PAPER

TCP versus UDP for Media Synchronization in PHS Internet Access*

Shuji TASAKA†, Masami KATO††, *and* Kotaro NAKAMURA†**, *Members*

SUMMARY A performance comparison between TCP and UDP in PHS Internet access is made by experiment from a media synchronization point of view. We consider a situation where PHS mobile terminals access H.263 video and G.726 audio stored at a media server by a streaming method. PIAFS is adopted as the data link protocol for the PHS wireless channels. We examined how white noise and Rayleigh fading on the PHS channel as well as the Internet traffic affect the performance. For the comparison, we evaluated several performance measures such as the coefficient of variation of output interval, and found that UDP outperforms TCP in almost all cases.

key words: *PHS, media synchronization, Internet, TCP, UDP*

1. Introduction

The Internet access is an explosively growing demand in the telecommunication services. In particular, wireless access for mobile computing is one of the booming services which are expected to be offered at higher speeds and lower cost. However, most of the currently available wireless networks do not meet the requirement sufficiently; wireless LANs provide high speed access only within limited geographical areas and also support low mobility, while cellular phone systems can cover wide areas with high mobility but their access speeds are not high*1. *PHS* (*Personal Handy Phone System*), which is a 1.9 GHz band microcellular digital cordless telephone system [1], is one of the promising access methods currently available; at present, it offers a 32 kb/s bearer service in both indoor and outdoor areas, and a 64 kb/s service has begun to be supported. It allows medium mobility, at least walking speeds.

PHS is based on TDMA/TDD where one carrier provides four duplex bearer channels of 32 kb/s each. Although major usage of PHS now is a speech transmission of 32 kb/s ADPCM, it is more suitable for digital data transmission because of its operational principle. We can expect the growing number of PHS Internet access users.

Currently, the great majority of the Internet access is concerned with multimedia; access only to text data is minor. Among the multimedia applications, access to continuous media such as video and audio is rapidly increasing. The traditional way of handling this type of application is to download the file of video and audio to our computer and then play it. However, the volume of the file is usually large, and therefore this method is time-consuming. In addition, it is not applicable to live media. In order to overcome the difficulty, the streaming method can be used: *Streaming multimedia* means that data is being sent to a user's computer and displayed as it is being sent [2]. This method began to be introduced a few years ago, and it has become quite popular*2. This paper considers the streaming multimedia over the Internet, especially playback of video and audio stored at a multimedia server. A typical example of the applications for PHS is information retrieval with portable computers by people out of their offices / home*3.

In order to obtain video and audio of high quality by the streaming method, care must be taken about the preservation of the temporal relations among media streams, which is referred to as *media synchronization* [3]. *Lip synch*, which adjusts the output timing between spoken voice and the movement of the speaker's lips, is a typical example. In general, the temporal relations of continuous media are classified into *intra-stream synchronization* and *inter-stream synchronization*. The former is to output *media units* (MUs)*4 of a single stream at the destination at the same intervals as the generation ones at the source, while the latter is synchronization among plural media streams, e.g., between a voice stream and a video stream. Lip synch needs both kinds of control. Note that the temporal relations can be disturbed by various causes including media capturing process and network delay jitters.

Many of existing audio and video tools for the Internet do not provide support for lip synch*5 [4]. The MBone audio

Manuscript received February 8, 1999.
Manuscript revised September 20, 1999.
† The authors are with the Dept. of Electrical and Computer Engineering, Nagoya Institute of Technology, Nagoya-shi, 466-8555 Japan.
†† The author is with SANYO Electric Co., Ltd., Gifu-ken, 503-0195 Japan.
* This paper was presented in part at PIMRC'99, Osaka, Japan.
** Presently, with OMRON Corporation.

*1 For example, the Cellular Digital Packet Data (CDPD), which takes advantage of the idle time on the analog AMPS channel, transmits packet data at a rate of 19.2 kb/s.

*2 Examples of the streaming video player, for instance, include RealVideo, StreamWorks, VivoActive, VDOLive and NetShow [2].

*3 The purpose includes, for instance, acquisition of knowledge for problem solving and presentation of sales information to customers.

*4 This is the information unit for media synchronization; an example is a video frame.

*5 Some of the streaming multimedia players seem to provide the support; however, we cannot know the details because of the proprietary products.

and video transmission with vat and vic, for instance, performs only intra-stream synchronization of audio with vat; neither intra-stream synchronization of video nor inter-stream synchronization is supported. In [4], a scheme with both control functions is proposed. Also, Ishibashi and Tasaka have proposed a lip synchronization scheme of streaming type [5], which is called the *virtual-time rendering* (*VTR*) algorithm [6], and implemented it on several experimental systems based on packet switching using the Internet protocol suite [6]-[11]. The authors have further applied a sophisticated version of the VTR algorithm called *slide control* [12] to a PHS Internet access system and evaluated the performance by experiment [13], [14].

The present work is continuation of our research efforts about the media synchronization issue in PHS Internet access. The experimental system in [13] utilizes UDP as the transport protocol, and therefore we have observed much MU loss at heavy data loads on the Internet and even at medium loads when the fading frequency on the PHS access channel is high. TCP, which has a retransmission capability of MUs, can recover the loss but may incur another problem of media synchronization, namely, increase in delay jitters. The application of TCP and its comparison with the case of UDP was left as a future work, which is the subject of this paper.

The remainder of the paper is organized as follows. Section 2 describes the system environment of PHS Internet access we consider here. Section 3 specifies the slide control scheme. Section 4 presents the experimental methodology. Section 5 shows and discusses experimental results.

2. System Environment

In Fig. 1, we show the system environment studied in this paper. PHS mobile terminals access a media server connected to the Internet and play out video and the corresponding audio by the streaming method. We suppose *ITU-T H.263* video [15] and *G.726* audio [16]. These types of video and audio were selected because of their coding rates suitable for the PHS transmission rate (i.e., 32 kb/s). In the beginning, the PHS mobile terminal calls a dial-up router connected to the Internet using the *PPP* (*Point-to-Point Protocol*). Then, a physical link is set up between the terminal and the dial-up router, and a data link control protocol specified by the PHS Internet Access Forum, namely, *PIAFS* [17], is applied to

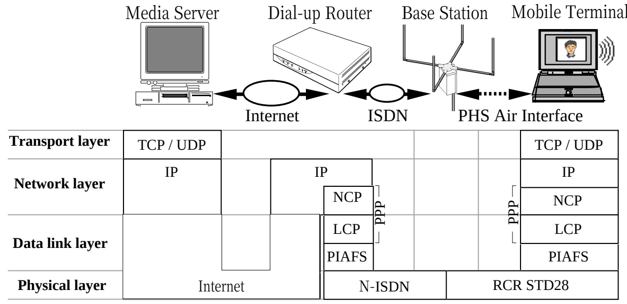


Fig. 1 Internet Access via PHS channels.

the link; it carries out a kind of selective repeat ARQ control[†]. Using the PPP, the dial-up router assigns an IP address to the terminal, which can thus communicate with the server as an Internet host. Either TCP or UDP is used as the transport protocol between the terminal and the server.

The video and audio are transferred over the Internet from the media server to the dial-up router in a packet-switching mode. The two media streams are then conveyed over the N-ISDN and the PHS channel in the circuit-switching mode. Note that the temporal relations of the video and audio can be disturbed by retransmission by PIAFS on the PHS channels as well as delay jitters over the Internet. When TCP is employed, the end-to-end retransmission of MUs also causes delay jitters.

3. Slide Control Scheme

For media synchronization, we use an extended version of the *slide control*. In this paper, a video frame is defined as a video MU, and a constant number of audio samples 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. The *TR* (Temporal Reference for pictures) in an H.263 video stream can be utilized as timestamps. The server sends out each MU according to its timestamp.

In order to specify the slide control scheme, we consider a stream (say stream i), which can be either the video stream (i = 1) or the audio stream (i = 2). For stream i, we define the following notations [5]. 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 stream i arrives at the terminal at time $A_n^{(i)}$ and is output at time $D_n^{(i)}$; this implies that we have the *output waiting time*[††] given by $\tau_n^{(i)} = D_n^{(i)} - A_n^{(i)}$.

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

$$x_1^{(i)} = A_1^{(i)} + \tau_1^{(i)} \tag{1}$$

$$x_n^{(i)} = x_{n-1}^{(i)} + \sigma_{n-1,n}^{(i)} \qquad (n = 2, 3, \cdots). \tag{2}$$

If there were no network delay jitter, 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 virtual-time rendering algorithm introduces the *target*

[†] H.263 video is based on interframe coding, which is very vulnerable to data error and loss. PIAFS can be used for the recovery from the error and loss on the PHS access channel.

[††] Let $\Delta_{max}^{(i)}$ and $\Delta_{min}^{(i)}$ be the maximum and minimum values, respectively, of the network delay in seconds for stream i. Then, the intra-stream synchronization can be achieved by choosing the output waiting time so as to satisfy $\tau_1^{(i)} \geq \Delta_{max}^{(i)} - \Delta_{min}^{(i)}$ [18]. However, it is not so easy to know the exact values of $\Delta_{max}^{(i)}$ and $\Delta_{min}^{(i)}$. Setting $\tau_1^{(i)}$ to the transmission time of the whole file is the downloading method, which usually results in a terribly long waiting time.

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 terminal, and media are rendered along the virtual-time axis [5], [6], [9]. Note that the introduction of the slide time leads to an increase in the playback time of the media stream. However, since we are dealing with stored media in this paper, the real-time requirement is not so severe†. We can obtain the improved quality of media synchronization at the expense of the increased playback time. Unless the increment of the playback time is perceptible, it does not matter.

Now, let us define the slide time of the n-th MU in stream i, which is denoted by $\Delta S_n^{(i)}$, as the difference between the original target output time $t_n^{(i)}$ and the modified one. Also, we denote the *total slide time* by $S_n^{(i)} = S_{n-1}^{(i)} + \Delta S_n^{(i)}$ ($n = 1, 2, \cdots$; $S_0^{(i)} = 0$). Then, $t_n^{(i)}$ and $D_n^{(i)}$ are calculated as follows:

$$t_1^{(i)} = x_1^{(i)} \tag{3}$$

$$t_n^{(i)} = x_n^{(i)} + S_{n-1}^{(i)} \qquad (n = 2, 3, \cdots) \tag{4}$$

$$D_n^{(i)} = t_n^{(i)} + \Delta S_n^{(i)} \text{ if } A_n^{(i)} \le t_n^{(i)} + \Delta S_n^{(i)} \; (n = 1, 2, \cdots) \tag{5}$$

$$D_n^{(i)} = A_n^{(i)} \qquad \text{if } t_n^{(i)} + \Delta S_n^{(i)} < A_n^{(i)} \; (n = 1, 2, \cdots) \tag{6}$$

Next, we consider how to set $\Delta S_n^{(i)}$. It is clear that the value should be determined depending on the amount of asynchrony $A_n^{(i)}–t_n^{(i)}$. Taking the influence of the slide control on the audio stream into consideration, we introduce two types of modification schemes: *gradual recovery* and *fast recovery* [12]. The gradual recovery scheme modifies the target output time step by step so that the disturbance of an audio stream is limited to a small one. In this case, we set $\Delta S_n^{(i)} = \theta_1$ ($\theta_1 > 0$) and let the minimum allowable interval between two successive control be ω_1. On the other hand, the fast recovery scheme modifies the target output time largely at once, i.e., $\Delta S_n^{(i)} = \theta_2$ ($\theta_2 \gg \theta_1$) in order to recover from the inter-stream asynchrony quickly, though the disturbance of an audio stream becomes large. The minimum allowable interval between two successive control in this case is denoted by ω_2 ($\omega_2 \gg \omega_1$).

Let T_{h1} and T_{h2} be threshold values of $A_n^{(1)}–t_n^{(1)}$ for gradual recovery and fast recovery, respectively ($T_{h1} < T_{h2}$). Also, suppose that the m-th MU is the last MU generated in stream 2 before the generation of the n-th MU in stream 1. Under these conditions, we adopt the following rule to determine the slide time.

(a) If $A_m^{(2)} \le t_m^{(2)}$ and $T_{h2} \le A_n^{(1)} - t_n^{(1)}$,
then $\Delta S_n^{(1)} = \Delta S_m^{(2)} = \theta_2$.
(b) If $A_m^{(2)} \le t_m^{(2)}$ and $T_{h1} \le A_n^{(1)} - t_n^{(1)} < T_{h2}$,
then $\Delta S_n^{(1)} = \Delta S_m^{(2)} = \theta_1$.
(c) If $A_m^{(2)} \le t_m^{(2)}$ and $A_n^{(1)} - t_n^{(1)} < T_{h1}$,
then $\Delta S_n^{(1)} = \Delta S_m^{(2)} = 0$.
(d) If $t_m^{(2)} < A_m^{(2)}$, then $\Delta S_n^{(1)} = \Delta S_m^{(2)} = A_m^{(2)} - t_m^{(2)}$.

It should be noted that rule (d) requires the video MU to wait for the output of the corresponding audio MU when the latter arrives behind the target output time. This implies that the audio is essentially the master stream in the VTR algorithm.

4. Experimental Methodology

This section describes the configuration of the experimental system, experimental conditions and performance measures used in the experiment. We evaluate the system performance with TCP and that with UDP. In particular, we examine how the quality of the PHS channel and the Internet traffic affect the performance.

4.1 Experimental System

We constructed an experimental system according to Fig. 1; it is composed of a media server, the Internet part, a dial-up router, an N-ISDN switch, a base station, a simulated wireless environment and the mobile terminal part. Figure 2 illustrates the detailed configuration. As in [13], we use two mobile terminals: one for video and the other for audio. That is, the video and audio go through the Internet as two distinct transport streams, each of which is mapped on a B channel of the N-ISDN; from the base station, the two streams are transmitted over two separate PHS channels.

The reason why we use two separate 32 kb/s channels for video and audio is to obtain good quality of each medium. The 32 kb/s bearer service, only which was available at the time of the experiment, is not sufficient for good quality transmission of both media. The 64 kb/s bearer service, which is currently available, can realize the same condition as that of two separate 32 kb/s channels. However, efficient sharing of a single 64 kb/s channel by video and audio is another important issue to be addressed. Regarding this, the authors have already performed a simulation study on an ap-

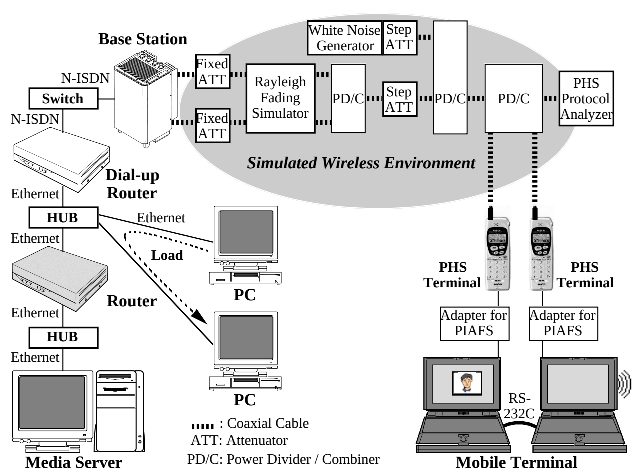


Fig. 2 Configuration of the experimental system.

† The VTR algorithm for live media decreases the target output time when the corresponding MU arrives a specified amount of time earlier than the target output time to preserve the real-time property [8].

plication of the H.223 Annex multiplexing protocol to the audio and video transmission over a 64 kb/s PHS channel [19]; its result provides a guideline to the issue.

We now make a brief comment on each component in Fig. 2 in turn. The media server (MMX Pentium II 266 MHz, Windows NT) stores an H.263 video file and the corresponding G.726 file; it sends them out to the mobile terminals on demand.

The internet part consists of two 10BASE-T Ethernet hubs interconnected by a router (Cisco 2514). One hub connects the media server to the Cisco 2514 router. The other† is shared by the dial-up router, the Cisco 2514 router and two PCs. One of the PCs sends fixed-size data messages of 1472 bytes each, which serves as interference with the audio-video stream, to the other PC under the UDP protocol at exponentially distributed intervals; the amount of the interference traffic is adjusted by changing the average of the interval.

Note that we have slightly changed the configuration of the Internet part from that in [13]: the Cisco 2514 router and the hub connected to the media server have been added so that we can examine the effects of the flow control and end-to-end retransmission mechanisms of TCP. The receive-buffer capacity of the TCP socket was selected to be 12000 bytes, which is slightly smaller than the receive-buffer size of the dial-up router. This way, the TCP flow control can prevent buffer overflow at the dial-up router. For UDP††, on the other hand, the buffer overflow at the dial-up router may occur when the quality of the PHS channel is not good; this is due to the retransmission by PIAFS and the lack of the flow control in UDP. The buffer overflow at the Cisco 2514 router is observed for both TCP and UDP when the *data load* (i.e., the interference traffic) is high. Thus, in our experimental system, the end-to-end retransmission by TCP is triggered only by the buffer overflow at the Cisco 2514 router, which is caused by the data load.

The dial-up router is connected to the base station via the N-ISDN switch.

The simulated wireless environment consists of a Rayleigh fading simulator, power dividers/combiners, a white noise generator, and fixed and variable attenuators, all of which are connected together with coaxial cables as shown in Fig. 2. In order to have a realistic propagation environment, we have inserted a fixed attenuator (36 dB) between each antenna terminal of the base station and the Rayleigh fading simulator. The two outputs of the fading simulator, which are provided so as to be uncorrelated with each other, are combined into a single signal; it is then input to a variable attenuator that can change the attenuation by the dB in the range from 0 to 121. White noise, which can be set to various levels ranging from 0 to 60 in steps of dB, is added to the output signal of the attenuator. The corrupted signal is divided among two PHS terminals and a PHS protocol analyzer, which can measure the received carrier level and noise level at the PHS terminal on each carrier frequency. This enables us to calculate the *CNR* (*Carrier to Noise Ratio*).

The transmitter power of the base station is 500 mW, and that of a PHS terminal is 10 mW. In the measurement, we have adjusted the attenuation so that the received carrier level at the PHS terminals can be high enough for their internal thermal noise to be negligible. Thus we set the received carrier level to about 48 dBμV, which is a medium level in PHS usage.

We also selected a frequency of 1916.150 MHz (the carrier number is 71) as a PHS control channel so that the frequency can be different from those of the PHS service providers.

The terminal part consists of a pair of mobile terminals (MMX Pentium 166 MHz, Windows 95) each equipped with a PIAFS adapter and a PHS terminal. In the experiment, information on media synchronization was exchanged between the two terminals via the RS-232C interface. The necessary time to exchange the information at a time is about 2 ms; therefore, the effect of the time on the measurement is negligible.

4.2 Experimental Conditions

(1) Specifications of video and audio

In the experiment, we used a man's head view and his voice, which have been encoded into H.263 video††† and G.726 audio, respectively; the original recording time is 241.0 seconds. Both were decoded with software at the terminals.

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. The target coding rate of video is 20 kb/s, 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: $TR_1 = 0$ and $T_n^{(1)} = TR_n / 30$; ($n \geq 2$).

The audio coding rate is 24 kb/s, 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 MU is given by $T_m^{(2)} = (m–1) / 10$.

(2) Parameters of slide control

The parameter values for slide control were selected for the experiment as follows. First, we chose the same values for the gradual and fast recovery as those in [13]: $T_{h1} = 100$ ms, $\theta_1 = 100$ ms, $\omega_1 = 1000$ ms, $T_{h2} = 2000$ ms, $\theta_2 = 4000$ ms, and $\omega_2 = 60000$ ms.

We establish the origin of time for media synchronization at the arrival time of the first video MU $A_1^{(1)}$. Consequently, we have $t_1^{(1)} = A_1^{(1)} + \tau_1^{(1)}$ and $t_1^{(2)} = t_1^{(1)} + T_1^{(2)} - T_1^{(1)}$. We then set $\tau_1^{(1)} = 500$ ms.

4.3 Performance Measures

In order to assess the quality of intra-stream synchronization, i.e., the smoothness of output in stream i, we use the *coefficient of variation of output interval* $C_V^{(i)}$, which is defined as the ratio of the standard deviation of the output interval to its average; that is,

† This is not a switching HUB.

†† The socket buffer capacity of UDP was 65280 bytes in our experiment.

††† None of the four coding options is used.

$$C_v^{(i)} = \left[\left\{ \Sigma_{n=1}^{N-1} \left(D_{n+1}^{(i)} - D_n^{(i)} - E^{(i)}\right)^2 \right\} / (N-1) \right]^{1/2} / E^{(i)}$$

$$E^{(i)} = \left(D_N^{(i)} - D_1^{(i)}\right) / (N-1),$$

where lost MUs are excluded in the calculation.

For inter-stream synchronization, we define the error between the n-th MU in stream 1 and the corresponding MU in stream 2 (say the m-th MU) as $\Delta_n^{(1\text{–}2)} = (D_n^{(1)} - D_m^{(2)}) - (T_n^{(1)} - T_m^{(2)})$, $(n \geq 1,\ m \geq 1)$. Then, as the performance measure of inter-stream synchronization quality, we use the average square of $\Delta_n^{(1\text{–}2)}$ with respect to n.

We also evaluate the *average MU rate*, which is defined as the average number of (either audio or video) MUs output in a second at the terminal. 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 terminals. 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 Results

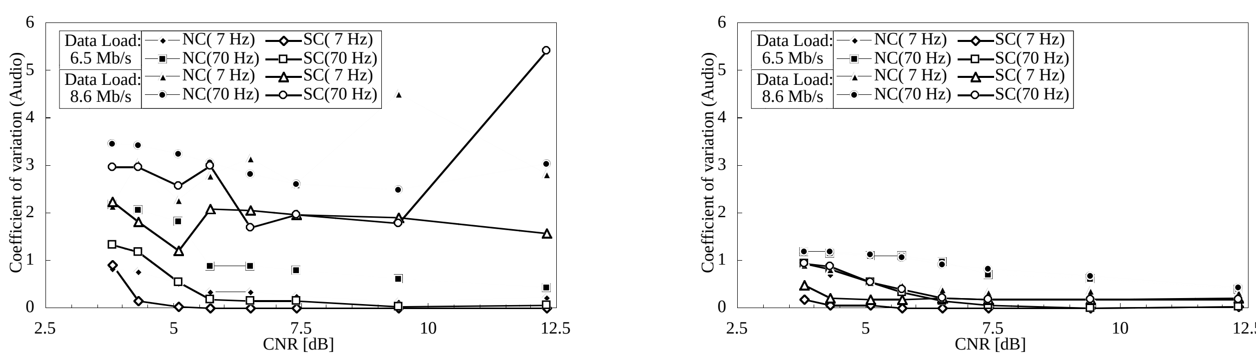
We made measurement of the performance measures defined in the previous section for both TCP and UDP, which will be presented in this section. We plot the experimental result of each measure as a function of the CNR for TCP and for UDP separately. Each figure presented below displays the result of the system with the slide control and the corresponding result of the system with no control, which are denoted by "SC" and "NC," respectively.

In both kinds of systems, we examine the effects of Rayleigh fading on the PHS channels and the data load (i.e., interference traffic) in the Internet on the media synchronization quality. We set the fading frequency to either 7 Hz or 70 Hz, which is indicated in the parentheses following "NC" or "SC" in each figure to be shown. The frequency of 7 Hz corresponds to a walking speed of 4 km/h and 70 Hz to a vehicle speed of 40 km/h. We also selected two kinds of the data load: 6.5 Mb/s and 8.6 Mb/s, which imply a medium load and a heavy load, respectively, in our experimental environment.

We now discuss the measurement results in turn.

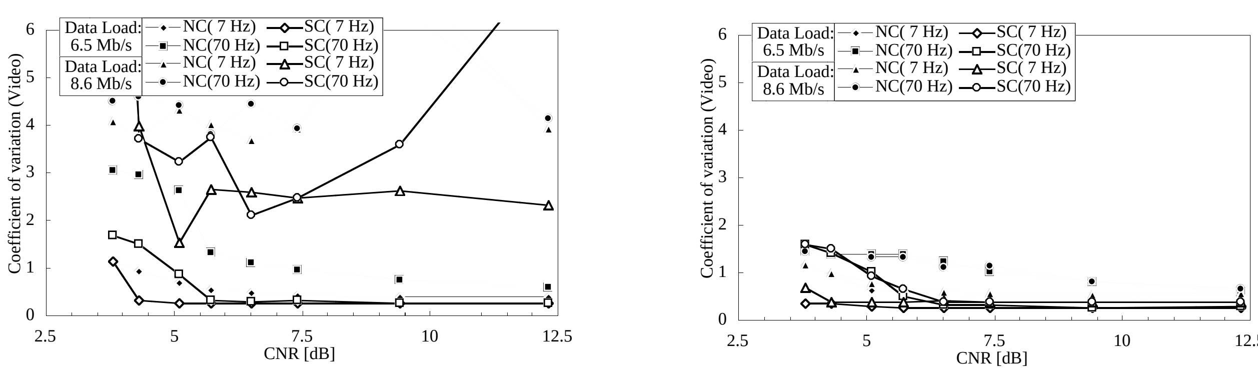
(1) Coefficient of variation of output interval

Figure 3 shows the coefficient of variation of output interval for audio, and Fig. 4 that for video. First, let us examine the audio case. Comparing Figs. 3(a) and (b), we immediately see that TCP produces larger values of the coefficient than UDP for the data load of 8.6 Mb/s; in this case, the coefficient value in TCP is very large for both SC and NC regardless of the CNR value. This implies that the Internet part becomes the performance bottleneck, which comes from the end-to-end retransmission by TCP to recover packets lost at the Cisco 2514 router. For the data load of 6.5 Mb/s, on the other hand, the difference between TCP and UDP is gener-



(a) TCP (b) UDP

Fig. 3 Coefficient of variation of output interval for audio versus CNR.



(a) TCP (b) UDP

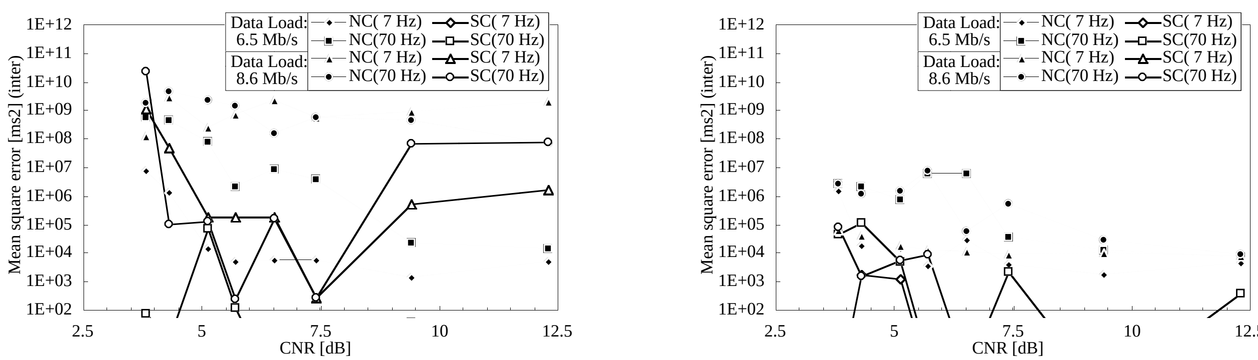
Fig. 4 Coefficient of variation of output interval for video versus CNR.

ally small; however, we notice somewhat large difference for small values of the CNR, say 3.8 dB, when the fading frequency is 70 Hz. This is due to the flow control of TCP. We also observe that the coefficient becomes smaller as the CNR increases at this data load since retransmission on the PHS channel by PIAFS decreases. In addition, we find that SC improves the quality over NC in both TCP and UDP.

Regarding the video, we make similar observations to those for the audio.

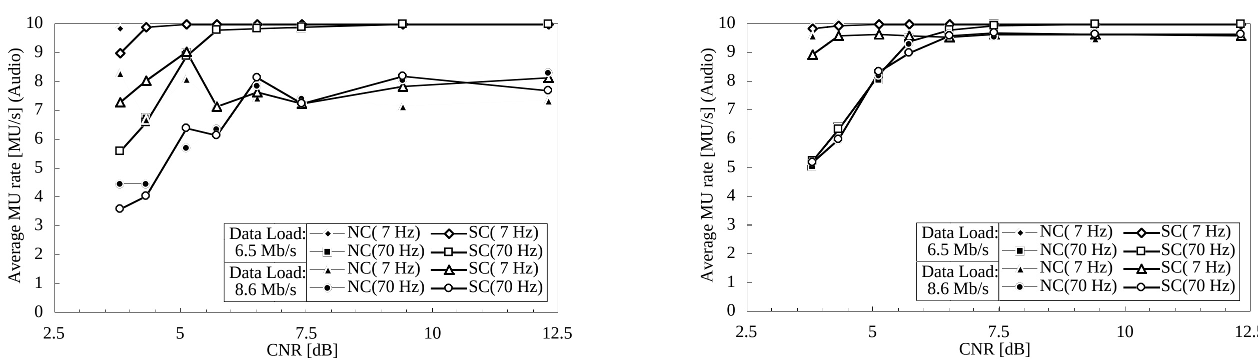
Thus we see that TCP achieves the quality of intra-stream synchronization comparable to UDP only when neither the data load nor the fading frequency is high and at the same time the CNR is high; otherwise, UDP outperforms TCP. It should be noted that even in the case of UDP, PIAFS is used to recover from the loss and error at the data link layer.

(2) Mean square error of inter-stream synchronization
Figure 5 plots the mean square error of inter-stream synchronization versus CNR. Again, we find poor performance of TCP, compared to UDP. To examine the quality in more detail, we utilize Steinmetz's report on human perception of jitters [20]: A skew of less than 80 ms between audio and video (i.e., a mean square error less than 6400 ms^2) attains good quality of inter-stream synchronization, while a time difference beyond ±160 ms (a mean square error larger than



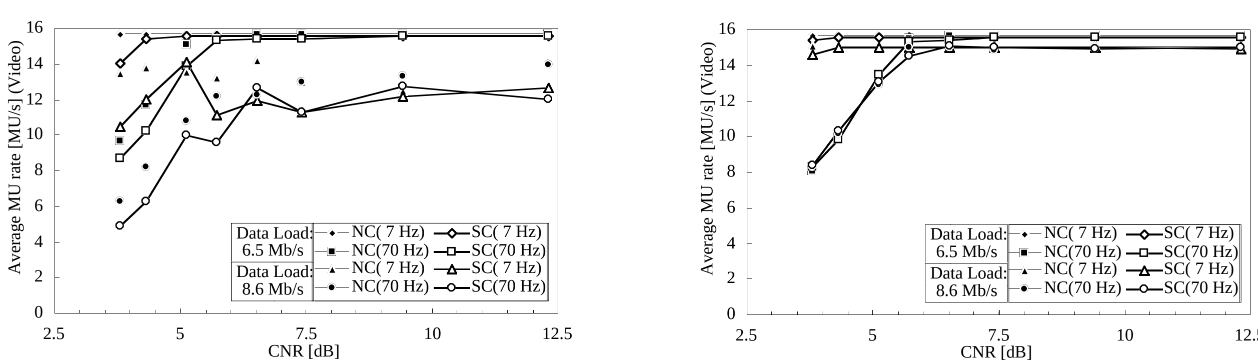
(a) TCP (b) UDP

Fig. 5 Mean square error of inter-stream versus CNR.



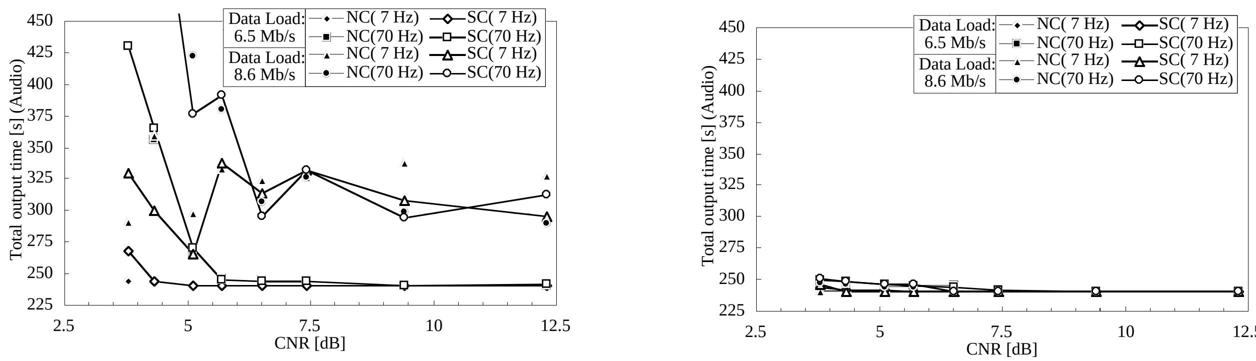
(a) TCP (b) UDP

Fig. 6 Average MU rate for audio versus CNR.



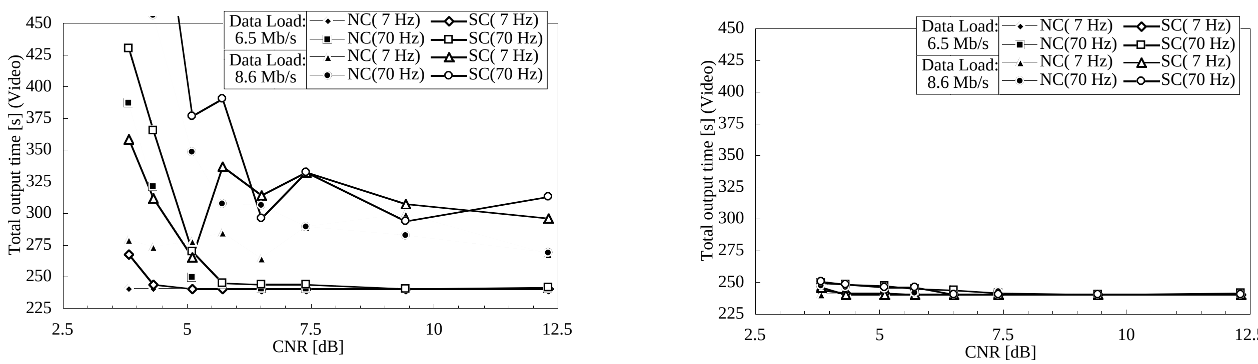
(a) TCP (b) UDP

Fig. 7 Average MU rate for video versus CNR.



(a) TCP (b) UDP

Fig. 8 Total output time of audio versus CNR.



(a) TCP (b) UDP

Fig. 9 Total output time of video versus CNR.

25600 ms^2) corresponds to asynchrony. The criteria tell us that it is very hard for TCP to attain good quality of inter-stream synchronization at the data load of 8.6 Mb/s even if the side control is carried out. When the data load is 6.5 Mb/s, the slide control is effective in improving the quality of TCP. For the 7 Hz fading frequency, the mean square error of TCP with NC is comparable to that of UDP unless the CNR is low.

We next consider the case of UDP. Then, we first notice in "NC" that the fading frequency is more dominant than the data load; this implies that the MU loss at the dial-up router occurs more frequently than that at the Cisco 2514 router. The slide control can attain good quality of inter-stream synchronization unless the CNR is low, say for CNR > 5 dB.

Thus, we also see that with respect to the quality of inter-stream synchronization, UDP outperforms TCP in almost all cases.

(3) Average MU rate and total output time

Figures 6 and 7 display the average MU rate for audio and video, respectively. Also, Fig. 8 plots the total output time of audio versus CNR, and Fig. 9 that of video. We first find a common feature to TCP and UDP in Figs. 6 and 7; that is, the average MU rate decreases as the CNR decreases. However, the reason for the decrease is different form each other. In the case of TCP, it is due to the increase of the total output time as we can see from Figs. 8 and 9; the increase is caused by the end-to-end retransmission and flow control of TCP as well as the slide control. On the other hand, UDP decreases the MU rate mainly because of the loss at the dial-up router.

Comparing TCP and UDP when the data load is 8.6 Mb/s for CNR > 6 dB, the difference in the MU rate between the two is large; this means bad quality of TCP.

6. Conclusions

We compared the media synchronization quality of TCP and UDP in PHS Internet access, using a simple experimental system. The streaming transmission of H.263 video and G.726 audio was considered. By experiment, the effects of the PHS wireless channel (especially, fading frequency and CNR) and the Internet traffic (i.e., the data load, which serves as the interference with the video and audio streams) were examined. We measured the coefficient of variation of output interval, the mean square error of inter-stream synchronization, the average MU rate and the total output time. As a result, we found that UDP outperforms TCP in almost all cases; TCP performs comparably to UDP only when neither the data load nor the fading frequency is high and at the same time the CNR is high. This is because PIAFS is effective in restoring the loss and error on the PHS wireless channel, whereas the end-to-end retransmission of TCP is too persistent to attain small delay jitters.

Thus we conclude that UDP along with PIAFS is enough to have good performance, and TCP with PIAFS has too strong retransmission capability since it produces tremendous delay jitters to recover MU loss completely. In order to obtain better performance than that of UDP with PIAFS, we need to employ a transport protocol which can retransmit MUs within a specified timeout interval as stated in [13]; the protocol can recover MU loss just with a small amount of jitter increase, though the recovery is not necessarily complete. This should be included in future works.

Acknowledgment

The work was supported by *Telecommunications Advancement Organization of Japan* and by the *Grant-In-Aid for Scientific Research of Japan's Ministry of Education, Science and Culture*.

References

[1] Association of Radio Industries and Business (ARIB), "Personal Handy Phone System ARIB Standard Version 3 RCR STD-28," Nov. 1997.
[2] J.Alvear, Web Developer.com Guide to Streaming Multimedia, John Wiley & Sons, Inc, 1998.
[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] I.Kouvelas, V.Hardman, and A.Watson, "Lip synchronization for use over the Internet: Analysis and implementation," Conf. Rec. GLOBECOM'96, pp.893-898, Nov. 1996.
[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, H.Nakanishi, and Y.Ishibashi, "Dynamic resolution control and media synchronization of MPEG in wireless LANs," Conf. Rec. GLOBECOM'97, pp.138-144, Nov. 1997.
[7] Y.Ishibashi, S.Tasaka, and E.Minami, "Performance measurement of a stored media synchronization mechanism: Quick recovery scheme," Conf. Rec. GLOBECOM'95, pp.811-817, Nov. 1995.
[8] Y.Ishibashi, S.Tasaka, and A.Tsuji, "Performance measurement of a live media synchronization mechanism in an ATM network," Conf. Rec. ICC'96, pp.1348-1354, June 1996.
[9] S.Tasaka, Y.Ishibashi, and H.Imura, "Stored media synchronization in wireless LANs," Conf. Rec. GLOBECOM'96, pp.1904-1910, Nov. 1996.
[10] S.Tasaka and Y.Ishibashi, "Stored media synchronization schemes in ATM and wireless networks: A performance comparison," Proc. IEEE ICUPC'97, pp.766-772, Oct. 1997.
[11] S.Tasaka and Y.Ishibashi, "Single-stream versus multi-stream for live media synchronization," Conf. Rec. ICC'98, pp.470-476, June 1998.
[12] M.Kato, N.Usui, and S.Tasaka, "Stored media synchronization based on buffer occupancy in PHS," Proc. PIMRC'97, pp.1049-1053, Sept. 1997.
[13] S.Tasaka, M.Kato, and K.Nakamura, "Stored media synchronization in PHS Internet access," Conf. Rec. GLOBECOM'98, pp.113-119, Nov. 1998.
[14] M.Kato, K.Nakamura, and S.Tasaka, "Experiment on stored media synchronization over PHS channels," IEICE Trans., vol.J81-B-I, no.11, pp.709-719, Nov. 1998.
[15] ITU-T Recommendation H.263, "Video coding for low bit rate communication," March 1996.
[16] ITU-T Recommendation G.726, "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)," Dec. 1990.
[17] PHS Internet Access Forum, "PHS Internet Access Forum Standard (PIAFS)," March 1997.
[18] H.Santoso, L.Dairaine, S.Fdida, and E.Horlait, "Preserving temporal signature: A way to convey time constrained flows," Conf. Rec. GLOBECOM'93, pp.872-876, Dec. 1993.
[19] M.Kato, Y.Kawai, and S.Tasaka, "Performance evaluation of media synchronization in PHS with the H.223 Annex multiplexing protocol," IEICE Trans. Commun., vol.E81-B, no.12, pp.2423-2431, Dec. 1998.
[20] R.Steinmetz, "Human perception of jitter and media synchronization," IEEE J. Sel. Areas Commun., vol.14, no.1, pp.61-72, Jan. 1996.

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 Electrical and Computer Engineering. In 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, high-speed networks 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.

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 and integrated wired and wireless network systems. He is presently a chief researcher in Hypermedia Research Center of Sanyo.

Kotaro Nakamura received the B.S. and the M.S. degrees in electrical and computer engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1997 and 1999, respectively. At Nagoya Institute of Technology, he studied mobile multimedia communication systems in PHS. In April 1999, he joined OMRON Corporation.