audio stream ($i = 2$). First, we let $T_n^{(i)}$ ($n = 1, 2, \cdots$) denote the timestamp of the n-th MU in stream i, and define $\sigma_{n,n+1}^{(i)} = T_{n+1}^{(i)} - T_n^{(i)}$. We also suppose that the n-th MU in

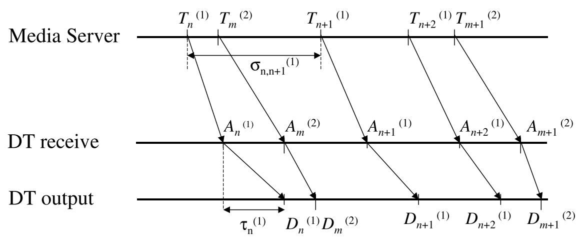

Fig. 2 Temporal relationship of MUs.

stream i arrives at the DT at time $A_n^{(i)}$ and is output at time $D_n^{(i)}$; this implies that we set the buffering time for absorbing delay jitter to $\tau_n^{(i)} = D_n^{(i)} - A_n^{(i)}$.

We now define the ideal target output time $x_n^{(i)}$ of the n-th MU in stream i as

$$x_1^{(i)} = A_1^{(i)} + \tau_1^{(i)}$$
$$x_n^{(i)} = x_{n-1}^{(i)} + \sigma_{n-1,n}^{(i)} \quad (n = 2, 3, \cdots).$$

If there were no network delay jitters, we should have $D_n^{(i)} = x_n^{(i)}$ for all values of n. In reality, however, there exist the jitters; therefore, we cannot always output the MU at the ideal target output time. In order to cope with this situation, the VTR (virtual-time rendering) algorithm, which the slide control scheme utilizes, introduces the target output time $t_n^{(i)}$, which is obtained by adding a delay (i.e., slide time) to the ideal target output time. In other words, in addition to the actual time, we introduce a virtual-time which extends according to the amount of delay jitters of MUs received at the DT, and media are rendered along the virtual-time axis [6]. The slide control scheme slides the virtual-time axis adaptively according to the amount of delay jitter of MUs received.

Because of space limitations in this paper, we omit the details of the algorithm for the slide control, which can be found in [6].

3. Segmentation Mismatch

3.1 HCI Flow Control

The HCI (Host Controller Interface) flow control is exercised between the Bluetooth Host (e.g., a PC) and the Bluetooth Host Controller (e.g., a Bluetooth module) as shown in Fig. 1. Data are transferred from the L2CAP layer to the Baseband layer through the HCI as HCI Data Packets.

On initialization of the flow control, the Host issues the *Read_Buffer_Size* command, which has two return parameters. One is *HC_ACL_Data_Packet_Length*, which determines the maximum size of an HCI Data Packet that can be sent from the Host to the Host Controller. The other is *HC_Total_Num_ACL_Data_Packets*, which specifies the total number of HCI Data Packets that the Host Controller can save in its buffers for waiting for transmission. The values depend on the implementation by the manufacturer or versions of the firmware. For examples, the values of HC_ACL_Data_Packet_Length and HC_Total_Num_ACL_Data_Packets are 800 bytes and 10 packets, respectively, in the Bluetooth module ROK 101 007 of Ericsson Bluetooth Starter Kit(LPY111243), which we used in our experiment.

Now let N denote the number of packets that the Host can transmit at a time. The initial value of N is set to HC_Total_Num_ACL_Data_Packets.

Once a connection is established to another device, the Host Controller uses the event of *Number Of Completed Packets* to control the flow of data from the Host. This event contains the number of HCI Data Packets that have been transmitted since the previous return of the event.

Every time the Host has sent an HCI Data Packet, it must assume that the free buffer space in the Host Controller has decreased by one HCI Data Packet ($N \rightarrow N - 1$).

When the Host receives a new event of Number Of Completed Packets, the Host gets information about how much the buffer usage has decreased since the previous return of the event. Let M denote the number of completed packets in the event. It can then calculate the current free buffer space as $N \rightarrow N + M$. The flow is controlled according to this free buffer space. For examples, when N reaches 0, the Host cannot send data to the Host Controller.

As mentioned above, the standard prescribes the HCI flow control not on the basis of the quantity of data, but on the basis of the number of packets. This means that the transmission efficiency decreases if the Host sends data whose size is less than HC_ACL_Data_Packet_Length. However, this is often the case. We will explain this problem in more detail in the next subsection.

3.2 RFCOMM Maximum Frame Size

The RFCOMM protocol has a parameter called *Maximum Frame Size*, which is referred to as RFCOMM Maximum Frame Size in this paper. If the data size exceeds the RFCOMM Maximum Frame Size, then the RFCOMM layer divides the data into two or more data within the size. This value is negotiated between local and remote RFCOMM layers before creation of the data link connection on an RFCOMM session. This negotiation is performed by the PN (Parameter Negotiation) command prescribed in TS 07.10 [17], on which RFCOMM is based.

The default value of the RFCOMM Maximum Frame Size is 127 bytes in the standard. It is much less than the default value of the L2CAP MTU size, which is 672 bytes. When the RFCOMM Maximum Frame Size is less than the L2CAP MTU size and HC_ACL_Data_Packet_Length of the HCI flow control, the RFCOMM Maximum Frame Size dominates the amount of data which can be carried in the payload of a baseband packet. For example, consider a DH5 packet, whose maximum payload size is 339 bytes. Then, if the RFCOMM Maximum Frame Size is set to the default value, the actual payload size of the DH5 packet is at most the sum of 127 bytes, the header length of an L2CAP packet and that of an HCI packet. Since the DH5 packet occupies five slots regardless of the amount of data that it actually carries, the transmission of data is inefficient.

As another example, take a DM3 packet, whose maximum payload size is 121 bytes. Then, with the default value of the RFCOMM Maximum Frame Size, a small number of byte may be left for the second packet, which also occupies three slots.

Thus, we see that the mismatch occurs either when the value of the RFCOMM Maximum Frame Size is much smaller than the payload size of a baseband packet or when the value is slightly larger than the payload size or its multiple. We refer to the former case as *Mismatch Pattern I*, and the latter as *Mismatch Pattern II*. These two types of the segmentation mismatch will decrease the throughput and will affect the media synchronization quality.

4. Experimental Methodology

4.1 Experimental System

We developed an experimental system shown in Fig. 3; it is composed of a media server, a LAN access point, a simulated wireless environment and a DT part.

The media server (Celeron 700 MHz, Windows2000) stores an H.263 video file and the corresponding G.726 audio file; it sends out the video and audio streams to the DT on demand by UDP. The media server is connected to the LAN access point by a 10BASE-T Ethernet. It is often the case that a LAN is shared by servers and many terminals, which generate interference traffic for each other. For simplicity of experiment, however, we have adopted a simple configuration of the experimental system in Fig. 3. The influence of the interference traffic should be examined at the next step of our study.

The LAN access point consists of a PC part and a Bluetooth unit. The PC part manages link control and data transmission by using HCI commands. The OS of the PC (Pentium III 1GHz) is Linux 2.2.16. A PCI high speed serial board (AccelePort 2r 920 by Digi International) has been installed into the PC. This enables serial port communication up to 460.8 kbps. Size of the buffer which is used by PPP to write data into the serial device has been changed from 256 bytes to 806 bytes. Thus, a PPP frame of up to 800 bytes is passed to the lower layer, RFCOMM, without segmentation.

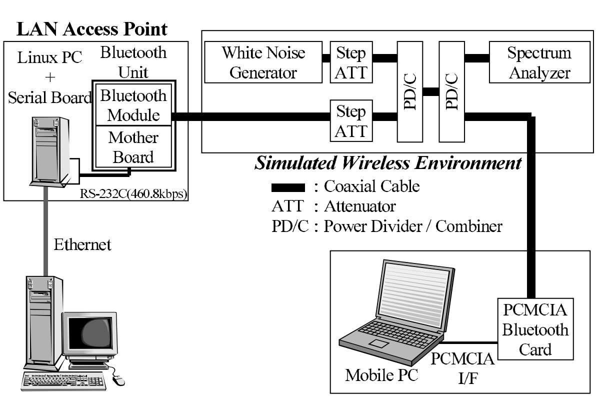

Fig. 3 The experimental system.

The Bluetooth unit consists of a Bluetooth module (ERICSSON ROK 101 007) and a high speed serial communication mother board. The mother board offers an interface between the UART(Universal Asynchronous Receiver/ Transmitter) and the serial communication port of the PC. The PC part and the Bluetooth unit are connected to each other with the RS-232C serial interface, whose speed is 460.8 kbps.

The simulated wireless environment comprises a white noise generator, attenuators, power dividers/combiners, and a spectrum analyzer [16]. Note that the white noise can be regarded as an aggregate interference signal from DSSS systems such as wireless LANs. The influence of other interference signals due to frequency hopping spread spectrum systems is one of our future studies. In this environment, we assume a radio channel without fading. The effects of fading are also for future study.

The DT is a mobile PC (Pentium II 366 MHz, Windows 98) with a PCMCIA Bluetooth card.

4.2 Procedures and Conditions of the Experiment

(1) Throughput and media synchronization experiments

We first measure the maximum throughput for each ACL packet at the application layer. The measurement is carried out for various values of the RFCOMM Maximum Frame Size and six packet types (namely, DM1, DH1, DM3, DH3, DM5 and DH5) with and without the white noise. The target coding rate of video is set to 300 kbps; this rate enables the measurement of the maximum throughput in the range of the RFCOMM Maximum Frame Size that we consider. The white noise is added so that the CNR is 18.5 dB and 19.3 dB; by a preliminary measurement, we have determined the former value of CNR below which the throughput with the DM5 packet decreases; the latter value has been selected so that the throughput with the DH5 is between the throughput without the white noise and that for CNR of 18.5 dB. These values are practical ones. Note that the maximum data transfer speed of the DM5 packets is the highest of the three types of DM packets, which adopt the FEC.

Next, we measure the media synchronization quality, using a video stream with a target coding rate of 125 kbps. This rate was selected as the maximum value below which no packet loss occurs with the DM3, DH3, DM5 and DH5 packets when RFCOMM Maximum Frame Size is 127 bytes, which is the default.

(2) Specifications of video and audio

Specifications of the video and audio used in the experiment are shown in Table 2.

A video frame (i.e., a picture) is defined as a video MU. The size of a picture is sub-QCIF, i.e., 128×96 pixels. We selected the target coding rates of video as mentioned

Table 2 Specifications of video and audio.

	Video	Audio
Compression scheme	H.263	G.726
Size of picture	Sub-QCIF (128×96)	-
Target coding rate [kbps]	300 (to measure maximum throughput) 125 (to measure media synchronization quality)	24
Original recording time [s]	120.8	120.7
MU rate [MU/s]	15	10
Media source	Head view of a speaker and his voice	

earlier, and the target MU rate is 15 MU/s. The generation time $T_n^{(1)}$ of the n-th video MU is given by the corresponding Temporal Reference (TR_n) of H.263 as follows: $T_1^{(1)} = TR_1 = 0$ and $T_n^{(1)} = TR_n/30; (n > 2)$.

The audio coding rate is 24 kbps, and the size of an audio MU is 2.4 kbits. This leads to an audio MU rate of 10 MU/s. The generation time $T_m^{(2)}$ of the m-th audio MU is given by $T_m^{(2)} = (m-1)/10$.

(3) Parameters of slide control

We use the same parameter values for the media synchronization control as those in [7].

We establish the origin of time for media synchronization at the arrival time of the first video MU $A_1^{(1)}$. We then set $\tau_1^{(1)} = 500$ ms.

4.3 Performance and Quality Measures

We adopt the maximum throughput at the application layer as a measure of transmission efficiency.

We also assess the media synchronization quality, which is measured by the mean square error of intra–stream synchronization and that of inter–stream synchronization. The error of intra–stream synchronization of the n-th MU in stream $i (i = 1, 2)$ is defined as $\Delta_n^{(i)} = D_n^{(i)} - t_n^{(i)}$. We also define the error of inter–stream synchronization between the n-th MU in stream 1 and the corresponding MU in stream 2 (say the m-th MU) as $\Delta_n^{(1-2)} = (D_n^{(1)} - D_m^{(2)}) - (T_n^{(1)} - T_m^{(2)}), (n \geq 1, m \geq 1)$. As measures of the synchronization quality, we use the mean square of $\Delta_n^{(i)}$ with respect to n for i=1 and 2, and that of $\Delta_n^{(1-2)}$ with respect to n. The corresponding relation between n and m in $\Delta_n^{(1-2)}$ is determined as follows. The MU rate of video is usually different from that of audio. Therefore, for a given video MU (say n-th one), we select the audio MU that is first generated after the video MU as the corresponding m-th audio MU. For example, in Fig. 2, the $(m+1)$-st audio MU corresponds to both the $(n+1)$-st video MU and the $(n+2)$-nd video MU.

The total output time is another important measure of synchronization quality; it is the necessary time for the video or audio file to be output at the DT. It should be noted that this is not necessarily equal to the original recording time at the server and usually becomes longer than it because of the network delays and the associated changes of the target output time.

5. Experimental Result

5.1 Maximum Throughput

We first examine the maximum throughput. The throughput here is the sum of video throughput and audio one. Figure 4 shows the maximum throughput in kbps versus the RFCOMM Maximum Frame Size in bytes without the white noise. Figures 5 and 6 are the results when the white noise is added so that the values of the CNR are 19.3 dB and 18.5 dB, respectively.

In Fig. 4, we see that the maximum throughput largely varies with the RFCOMM Maximum Frame Size. In the case of DH5, for instance, the maximum throughput at the default value, 127 bytes, of the RFCOMM Maximum Frame Size is approximately 62% of that when the size is 250 bytes. This corresponds to Mismatch Pattern I. Thus, we find that supporting only the default value of the RFCOMM Maximum Frame Size is not sufficient to attain the best performance that the Bluetooth Baseband layer offers.

As we observe in Fig. 4, in the case of multi–slot packets (DM3, DH3, DM5 and DH5), the maximum throughput drops at some values of the RFCOMM Maximum Frame Size; it does not monotonically increase in proportion to the RFCOMM Maximum Frame Size. In the case of DM3, for example, the maximum throughput drops at 127 bytes and 240 bytes. In the case of DH3, it drops at 180 bytes. These cases correspond to Mismatch Pattern II.

Figure 4 also indicates that when the value of the RFCOMM Maximum Frame Size is smaller than 127 bytes, the packet type that has the largest maximum throughput in the multi–slot packet types is not DH5, though it supports the highest data transfer speed as shown in Table 1; DM3 and DH3 have larger throughput than DH5. If the RFCOMM Maximum Frame Size is very small, then the data divided at RFCOMM can be transmitted by a single baseband packet of either DM3, DH3, DM5 or DH5. Because DM3 and DH3 can send the data in a shorter time (3 slots) than that for DH5 (5 slots), DM3 and DH3 have larger throughput than DH5.

Figures 5 and 6 also show the same tendency as that of Fig. 4 with respect to the dependence on the RFCOMM Maximum Frame Size in each packet type.

Comparing Figs. 4, 5 and 6, we notice that the maximum throughput of DH packets with the white noise decreases largely as the CNR becomes small, while the DM

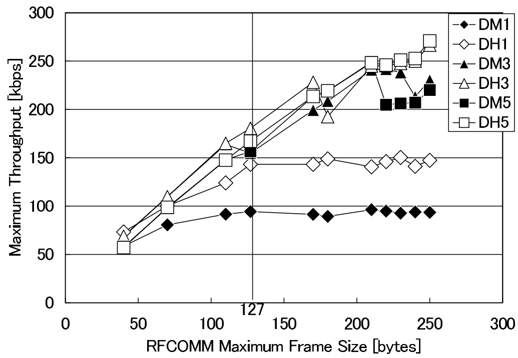

Fig. 4 Maximum throughput. (no white noise)

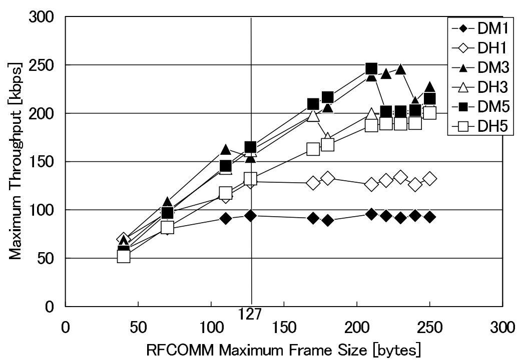

Fig. 5 Maximum throughput. (CNR=19.3 dB)

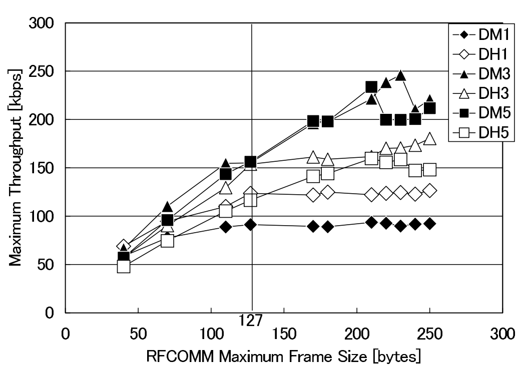

Fig. 6 Maximum throughput. (CNR=18.5 dB)

packets are scarcely affected. This is due to FEC of the DM packets. DH5 packet, which occupies the most slots, is most affected by the white noise, while DH1 has the least effect. This is because a multi–slot packet takes more time than a single–slot packet for its retransmission.

5.2 Mean Square Error of Media Synchronization

Let us examine the mean square error of intra–stream and inter–stream media synchronization by referring to Figs. 7 through 10. Each figure displays the measurement results of the system with slide control and those of the system with no control, which are denoted by "SC" and "NC," respectively. The results for DH1 and DM1 are not shown in

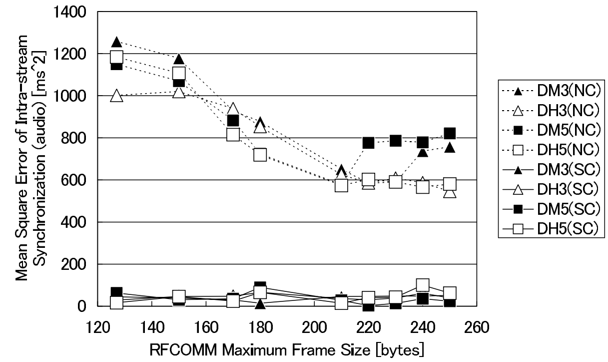

Fig. 7 Mean square error of intra–stream media synchronization of audio. (no white noise)

the figures because their maximum throughput is hardly affected by the RFCOMM Maximum Frame Size in the range over 127 bytes.

(1) Mean square error of intra–stream synchronization

We first consider the case without the white noise. Figure 7 plots the mean square error of intra–stream synchronization for audio in ms^2 versus the RFCOMM Maximum Frame Size in bytes. We first note that NC in Fig. 7 exhibits similar behavior to that in Fig. 4 with respect to the dependence on the RFCOMM Maximum Frame Size in each packet type. This implies that the segmentation mismatch affects the intra–stream synchronization quality of audio. Thus, we see that regarding the audio quality with NC, the influence of the RFCOMM Maximum Frame Size is larger than that of the packet types.

In subjective assessment, however, we hardly noticed the difference in the quality among the packet types since the subjective quality is not low for all the types. This is because the maximum throughputs for the multi–slot packets (see Fig. 4) are larger than the total bit rate of the audio and video (i.e., 149 kbps) when the RFCOMM Maximum Frame Size is larger than 127 bytes.

Regarding the audio quality with SC, we hardly notice the effect of RFCOMM Maximum Frame Size. This implies that the effect is absorbed by the slide control. This is because the slide control postpones the target output time $t_n^{(i)}$ when delay jitter becomes larger. In other words, it increases the buffering time of MUs.

Figure 8 shows the mean square error of intra–stream synchronization for audio versus the RFCOMM Maximum Frame Size. In this figure, empty symbols, light gray symbols and dark gray symbols denote the results when the white noise is not added, those when the CNR is 19.3 dB and those when the CNR is 18.5 dB, respectively. The result for DM packets are not shown since the mean square error of intra–stream synchronization was not much changed by the CNR.

Let us examine the results for NC. Then, we find that the intra–stream synchronization quality becomes worse as the CNR becomes lower. In particular, the effect of the CNR on DH5 is remarkable. A DH5 packet uses five slots, whose

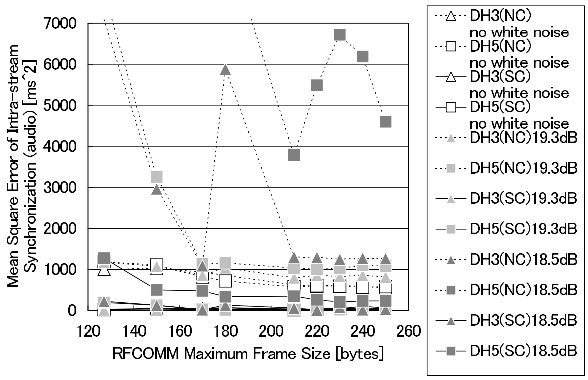

Fig. 8 Mean square error of intra–stream media synchronization of audio.

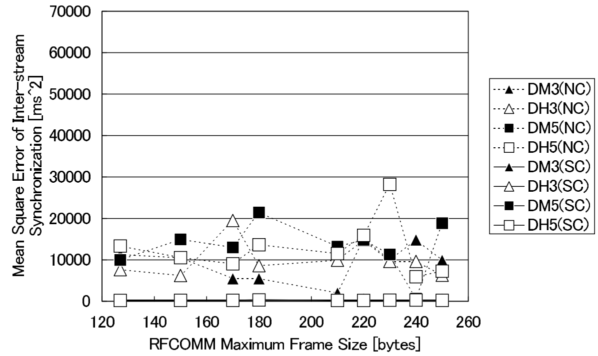

Fig. 9 Mean square error of inter–stream media synchronization. (no white noise)

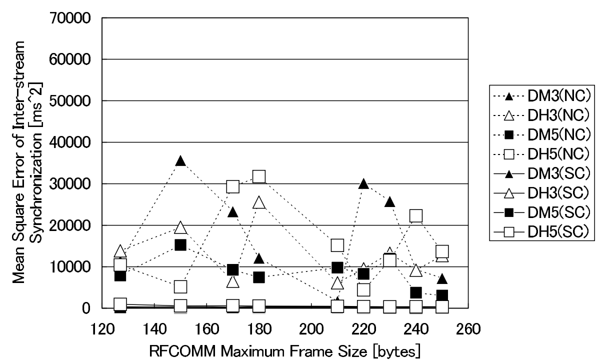

Fig. 10 Mean square error of inter–stream media synchronization. (CNR=18.5 dB)

number is more than that of a DH3 packet. For this reason, DH5 packet takes longer time to be retransmitted when CNR is low. This implies that delay and delay jitter of a DH5 packet are larger than those of a DH3 packet. Hence, disturbance of temporal structure of MUs when DH5 packets are used becomes much larger than that when DH3 packets are used as the CNR becomes lower.

We also see that the mean square error of intra–stream synchronization for DH3 with NC (CNR = 18.5 dB) has a local peak at 180 byte of the RFCOMM Maximum Frame Size. At this value of the RFCOMM maximum Frame Size, the maximum throughput for DH3 drops slightly as seen from Fig. 6. However, large influence on the intra–stream synchronization quality appears. Also, the mean square error of intra–stream synchronization for DH5 with NC (CNR = 19.3 dB) when the RFCOMM Maximum Size is smaller than 170 bytes increases steeply as the value of RFCOMM Maximum Frame Size decreases. These two observations are due to the same reason: the target coding rate is very close to or larger than the maximum throughput at these values of the RFCOMM Maximum Frame Size.

Regarding SC, we observe that the slide control is effective in improving the quality of intra–stream synchronization for DH3 and DH5. We can see only small effect of RFCOMM Maximum Frame Size even for DH5 when CNR is 18.5 dB.

(2) Mean square error of inter–stream media synchronization

Figures 9 and 10 plot the mean square error of inter–stream synchronization without the white noise and with the white noise (CNR = 18.5 dB), respectively.

At first, we consider the case when the white noise is not added. We see that inter–stream synchronization is affected by the RFCOMM Maximum Frame Size. However, even for NC, most of the error values are below 25600 (= 160^2) ms^2, which means not asynchrony [18].

Comparing Figs. 9 and 10, we then find that the errors for NC in Fig. 10 tend to be larger than those in Fig. 9. That is, the inter–stream synchronization quality degrades when the white noise is added.

Regarding SC in Figs. 9 and 10, we see that the slide control is effective in improving the quality of inter–stream synchronization.

5.3 Total Output Time

Figures 11 and 12 show the total output time of audio for DM5 and DH5 without the white noise and that with the white noise, respectively. The total output time of video is not shown here because it has the same tendency as that of audio. The total output time when three–slot packets are used is not shown because of the same reason.

First, let us consider NC of Fig. 11. Since the state of the Bluetooth radio channel is fine, retransmission does not take place so much. Therefore, the total output time is almost equal to the recording time.

Next, we examine NC of Fig. 12. The total output time of DH5 is much larger than that of Fig. 11. This is because retransmission occurs many times since FEC is not applied to DH packets.

Comparing NC and SC in Figs. 12 and 11, we see that the total output time of SC is larger than that of NC. The value of SC must be larger than that of NC because the slide control scheme postpones the target output time when an MU arrives at the DT later than a threshold. At the expense of the total output time, we get higher media synchronization quality, which we have observed in Figs. 7 through 10

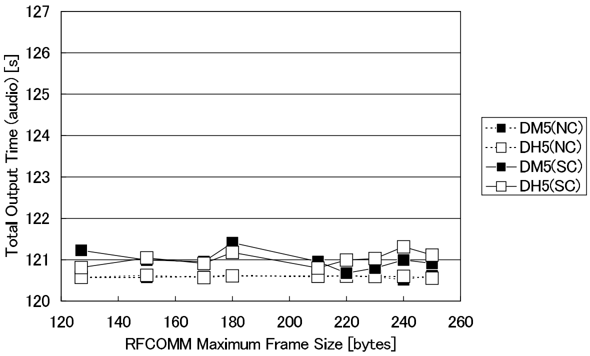

Fig. 11 Total output time of audio. (no white noise)

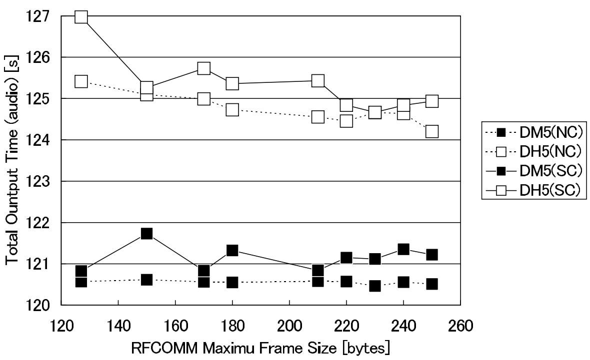

Fig. 12 Total output time of audio. (CNR=18.5 dB)

when white noise is added. The increase of the total output time due to the slide control can affect the subjective quality. In the experiment in this paper, however, the increase is at most two seconds compared to the original recording time of 120 seconds. It is neglectable for stored media transmission. The permissible range of the increase from the subjective point of view is not clear. It is left for our future study.

6. Conclusions

We examined the influence of the segmentation mismatch on the maximum throughput and media synchronization quality of audio-video transmission on the Bluetooth LAN access system under the interference from DSSS systems.

We first observed that if the RFCOMM Maximum Frame Size is set to the default value, 127 bytes, the maximum throughput is smaller than that for its larger values. This means that the default value of the RFCOMM Maximum Frame Size is not sufficient to utilize the high speed communication capability that the Bluetooth Baseband layer can provide. We also noticed that the packet type with the largest maximum throughput depends on the RFCOMM Maximum Frame Size.

We next found that the RFCOMM Maximum Frame Size more affects the media synchronization quality than the packet types when no media synchronization control is exerted. Especially when the bit rate of the stream is close to the maximum throughput, the influence of the segmentation mismatch appears largely.

Thus, we know that the initial value of the RFCOMM Maximum Frame Size should be negotiated by taking into consideration the given network condition. However, the RFCOMM Maximum Frame Size decided this way may be unsuitable when the quality of the radio link changes. We saw this in the experiment with the white noise. One of the solutions is to select the best packet type adaptively under the given condition. This is included in our future work.

The media synchronization control is effective especially when the white noise is added. We see that media synchronization quality becomes higher by the slide control scheme at the expense of the total output time.

The influence of FH interference, other terminals in the LAN, and other ACL links in the piconet are left as future subjects to be studied.

Acknowledgment

The authors thank Johnson Bao for his assistance in the experiment. This work was supported in part by the Grant-In-aid for Scientific Research of Japan Society for the Promotion of Science under Grant 14350200.

References

[1] Bluetooth SIG, "Specification of the Bluetooth System version 1.1," Feb. 2001.
[2] S. Tasaka and Y. Ishibashi, "Mutually compensatory property of multimedia QoS," Conf. Rec. ICC2002, pp.1105–1111, April/May 2002.
[3] G. Blakowski and R. Steinmetz, "A media synchronization survey: Reference model, specification, and case studies," IEEE J. Sel. Areas Commun., vol.14, no.1, pp.5–35, Jan. 1996.
[4] Y. Ishibashi and S. Tasaka, "A comparative survey of synchronization algorithms for continuous media in network environments," Proc. LCN2000, pp.337–348, Nov. 2000.
[5] Y. Ishibashi and S. Tasaka, "A synchronization mechanism for continuous media in multimedia communications," Proc. INFOCOM'95, pp.1010–1019, April 1995.
[6] S. Tasaka, M. Kato, and K. Nakamura, "Stored media synchronization in PHS Internet access," Conf. Rec. GLOBECOM'98, pp.113–119, Nov. 1998.
[7] M. Kato, S. Tasaka, and K. Nakamura, "A comparison of media synchronization quality between TCP and UDP in PHS Internet access," Proc. PIMRC'99, pp.1092–1097, Sept. 1999.
[8] S. Zürbes, W. Stahl, K. Matheus, and J. Haartsen, "Radio network performance of Bluetooth," Conf. Rec. ICC 2000, pp.1563–1567, June 2000.
[9] A. Das, A. Ghose, A. Razdan, H. Saran, and R. Shorey, "Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network," Proc. INFOCOM 2001, April 2001.
[10] J. Lansford, A. Stephens, and R. Nevo, "Wi-Fi(802.11b) and Bluetooth: Enabling coexistence," IEEE Netw., vol.15 no.5, Sept.–Oct. 2001.
[11] D. Famolari, "Link performance of an embedded Bluetooth personal area network," Conf. Rec. ICC 2001, June 2001.
[12] J. Zyren, "Reliability of IEEE 802.11 hi rate DSSS WLANs in a high density Bluetooth environment," Proc. ICT 2001, vol.1, pp.117–127, June 2001.
[13] M.V.S. Chandrashekhar, P. Choi, K. Maver, R. Sleber, and K. Pahlavan, "Evaluation of interference between IEEE 802.11b and Bluetooth in a typical office environment," Proc. PIMRC 2001, pp.71–75, Sept. 2001.

[14] R.J. Punnoose, R.S. Tseng, and D.D. Stancil, "Experimental results for interference between Bluetooth and IEEE 802.11b DSSS systems," Conf. Rec. VTC 2001, vol.1, pp.67–71, Oct. 2001.
[15] K. Matheus and S. Zürbes, "Co-existence of Bluetooth and IEEE 802.11b WLANs: Results from a radio network testbed," Proc. PIMRC 2002, pp.151–155, Sept. 2002.
[16] H. Okura, M. Kato, and S. Tasaka, "A media synchronization experiment on continuous media transmission in Bluetooth LAN access," Proc. PIMRC 2001, pp.D-64–D-70, Sept. 2001.
[17] "TS 07.10, Ver 6.3.0," ETSI.
[18] R. Steinmetz, "Human perception of jitter and media synchronization," IEEE J. Sel. Areas Commun., vol.14, no.1, pp.61–72, Jan. 1996.

Hirotsugu Okura received the B.S. degree in electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1986, the M.S. degree in electrical and computer engineering from the Institute in 1989. He joined Sanyo Electric Co., Ltd. in 1989, and has been engaging in the development of mobile multimedia communication systems. He is presently a Senior Staff in Digital Systems Development Center of Sanyo. He is a member of the IEEE.

Masami Kato received the B.S. degree in physics from Nagoya University, Nagoya, Japan, in 1984, and the Ph.D. degree in electrical and computer engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1999. He joined Sanyo Electric Co., Ltd. in 1984, and engaged in the development of mobile multimedia communication systems. He is presently a manager in Digital Systems Development Center of Sanyo.

Shuji Tasaka received the B.S. degree in electrical engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1971, and the M.S. and Ph.D. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1973 and 1976, respectively. Since April 1976, he has been with Nagoya Institute of Technology, where he is now a Professor in the Department of Computer Science and Engineering. In the 1984–1985 academic year, he was a Visiting Scholar in the Department of Electrical Engineering at the University of California, Los Angeles. His current research interests include wireless networks, multimedia QoS, and multimedia communication protocols. He is the author of a book entitled *Performance analysis of Multiple Access Protocols* (Cambridge, MA: The MIT Press, 1986). Dr. Tasaka is a member of the IEEE, ACM, and Information Processing Society of Japan.