

July 21th, 2022

To Our Valued Customers;

SAISON Information Systems Co., Ltd.
HULFT Technical Support Center

Regarding the failure of the request acknowledgment process in HULFT8 for

zOS, MSP, and XSP Ver. 8.4.0 or later

We have confirmed a failure related to the request acknowledgment process in HULFT8 for zOS, MSP, and XSP Ver. 8.4.0 or later.

Please review the following information if you are currently using this service.

1. Occurrence Event

The request acknowledgment itself enters a state of infinite loop processing and continues to consume CPU resources.

2. Affect

This may occur when the request acknowledgment itself is accepting various services from the opposite side HULFT. This will not affect customers who are not using the request acknowledgment process.

Since this failure occurs only when a communication error occurs at a specific processing timing, we have not received any reports of the event occurring in customer environments since the release of the target version on March 5, 2019.

3. Cause of Occurrence

When a communication error occurs during a specific telegram transmission in the request acknowledgment process, the memory area allocated by the request acknowledgment is replaced by an improper value.

4. Occurrence Condition

When either of the following 【Condition 1】 or 【Condition 2】 is met

- 【Condition 1】 If all of the following are met
 - ✓ The request acknowledgment itself is executing one of the following processes
 - Send Request (Synchronization)
 - Remote Job Execution (Synchronization)
 - ✓ Some kind of communication error occurs in the following telegram transmission process from the general-purpose machine side
 - No communication timeout prevention telegrams
Communication telegram to prevent no communication timeout when accepting SEND and HULRJOB services with synchronization specification at request reception.
- 【Condition 2】 If all of the following are met
 - ✓ When the request acknowledgment itself receives a service request
 - ✓ Some kind of communication error occurs in the following telegram transmission process from the general-purpose machine side
 - CANCEL Telegram
Communication telegram notifying non-acceptance at the time when the request acknowledgment has accepted the service

Please refer to the "Appendix (Occurrence Conditions and Workaround)" for details on the settings on the HULFT side, which are the preconditions for this failure.

5. Applicable Versions

HULFT8 for zOS Ver.8.4.0~Ver.8.4.4

HULFT8 for MSP Ver.8.4.0~Ver.8.4.3

HULFT8 for XSP Ver.8.4.0~Ver.8.4.3

6. Future Support

- Release of Minor Revision Up Version
Date: September 9, 2022, is scheduled

7. For Inquiries Regarding This Announcement

Please contact your technical support service contractor.

End

Attachment (occurrence conditions / workarounds)

[Condition 1] Send Request (Synchronization)

In the following occurrence condition, the problem occurs when a particular communication error happens upon sending a telegram to prevent a non-communication timeout. The meaning of "a particular communication error" is a situation where a network communication fails due to forced termination or such on an issuer when sending a telegram to prevent a non-communication timeout.

Occurrence Condition

The problem occurs when all of the following conditions are met:

- 1 is specified in the Wait Synchronous Request (REQWAITMODE) on the system environment settings

REQWAITMODE=1: The synchronous transfer wait time specified by the Send Request command is enabled

REQWAITMODE=0: The synchronous transfer wait time is disabled, and it can wait forever (Default value)

- The synchronization is specified on a Send Request command issuing side.

- The synchronous transfer wait time is specified on a Send Request command issuing side

When utlrecv is "-w"

: Default value = "Utility Response Waiting Time" (UNIX/Linux) on the system environment settings file

Socket Lead Timeout (Windows) on the system environment settings file

When XRRCVREQ is "TIME="

: Default value = 36000 sec

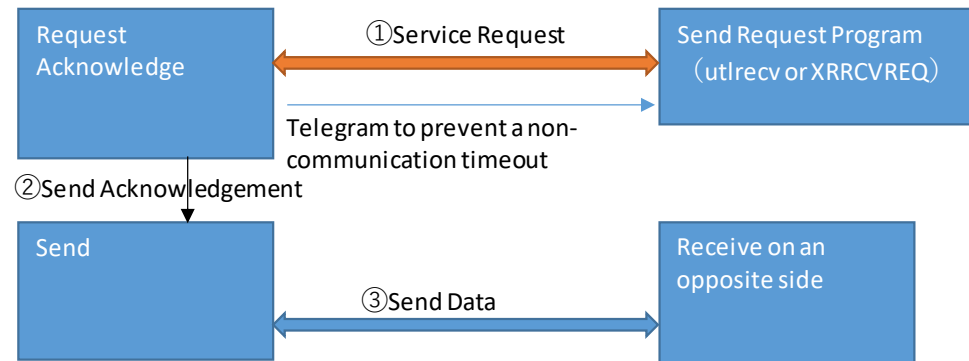
- The HULFT version on a send request command issuing side is Ver.6.3.0 or later.

Workaround

The problem can be avoided by specifying "0" in the Wait Synchronous Request (REQWAITMODE) on the system environment settings because it does not send a telegram to prevent a non-communication timeout.

However, depending on values in the system environment settings (socktime) or (TIMEOUT) on a request issuing side, send requests might time out.

Situation when the Problem Occurs



1. Acknowledging requests from Send Requests (Synchronization) on an opposite side's HULFT
2. Processing Send Acknowledgements by Request Acknowledge
3. Sending data to an opposite side's HULFT

-The socket communication on the above graph 1 lasts until the sending on #2 completes if it's synchronization.

However, suppose the sending on #3 lasts for a long time. In that case, a telegram to prevent the socket communication in #1 from time outing is the telegram to prevent a non-communication timeout.

-While data is sent in #1, a telegram to prevent a non-communication timeout is sent periodically to an opposite side's send request program from a request acknowledgment side.

-The interval in which a telegram to prevent a non-communication timeout is sent is an interval of -w upon processing send requests or 1/10 of the value of TIME=.

[Condition 1] Remote Job Execution (Synchronization)

In the following occurrence condition, the problem occurs when a particular communication error happens upon sending a telegram to prevent a non-communication timeout. The meaning of "a particular communication error" is a situation where a network communication fails due to forced termination or such on an issuer when sending a telegram to prevent a non-communication timeout.

Occurrence Condition

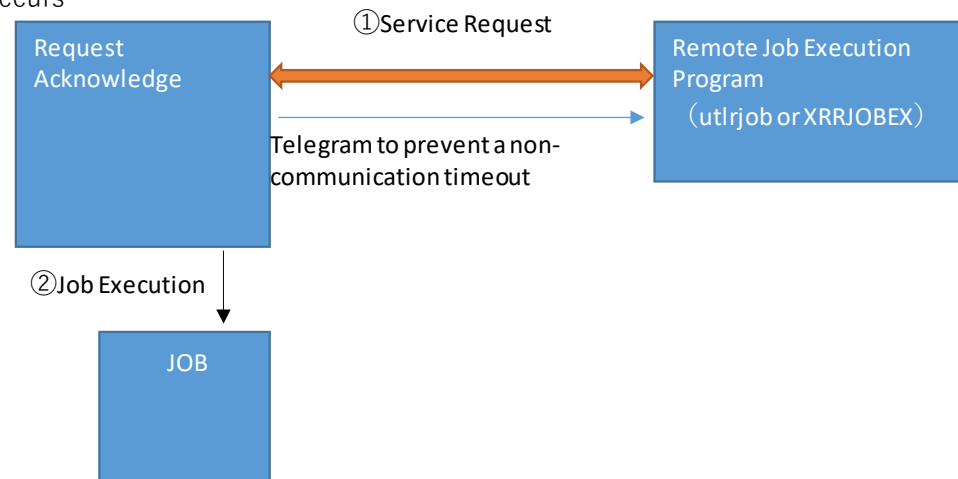
The problem occurs when all of the following conditions are met:

- Synchronization is specified on the side where the Remote Job Execution command is issued.
- the HULFT version on the side where the Remote Job Execution command is issued is Ver.6.3.0 or later.

Workaround

None.

Situation when the Problem Occurs



1. Accepting remote job (synchronization) requests by an opposite side's HULFT
2. Request Acknowledges submitting requested jobs

1. Accepting remote job (synchronization) requests by an opposite side's HULFT
2. Request Acknowledges submitting requested jobs

- The socket communication on the above graph 1 lasts until the executed jobs on #2 complete if it's synchronization. However, suppose the execution on #2 lasts for a long time. In that case, a telegram to prevent the socket communication in #1 from time outing is the telegram to prevent a non-communication timeout.
- While jobs are executed in #2, a telegram to prevent a non-communication timeout is sent periodically to an opposite side's sremote job execution program from a request acknowledgment side.
- The interval in which a telegram to prevent a non-communication timeout is sent is an interval of -w upon processing send requests or 1/10 of the value of TIME=.

When utlrjob is "-w"

: Default value = "Job Finish Waiting Time" (UNIX/Linux) on the system environment settings file
"Job Timeout" (Windows) on the system environment settings file

When XRRJOBEXT is "TIMEOUT="

: Default value = 300 sec

[Condition 2] In case of the Request Acknowledgment itself refuses service requests

In the following occurrence condition, the problem occurs when a particular communication error happens upon sending a CANCEL telegram. The meaning of "a particular communication error" is a situation where a network communication fails due to forced termination or such on an issuer when sending a CANCEL telegram.

Occurrence Condition

The problem occurs when one of the following conditions are met:

- A lack of REGION when service requests occur

The service requests represent the following:

1. Send Request (SEND)
2. Resend Request (RESEND)
3. Requests to view the post-receive job results (HULJOB)
4. Job Result Notification (HULSNDRC)
5. Remote Job Execution (HULRJOB)
6. HULFT Manager (HULADMIN)

- Prohibited errors when service requests occur

The prohibited error occurs in the following situation:

"1" (Enabled) is set in "Forced Strong Key Mode (STRONGKEYMODE)" on the system environment settings, and services are acknowledged.

However, an error occurs on a request issuer for one of the following reasons:

- The Request Issuer is using versions earlier than Ver.7
- "1" (Enabled) is not specified in "Forced Strong Key Mode (STRONGKEYMODE)" on the system environment settings

-Refuse errors when service requests occur

The refuse error occurs in the following service request acknowledgment:

1. Send Request (SEND)
2. Resend Request (RESEND)
3. Job Result Notification (HULSNDRC)

The refuse error occurs in the following situation:

-When information is registered in the host information

- "N" (Refuse) is set in "Allow Send Request/Resend Request (SENDPERMIT)" on the host information

- "N" (Refuse) is set in "Allow to Notify Job Result (HULSNDRCPERMIT)" on the host information

When information is not registered in the host information

- "0" (Continue request acknowledge processes) is set in "Request Acknowledge Host Check" on the system environment settings

- "N" (Refuse service acknowledgments) is set in "Allow Send/Resend Request from Unregistered Host (ALLOWSNDREQ)" on the system environment settings

- "N" (Refuse service acknowledgments) is set in "Allow Job Result Notification Request from Unregistered Host (ALLOWJOBRSLTNTFY)" on the system environment settings

- 1 (Refuse connections from a request issued host and cause an error) is set in

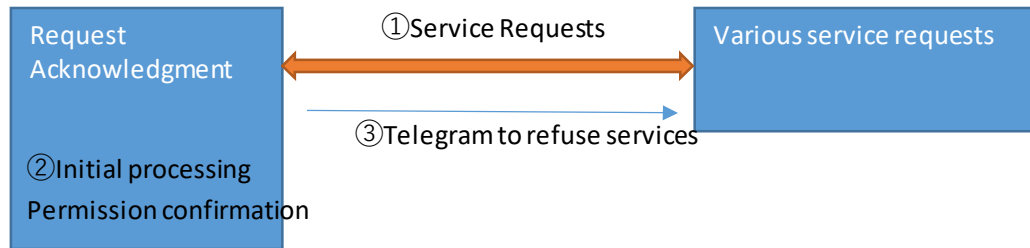
"Observe host check (APTHSTCHK)" on the system environment settings

Workaround

None.

Theoretically, the problem is not likely to occur in the above occurrence condition. However, a response telegram immediately after the socket gets established upon service requests is a CANCEL telegram. Therefore, the probability of this problem occurring is relatively low.

Situation when the Problem Occurs



1. Accepting service requests by an opposite side's HULFT
2. Establishing initial processing within the request acknowledgment and confirming the legitimacy of the requests
3. Processing service refuse toward the opposite side in case of an initial processing error or no permission.

-The socket communication in the above graph #1 is established by service requests from an opposite side's HULFT.

-Regarding the process in #2, in case of an initial processing error or service requests had no permission, a telegram sent to an opposite side's HULFT is a CANCEL telegram.

【Revision History】

July 21 th , 2022	First edition created
------------------------------	-----------------------

