

Commission d'études ;Study Group;Comisión de Estudio} 15
Contribution tardive;Delayed Contribution ;Contribución tardía} **563(WP1/15)**

Texte disponible seulement en ;Text available only in;Texto disponible solamente en} **E**

Question: 4/15

SOURCE*: MATSUSHITA ELECTRIC INDUSTRIES, CO. LTD.

TITLE: G.HS.BIS: RETRANSMISSION ENHANCEMENT

ABSTRACT

Although a retransmission mechanism for G.hs was discussed during the development of G.hs, no formal proposals were made. This contribution provides some background, ideas, and a proposal for a retransmission mechanism for the next generation G.hs.

1. Introduction

One of the discussion areas of G.hs was the definition of message segmentation and error recovery via retransmission. In the current version of G.hs, if an error occurs in a frame (as indicated by the FCS), the G.hs session must be completely restarted. Since initialization procedures often involve several messages or transactions, restarting from the beginning represents a significant loss of information and time. What is needed is a method to allow recovery in the procedure by retransmitting the errored message frame instead of restarting.

The proposed retransmission method is described in §2. Some questions and scenarios related to the proposal are discussed in §3.

2. Proposal

2.1 Overview

What is proposed here is a retransmission mechanism for the next generation G.hs. The procedure is implemented as a backward compatible extension to the existing G.hs. It is implemented as a new message type is defined: "Request Retransmission" which is abbreviated as "RTX". If an HSTU-X receives an errored message frame during a session, the HSTU-X can request retransmission of the errored message with the request message (RTX) indicating the last correctly received message. Additionally the RTX message

¹ Contact: Stephen Palm
Matsushita Graphic Communication Systems

T: +81 3 5434 7078
F: +81 3 5434 7159
E: palm@kiwin.com

can suggest the length of the message frame to be used by the transmitter in order to help reduce the occurrence of frames with errors.

2.2 Format of the message

The RTX message has the format shown in Table 1.

Table 1. RTX Frame Format

Octet Content	Octet Index #
leading Flags	
Message type (RTX)	1
Version	2
Last Correctly Received Message (LCRM)	3
Multi Segment Frame Number (MSFN)	4
Suggested Frame Size (SFS)	5
Frame Check Sequence (FCS) (2 octets)	6 7
trailing Flags	

Description of Octet Contents

Message Type: The unique number of the RTX message type

Revision: indicates the version number of the transaction protocol

LCRM: This octet contains the message type code of the last correctly received message.

NULL: The NULL message code (FF₁₆) is used for the LCRM octet if an error free message has not been received in the session.

MSFN: This octet contains the segment index number of a message the has been segmented into more than one frames. The first segment or messages contained in one frame have a MSFN value of 0. The second segment has the MSFN value of 1, and so on. Although the segment frames are not explicitly numbered, the HSTU-R and HSTU-C must maintain internal counters.

SFS: This octet suggests to the remote side the length of subsequent message frames to be transmitted by the remote side. The value of the octet is defined as:

FF₁₆ Reserved for Future Use

00₁₆ No change of frame size suggested

00xx xxxx₂ Size of frame

2.3 Procedures:

- If an HSTU-x receives an errored message, it can send an RTX message instead of a NAK-EF. The LCRM field must contain the type of the last correctly received message. (An example transaction is shown in Figure 1.) The MSFN and SFS octets are encoded as described above.
- If an HSTU-x receives an errored multi segmented message, the MSFN field contains the message segment number. The first segment has a MSFN value of 0. The second segment has the value of 1, and so on. Although the segment frames are not explicitly numbered, the HSTU-R and HSTU-C must keep internal counters. (An example transaction is shown in **Figure 4**.)
- If an HSTU-x has not received an error free message during the handshaking session, the LCRM octet shall contain the NULL message type code. (An example session is shown in Figure 2.)

- If an HSTU-C receives an RTX message with LCRM set to NULL, it shall respond with a NAK-CD message to clear down (hangup) the session. (An example session is shown in Figure 3)
- If an HSTU-x receives three RTX messages in succession, it shall send a NAK-CD message to clear down (hangup) the session.
- An RTX message is not "acknowledged". The transaction proceeds as if the errored message and the RTX message were not sent.

Note: These procedures assume that the first message is always sent by the HSTU-R (as is currently specified in G.hs).

2.4 Examples:

Notes to the Figures:

An arrow indicates a successfully received message.

An "X" indicates a received message that is errored.

A box indicates the type of message sent.

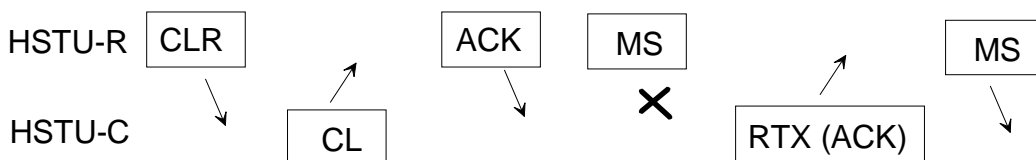


Figure 1. Type retransmission example

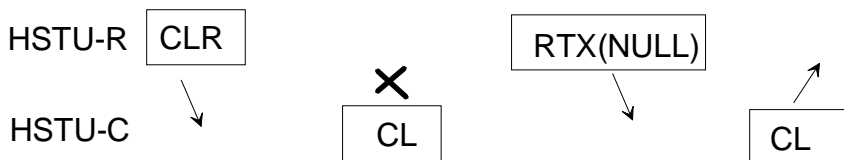


Figure 2. First HSTU-C message is errored example.

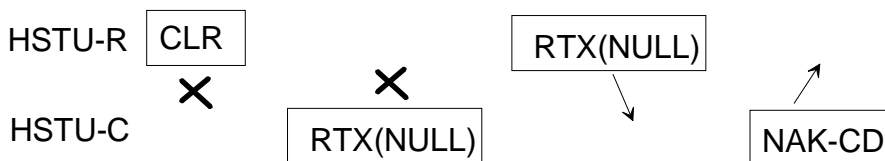


Figure 3. First HSTU-R message is errored example.

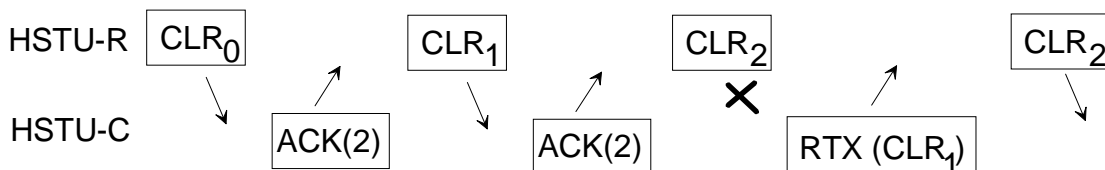


Figure 4. One Frame of multipart message is errored example

3. Problem and Scenario Discussion

During discussion on a retransmission method for G.hs, several questions were raised. These are presented here to familiarize readers with some of the issues related to retransmission. Some suggested answers are also included.

Is Frame Numbering necessary?

Only to correctly identify frames in a multi-segment message. Even if two message types had been sent since the last correctly received frame, the transmitter will know which frame to send.

How many times should we be allowed to ask for a resend? 1, 2, 3 or x? Are infinite loops OK?

T.30 (half duplex fax) allows 3 times and then it sends a DCN (disconnect).

Do we need to define another NAK that says, "I tried asking for resends (x times) but I just don't understand?"

No, We can simply use NAK-CD (Cleardown) after a (predetermined) number of times.

What is the criteria for deciding to send an NAK-EF (errored frame) versus RTX ?

Implementation dependent?

Or do we prohibit or discourage the use of NAK-EF?

Should we send a RTX in response to an invalid frame?

Possibly, since it would not disrupt the transmitting HSTU-X. If some noise on the line caused a invalid frame to be received, the transmitting HSTU-X would be notified that there is noise on the line, but it would also know that it is in sync with the receiving HSTU-X since the RTX message would contain the message type of the last frame that the transmitting HSTU-X had sent. Knowing that there is noise on the line may encourage the use of shorter frames.

Do we allow RTX to be sent in response to all types of messages including ACK and NAKs?

What are valid responses to a RTX? Are NAK-NR (Not Ready), NAK-NS (Not supported), NAK-NU acceptable?

Yes, with the current proposal, there is no confusion about which frame should be retransmitted.

Is the HSTU-x required to send the exact same message and of same length in response to a RTX?

No. Since the errored message type cannot be known for sure, the receiving HSTU-X should not make any assumptions about the contents of an errored frame. When the transmitting HSTU-X has been notified of an RTX, it can then decide to shorten the message length based on the Suggested Frame Size (SFS) octet. Additionally, the transmitting HSTU-X may decide to change the contents knowing that the channel is likely to have errors.

On the Q4/15 mail list, the following scenario was described. The proposal in the contribution handles this scenario. Basically, the problem was how to handle the situation where both HSTU-X each received an errored frame and then the channel allowed error free frames to be received.

1. An ATU-C sends some message.
2. Some kind of channel conditions exist that cause the ATU-R to receive a garbled message.
3. ATU-R sends a NAK-RS (resend)
4. The same channel conditions as #2 exist an the ATU-C also receives a garbled message.
5. ATU-C also sends a NAK-RS
6. Miracle has happened... (e.g. hair dryer is turned off) channel conditions had become better.
7. ATU-R now receives the NAK-RS... so he resends his last message a NAK-RS.

8. Well, ATU-C last message was also a NAK-RS... or did he mean the one before that?

What if they looped around a few times before conditions improved?

If link fails in one direction, it is not unlikely to fail in the other as well. Does RTX work in the case of double link failure?

Yes, see Figure 5 and Figure 6.

Can the case of RTX in response to ACK/NAK/RTX be adequately covered?

It seems all scenarios work...

NAK-EF

For two second generation machines, it is preferred to send RTX instead of NAK-EF.

If an HSTU-X does send a HAK-EF, it will be shutting down the link anyways. (Figure 7)

ACK-1

Works like any other message. (See Figure 8 and Figure 9)

ACK-2

Behaves like a segmented message (implicit numbering) (Figure 10)

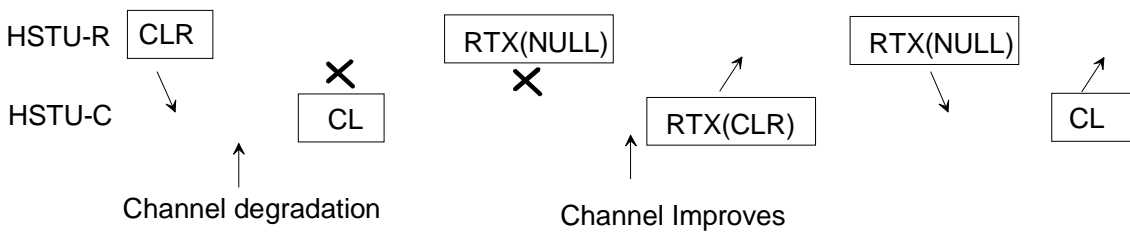


Figure 5. Multiple Errors Example #1

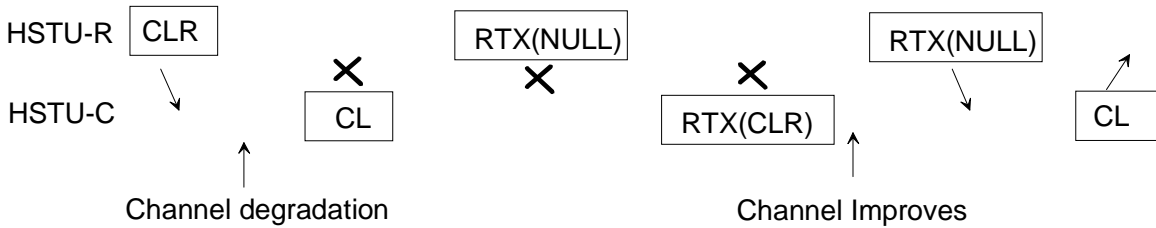


Figure 6. Multiple Errors Example #2

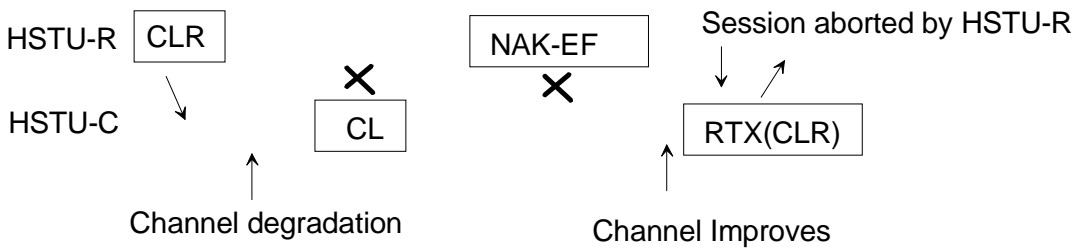


Figure 7. An HSTU-X uses an NAK-EF instead of RTX

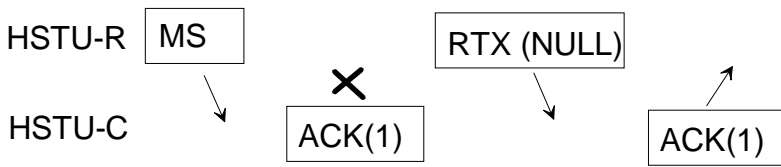


Figure 8. Recovery with ACK(1) message

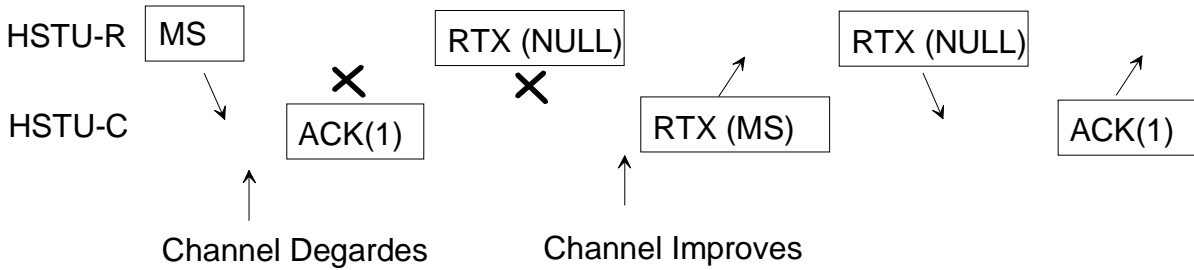


Figure 9. Recovery with ACK(1) message and double link failure

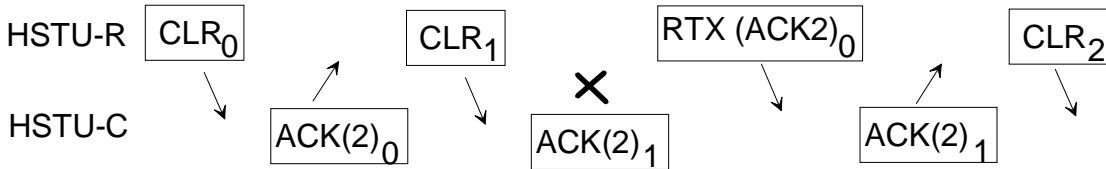


Figure 10. Recovery with ACK(2) messages

4. Summary:

The contribution shall be discussed under new enhancements for the next generation G.hs.

Expectations:

- Add Open Issue Item: Shall a retransmission mechanism be defined for G.hs.bis?
- If the general proposal seems agreeable to the group, specific text additions to G.hs will be proposed at the next meeting.