PAPER Special Section on Applications and Implementations of Digital Signal Processing

Extension of Image Transport Protocol Allowing Sever-Side Control of Request for Retransmission

Kiyoshi NISHIKAWA[†a)], ***Member***, **Takako SASAKI**[†], ***Nonmember***, ***and*** **Hitoshi KIYA**[†b)], ***Member***

SUMMARY In this paper, we propose an extension to the image transport protocol (ITP). When images are transmitted through the Internet, TCP is generally used because it ensures the reliable transmission. However, interactivity will largely affected because of its acknowledgement scheme. This becomes remarkable in the network where packet-loss rate is relatively higher like wireless LANs. For more efficient image transmission, ITP was proposed. Like UDP, in ITP transmission, packets can be transmitted without acknowledgement of the reception. This contributes to improve the interactivity, on the other hand, some of packets may lost during transmission. ITP has a mechanism that the receiver-side can control the retransmission of the lost packets to maintain the quality of the received image. However, it is a hard task for the receiver to select which packets to be retransmitted. In this paper, we propose an extension to ITP by which the server can mark the importance of each packet. This helps the receivers to select important packets for requesting retransmission for server.

key words: *ITP, image transmission, ADU, transmission delay*

1. Introduction

In this paper, we propose an extension to the image transport protocol (ITP) [1] to enable server-side control of request retransmissions.

Transport of still images constitute a significant part of the Internet traffic. When transmitting images, transmission control protocol (TCP) is normally used as the transport layer protocol over internet protocol (IP) transparently from a higher protocol layer like hyper text transfer protocol (HTTP). TCP was originally designed [2] for data transmissions like FTP (file transfer protocol), SMTP (simple mail transfer protocol), etc. These protocols require reliability, that is, all bits should be transferred without errors, and in order. However, interactivity is not required for them [3]. Note that, in this paper, we use the term "interactivity" to indicate "the user-perceived latency in displaying the requested image" as the original ITP paper [1]. On the other hand, transmission of images (and movies, or audio-streams) requires interactivity, but it tolerates to some packet-loss. For transmitting movies or audio streaming, the real-time transfer protocol (RTP) was proposed and is used in such systems as voice over IP (VoIP). However, this protocol is not suitable for the transmitting of still images.

ITP was proposed for sending still images efficiently through the Internet. It is implemented on user datagram protocol (UDP), which is an alternative transport layer protocol above IP. UDP provides almost the same functionalities as those of raw-IP protocol. Therefore, we can construct a customized transport protocol above the UDP with only slight overhead.

In ITP transmission, images are packetized at application data unit (ADU) boundaries. An ADU is a piece of image that can be processed independently [4]. The ADU-based transmission of the ITP provides some advantages over the TCP [1]: (i) out-of order delivery of ITP packets, (ii) receiver-side quality control by selectively requesting retransmission when packets are lost. Advantage (ii) is important because ITP uses UDP so that all packets are not guaranteed to be received. In ITP, receivers select which packets to be retransmitted and send requests for retransmission from the server when packet-loss occurs.

In view of interactivity, the number of requests for retransmission should be small. At the same time, perceptual quality of images becomes better as the number of received packet increases. Therefore, an important task is to select which packets should be retransmitted for the tradeoff between image quality with interactivity. However, in ITP, the selection is rather a hard task for receivers because they receive no information about the perceptual importance of each ADU, or each packet. Therefore, receivers must estimate the importance of lost packets, but this seems impossible in most cases.

In the view point of implementing ITP receivers, the description in the original paper [1] is not satisfactory. In the paper, a customized ITP for JPEG formated images, JPEG-ITP is described: They suggest to use "priority list" to control scheduling of the packet sending when packet losses occur. However, there is no detailed description on how to make the list nor how to use those information to determine for sending request for retransmission to the server. In this paper, we consider the behavior of the receivers in packet-loss environments where JPEG formated image are transmitted, and this is missing in [1]. It should be noted that we can apply the idea of the proposed method to other image formats that have similar structure to JPEG such as JPEG2000 [5].

Our purpose in this paper is to propose a simple method by which the server can mark the importance of each packet and send that information to the receivers. For that purpose, we extend the protocol-header of ITP to include the perceptual importance of the ADUs. According to that information, receivers can easily select important packets without

Manuscript received June 30, 2003.
Manuscript revised October 6, 2003.
Final manuscript received November 21, 2003.

[†]The authors are with the Dept. of Electrical Engineering, Tokyo Metropolitan University, Hachioji-shi, 192-0397 Japan.
a) E-mail: kiyoshi@eei.metro-u.ac.jp
b) E-mail: kiya@eei.metro-u.ac.jp

any computational cost. Using the proposed method, receivers can maintain a satisfactory level of image quality with the smallest number of requests for retransmission. We demonstrate the effectiveness of the proposed method by applying it to JPEG images.

This paper is organized as follows. In Sect. 2, a review of image transmission using TCP and ITP is given as background. The proposed method is then described in Sect. 3. Simulation results are provided in Sect. 4 and Sect. 5 to demonstrate the effectiveness of the proposed method.

2. Background

Here, image transmission using TCP and ITP are briefly described.

2.1 Problems of Image Transmission when Using TCP/IP

As previously stated, when transmitting still images via the Internet, TCP is normally used as the transport layer protocol through the higher level protocols like HTTP. TCP achieves reliable transmissions by mechanisms such as the automatic retransmission of lost packets [2].

When a packet loss occurs in TCP transmission, to ensure in-order packet delivery, other already-received packets that should be located after the lost packet are buffered and wait for the lost packet to be retransmitted. This may require a rather long time, and hence, interactivity of the image transmission will be degraded. As shown in [1], the interactivity will be affected even if only one packet is lost. This becomes remarkable in TCP-based image transmission over the networks where packet-loss rate is relatively higher like wireless local area networks (WLAN).

The TCP protocol was originally designed for data transmission, such as FTP (file transfer protocol), or SMTP (simple mail transfer protocol) where all the information bits should be delivered without an error and in-order, but interactivity is less important. However, when transporting images (also movies, or audio-streams), interactivity is required, but some packet-loss is tolerated so long as a satisfactory level of perceptual quality is maintained [3].

2.2 ITP: Image Transport Protocol [1]

To overcome the above mentioned inconvenience of transmitting images via TCP, ITP was proposed [1]. ITP is a customized transport layer protocol implemented on UDP for the sake of effective image transmission. UDP is an alternate transport layer protocol and it enables unbuffered transmission of data, and hence, it can improve the interactivity of the image transmission via the Internet. However, we should provide a transport control mechanism like the one provided by TCP.

ITP provides an original transport control suited for still-image transmission. As a result, it provides several advantages over the image transport using TCP. Some of them are

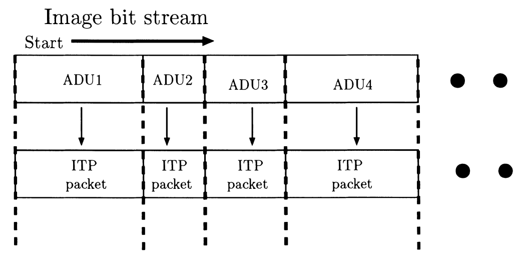


Fig. 1 Packetizing image stream into ITP packets.

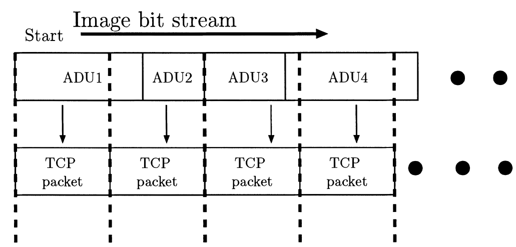


Fig. 2 Packetizing image stream into TCP packets.

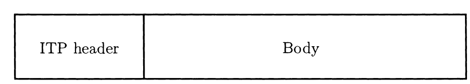


Fig. 3 Structure of ITP packet.

1. **ADU-based packet construction** [4]
   An ITP packet consists of the data that is fragmented at application data unit (ADU) boundaries as shown in Fig. 1. An ADU is a piece of an image that can be processed independently. In TCP transport, images are packetized at non ADU boundaries as shown in Fig. 2 so that each packet cannot be processed independently. Each ITP packet contains an '*ITP header*,' that contains protocol-related information, and a '*body*,' that consists of an ADU, or a part of an ADU when the ADU is too bigger to be contained in a single packet (Fig. 3).
   In Fig. 4, the structure of the ITP header information is given. For a detailed explanation of each component, readers are asked to refer to the ITP paper [1].
2. **Out-of-order delivery**
   ITP allows out-of-order delivery of packets. Thanks to the ADU-based packet structure, the order of packets loses significance.
3. **Congestion control**
   Because the ITP is constructed using the UDP, congestion control is an important task. The ITP uses a dedicated process called congestion manager [6] for this control.
4. **Receiver-side control of perceptual quality of images**
   When packet losses occur, ITP provides a mechanism that enables the receiver side to control the perceptual quality of the decoded images. Namely, receivers can select whether to request servers to retransmit the lost packets. If the receiver evaluates the perceptual impor-

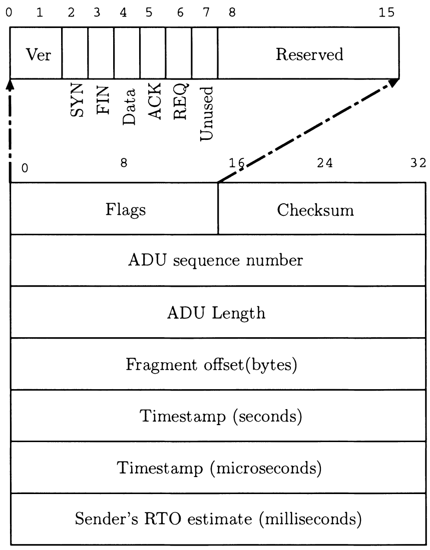


Fig. 4 Details of ITP header. (RTO is for retransmission timeout and it works as in TCP [7])

tance of the lost packets to be low then those packets are ignored and an alternate process such as interpolation will be used. This mechanism reduces unnecessary data transmission without reducing the perceptual quality of images.

We concentrate on the last of these characteristics in this paper, specifically the control of perceptual quality of the received image in the event of packet-loss. As previously stated, to control this quality, the ITP enables the receiver-side to choose to request the server to retransmit the lost packets. However, this selection requires knowledge of the perceptual importance of ADU contained in the lost packets. This information is unavailable in most cases.

Moreover, we know that a compressed image is structured so that it can be divided into several parts according to the importance of the information. That is, the header information is the most important part of the image because it contains informations such as image format, image size, number of bit per pixel, etc. This information should be delivered without errors to ensure that the image can be decoded. Then, DC and lower frequency components of the image have significant importance for maintaining the perceptual quality of the decoded image. On the other hand, higher frequency components have less perceptual importance so that receivers can decode the image with only slight degradations of perceptual quality even if those components are not obtained.

3. Proposed Method

Here, we describe the proposed method. The proposed method is a simple extension of the ITP to enable the server (sender) to mark the importance of ADUs as an aid to receivers. Using these marks, receivers can determine the importance of each packet and thus maintain the perceptual quality of the decoded image. With this information, receivers can easily determine whether to send a request for retransmission of a lost packet. This feature reduces unnecessary retransmission without sacrificing the perceptual quality of the received images.

3.1 Extension of ITP Header

As described in the preceding section, when we transmit images using the ITP, the receiver-side must determine whether to request retransmission of lost packets. Knowing the perceptual importance of the ADU contained in those lost packets is difficult, and this makes the determination to request a retransmission a hard task for the receiver. Specifically, to estimate the importance of a lost packet, or an ADU, receivers need some process such as decoding, or checking. This requires calculation and therefore wastes CPU power, and may affect interactivity.

To overcome this shortcoming of the original ITP, we propose extending the ITP. Specifically, we propose extending the header information of each ITP packet to include the perceptual importance of the ADU.

Let us consider how to include that information in the ITP header. The easiest way is to include the importance for each packet in its protocol header. However, this idea cannot work because the information will be lost when the packet itself is lost. Therefore, we cannot obtain the extended information when we need it. We describe the proposed method in the next subsection.

3.2 Image Format and Perceptual Importance

Instead of the straight-forward method described above, we propose to use the structure of compressed images in standard formats such as JPEG [8], [9], or JPEG2000 [5].

By considering the structure of those compressed images, we can find the following characteristics:

1. The most important information, sometimes called global header information (GHI), such as the size of the image, image format, etc, are contained at the beginning of the image data.
2. When an image is compressed using a format supporting progressive representation, e.g., progressive JPEG or JPEG2000, the image data that contains the perceptually important parts, namely the DC and the lower frequency components, are located at, after, or near the GHI.
3. On the other hand, perceptually unimportant parts are located at the end of the bit stream in such image formats.

Therefore, we can classify the perceptual importance of an ADU from its position in the bit stream of the compressed image file. In other words, we can classify the perceptual importance of an image by the sequential number of packets counted from the first packet.

We propose to extend the ITP by adding information

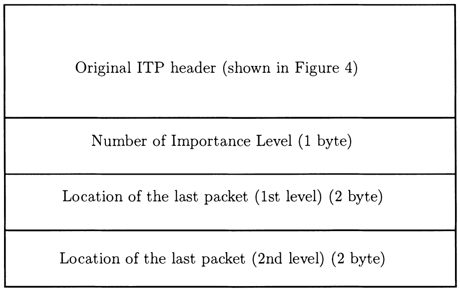


Fig. 5 Structure of the header information of the proposed method.

expressing the importance of a packet's contents. We would add the following two components to the original ITP header information, as is shown in Fig. 4:

1. **Importance-level number**
   As shown in the following subsection, we propose to introduce the concept of '*importance level*' to provide a criterion for the receiver-side. This header component would show the importance level set by the server. We assume that the importance-level number will depend on the image transmitted itself and the image compression method used.
2. **Location of the last packet in each importance level**
   This information indicates the sequential number of the last packet in each importance level counted from the first packet. We assign two bytes for each importance level. The length of this part varies depending on the number of importance-level.
   Note that we can omit the information of the last importance level because this is the last packet of an image. We also note that we assumed that the total number of packets of a image is less than 65536.

The structure of the header information of the proposed method is shown in Fig. 5. Note that this figure shows an example when the importance-level number is three. As shown in this figure, we add additional information at the bottom of the original ITP header which is 28-byte long (Fig. 4).

3.3 Example: Progressive JPEG Image

Here, we describe the proposed method when we apply it to the JPEG images as an example. There are several variations in the JPEG image compression format, and one of them is the progressive JPEG image format.

As the name suggests, progressive JPEG is an image format that supports progressive-decoding. There are three ways to implement progressive JPEG, namely, spectral selection (SS), successive approximation (SA), and hierarchical mode (HM). Of these three, we consider only the SS method here. The structure of an image is shown as in Fig. 6.

The image-header information is at the beginning of the image file. Then, components of the image are ordered from low to high frequencies: DC, low frequency AC, and high frequency AC components. Notice that these three are arranged according to their perceptual importance.

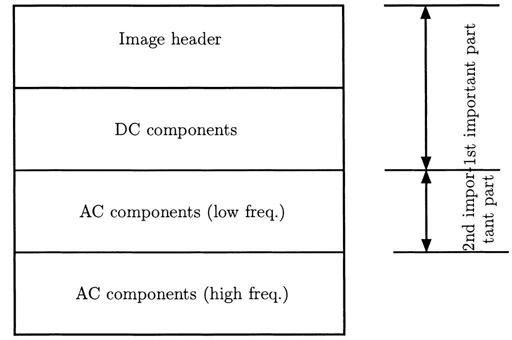


Fig. 6 Structure of an image in progressive JPEG format.

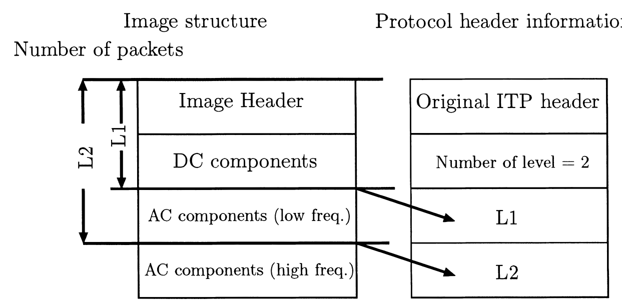


Fig. 7 Relation between the image structure and the contents of the protocol header.

In this case, the proposed method marks the packets that contain '*Image header*' and '*DC component*' as ones of first-level importance. Then, those containing '*AC component*' at low frequency are marked as being of the second-level importance. Finally, the proposed method marks the high frequency AC components as those of third-level importance. Note that we do not need to send the location of the last packet in the last importance level because it is the last packet to be transmitted.

Therefore, for this case, we set the importance-level number as 2 (counted from 0). Figure 7 describes this situation. Locations of each level can be found by searching the start-of-scan (SOS) markers.

However, the baseline JPEG does not have a structure like that of the progressive JPEG. Even so, we must receive the '*header*' information in order for the receiver to decode it. Therefore, we set the importance-level number as one, and we include the header information in this importance level.

The proposed method also can be applied to JPEG2000 images. JPEG2000 is a recent standard for image compression [5]. A JPEG200 image has a structure similar to that of the progressive JPEG image described above. However, it is a more complicated standard for achieving higher compression rates. In this case, we may classify the image in more importance levels according to the structure of the image in JPEG200 format.

4. Simulation

Here, we demonstrate the effectiveness of the proposed method by describing computer simulations of it.

4.1 Configuration

We simulated a small IP connected-network using the network simulator NS-2 [10]. The configuration used in the simulation is shown in Fig. 8. As shown in the figure, we made three LANs connected each other. In this figure, d_0, d_1, and d_2 show the transmission delay of the indicated path, and BW_0, BW_1 and BW_2 show the bandwidth of each path.

We assumed that the both of LAN#0 and LAN#1 has 100 MB connections with 1 ms inner transmission delay. Here, the *inner transmission delay* means the transmission delay between hosts and the router in the LAN. On the other hand, LAN#2 has 10 MB connections with variable inner transmission delay δ. In the simulation, we compared the throughput characteristics by setting δ as $\delta = 1$ [ms] and $\delta = 10$ [ms] to compare the effect of transmission delay on the throughput characteristics. We used the delay of $\delta = 10$ to simulate a LAN having relatively long delay such as a wireless LAN. We also assumed that the transmission server is in LAN#0 and the receiver in LAN#2. Note that we added LAN#1 for simulating the background traffic and it was configured that the amount of traffic between LAN#0 and LAN#1 are almost equal to those between LAN#0 and LAN#2.

In the following, we define the throughput as the number of received packets per unit time and we show the results using the packet receiving characteristics as in Fig. 1 of [1].

4.2 Throughput Characteristics versus Amount of Transmission Delay

Using this configuration, we compared TCP and the ITP transmissions with the proposed retransmission mechanism in terms of number of packets received per unit time. Note that we adjusted the data rate of the proposed method to become almost same rate of TCP transmission, i.e., 500 packets/s. First, we show the throughput characteristics of both protocols under the transmission delay of LAN#2 is 1 ms.

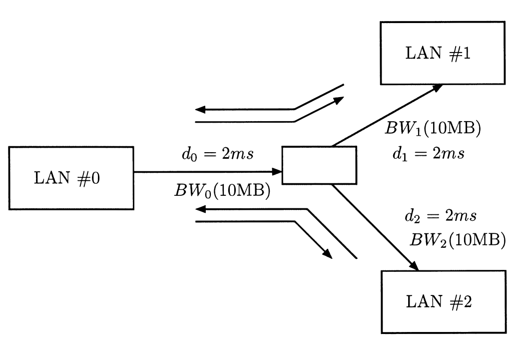


Fig. 8 Configuration of the network used in simulation.

In Fig. 9, the throughput characteristics using the TCP is shown, and in Fig. 10, that of using the propose method is shown. The vertical axis shows the sequential number of each packet and the horizontal axis shows the time.

From those figures, both protocols provide almost same throughput characteristics under the condition that transmission delay was set as $\delta = 1$ [ms]. Only one packet was lost in TCP, and the packet automatically retransmitted after a short time. On the other hand, in the proposed method, six packets are lost and, according to the proposed scheme, those packets are retransmitted after some delay.

In Figs. 11 and 12, the throughput characteristics of TCP and the proposed method are shown. In this case, transmission delay was set as $\delta = 10$ [ms]. By comparing Figs. 9 and 11, we notice that the throughput of TCP is greatly degraded. The reason of this degradation is that delayed transmission of the acknowledgment packet from the receiver to the server. Although the received time of those acknowledgment packets are not shown in the figure, this reasoning is led by checking the results of simulation. On the other hand, the transmission using the proposed method is not affected by the value of the delay δ. Because there are no acknowledgment mechanism in the UDP transmission.

Hence, we can conclude that the proposed method provides better throughput characteristics than the TCP under

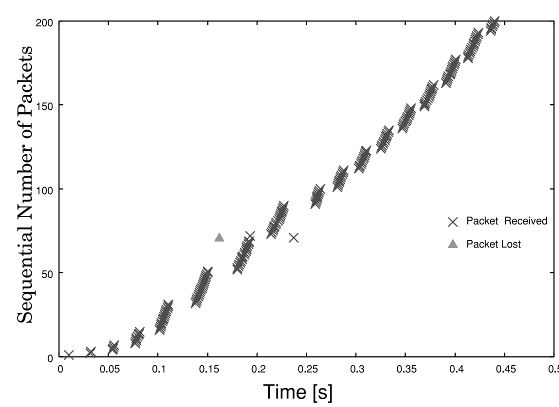


Fig. 9 Results of transmission using TCP. ($\delta = 1$ ms)

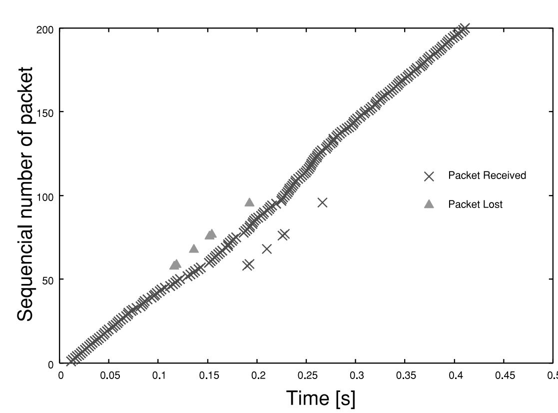


Fig. 10 Results of transmission using extended ITP. ($\delta = 1$ ms)

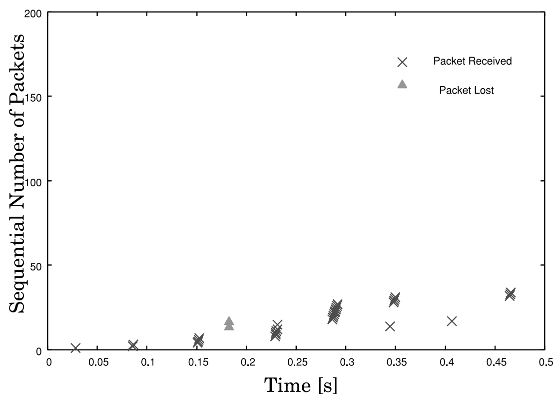


Fig. 11 Results of transmission using TCP. (δ = 10 ms)

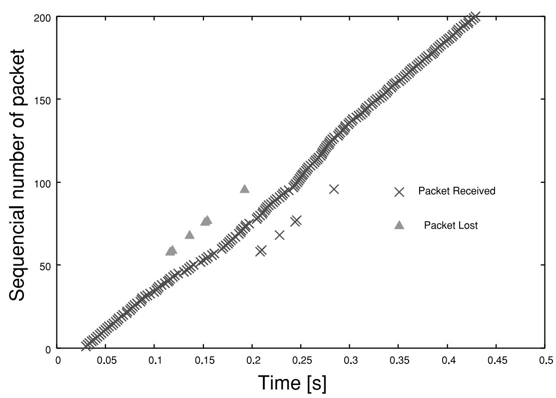


Fig. 12 Results of transmission using extended ITP. (δ = 10 ms)

long delay environments such as wireless LANs. Besides, when a packet loss occurs in TCP transmission, the subsequent packets will be buffered in network module of the operating system and they cannot be processed by the application.

4.3 Throughput Characteristics versus Error Rate

Here, we show the throughput characteristics and packet error rates are in trade-off relations. It is shown that we can control the packet error rate by adjusting the throughput characteristics. In this simulation, we set the inner delay of LAN#2 as $\delta = 10$ [ms] and other configurations were same as those in the previous subsection.

We compared two data rates from the server, namely, 1000 packets/s and 250 packets/s for the proposed method. In Fig. 13, we show the results of the simulation. From the figure, we can say that we can trade the packet error rate of the path by adjusting the data rate of the transmission. This characteristics is due to the fact that the ITP, and hence, the proposed method, is constructed based on the UDP. In other words, we need some mechanism to control the data rate to maintain the acceptable packet error rate, like the congestion control used in ITP [1].

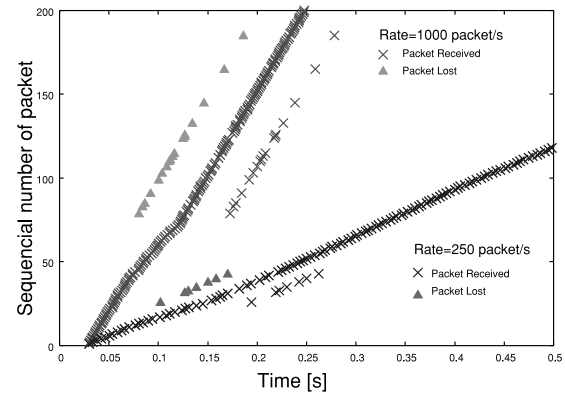


Fig. 13 Throughput characteristics vs. data rate.

5. Comparison with ITP

Here, we compare the proposed method with the ITP in terms of the PSNR.

5.1 Conditions

We used 'Lenna' as our original image, its size was 512 × 512 and was in 8-bpp gray-scale. We transmitted a baseline JPEG image and a progressive JPEG image over a simulated wireless channel with a signal to noise ratio (SNR) of 8 dB. The ADU boundaries were made by inserting restart markers (RSTm) in the image as suggested in [1].

The baseline JPEG image was divided into 65 packets and 7 packets were lost during transmission. In comparison, the progressive JPEG one was divided into 384 packets and 41 packets were lost.

We evaluated the image quality in terms of peak signal-to-noise ratio (PSNR) defined as

$$\text{PSNR [dB]} = 10 \times \log_{10} \frac{255^2}{E\|x(i,j) - \hat{x}(i,j)\|^2} \qquad (1)$$

where $x(i, j)$ denotes the (i, j)-th pixel of the transmitted image, and $\hat{x}(i, j)$ denotes the received image.

To maintain interactivity we limited our requests for retransmission to only 2 ms after the receiver received the first transmitted packet.

5.2 Results

The simulation results for the baseline JPEG image are shown in Figs. 14, 15 and 16. On the other hand, those for the progressive are shown in Figs. 17, 18, and 19. In those figures, Figs. 14 and 17 show the received image without retransmission; Figs. 15 and 18 show the results using ITP; and Figs. 16 and 19 those of using the proposed method. Note that, for ITP, we implemented only the error concealment method suggested in [1] and not implement the request for retransmission because [1] does not mention the detail on the selection, and hence, we cannot implementing the

Fig. 14 Results of transmission without any post processing. (PSNR = 22.21 dB)

Fig. 15 Results of ITP transmission with error concealment. (PSNR = 25.41 dB)

Fig. 16 Results of proposed method. (PSNR = 28.22 dB)

Fig. 17 Results of simulation using progressive JPEG image. (PSNR = 23.12 dB)

Fig. 18 Results of ITP transmission with error concealment using progressive JPEG image. (PSNR = 24.44 dB)

Fig. 19 Results of simulation using progressive JPEG image. (PSNR = 26.8 dB)

mechanism of retransmission of the lost packets.

We point out that, as mentioned in Sect. 3.3, the baseline JPEG image does not have multiple importance levels so that requests for retransmission were sent sequentially from the smallest number of packets (the sequential number of a packet can be obtained from the original ITP header information.) In this case, the receiver requested the server to retransmit the lost seven packets and five packets were retransmitted. The PSNR was improved from 22.21 dB to 28.22 dB. By comparing Figs. 15 and 16, we can see that the proposed method provides a better PSNR performance than ITP. Note that we can improve further the performance of the proposed method by applying the error concealment method of [1] simultaneously.

For the progressive JPEG image, the ITP with the error concealment does not improve the image quality as in the case of baseline JPEG image as seen from Fig. 18. Because the pixels used for interpolation were not correct one due to the progressive nature of the image format.The proposed method, on the other hand, requested fourteen packets of forty one lost-packets for retransmission according to the proposed method. During the preset time, only five packets could be retransmitted. The PSNR was improved from 23.6 dB to 26.8 dB. Note that the selection of packets for retransmission was easily done according to the proposed method and that no additional time or additional calculations were required for the selection.

Our results show that using our method improved the perceptual quality of the image by retransmitting only the ADUs of the most important level. Use of the proposed method could easily determine which packet was perceptually important, so selective requests for retransmission could be sent to the server without any computational cost.

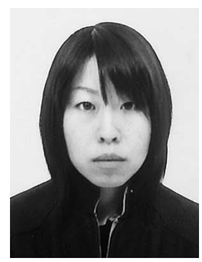
6. Conclusions

In this paper, we proposed an extension of the image transport protocol. The proposed method enables a server to mark the importance of the image ADUs according to the image format. Using this information, receivers can selectively request retransmission of the lost-packets with no computational cost. Our method can reduce the number of requests for retransmission without sacrificing the perceptual quality of received images. It is considered as a future work to apply the proposed method to JPEG2000 images.

References

[1] S. Raman, H. Balakrishnan, and M. Srinivasan, "ITP: An image transport protocol for the Internet," IEEE Trans. Netw., vol.10, no.3, pp.297–307, June 2002.
[2] W.R. Stevens, TCP/IP Ilustrated, vol.1, Addison-Wesley, 1994.
[3] F. Fitzek, A. Kopsel, A. Wolisz, and M.K.M. Reisslein, "Providing application-level QoS in 3g/4g wireless system: A comprehensive framework based on multirate CDMA," IEEE Wireless Commun., vol.9, no.2, pp.42–47, April 2002.
[4] R. Gupta, M. Chen, S. McCanne, and J. Walrand, "WebTP: A receiver-driven transport protocol for the Web," Proc. INFORMS2000 Telecommunications Conf., 2000.
[5] D.S. Taubman and M.W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards, and Practice, Kluwer Academic Publishing, 2001.
[6] H. Balakrishnan, H.S. Rahul, and S. Seshan, "An integrated congestion management architecture for Internet hosts," Proc. ACM SIGCOMM'99, pp.175–188, Sept. 1999.
[7] V. Jacobson, R. Braden, and D. Borman, TCP extensions for high performance, Internet engineering task force, RFC1323, May 1992.
[8] "Information technology—Digital compression and coding of continuous-tone still image: Requirements and guidelines," Int. Std. ISO/IEC IS-10918-1, 1994.
[9] "Information technology—Digital compression and coding of continuous-tone still image—Requirements and guidelines," Rec. ITU-T Rec.T.81, Sept. 1992.
[10] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/

Kiyoshi Nishikawa received the B.E., the M.E., and the D.E. degrees in electrical engineering from Tokyo Metropolitan University in 1990, 1992 and 1996 respectively. From 1992 to 1993, he was at the Computer Systems Laboratory, Nippon Steel Corp. as a researcher. Since 1993, he has been with Tokyo Metropolitan University as an Assistant Professor and he is currently a lecturer of the Department of Electrical Engineering. His research interest includes the digital communications, computer networks, and stochastic signal processing. He is a member of IEEE Signal Processing, Communications, and Vehicular Technology societies.



Takako Sasaki was born in Nagano, Japan in February 23, 1981. She received B.E. degree in Electronics and Information Engineering from Tokyo Metropolitan University in 2003. In 2003 she joined EPSON KOWA Corporation.

Hitoshi Kiya was born in Yamagata, Japan, on November 16, 1957. He received the B.E. and M.E. degrees in electrical engineering from Nagaoka University of Technology, Niigata, Japan, and the D.E. degree in electrical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 1980, 1982, and 1987, respectively. In 1982, he joined Tokyo Metropolitan University, where he is currently a Professor of Electrical Engineering, Graduate School of Engineering. He was a visiting researcher of the University of Sydney in Australia from Oct. 1995 to March 1996. His research interests are in digital signal processing, multirate systems, adaptive filtering, image processing, and efficient algorithms for VLSI implementation. He is an Associate Editor of Trans. of IEICE and served an Associate Editor of Trans. on Signal Processing of IEEE from 1998 to 2000. He is a Senior Member of the IEEE, and a member of the Image Electronics Engineers of Japan and the Institute of Television Engineers of Japan.