
Question

[Mathieu Van Sevenant](#) · Aug 13, 2020

Connection reset while transferring HL7 through TCPOperation

Hello everyone 😊

We are facing what seems to be a network problem while transferring HL7 messages from Ensemble/Healthshare to a distant target through TCP/IP.

Here is the version of the system in any case it could be useful: Cache for Windows (x86-64) 2017.2.1 (Build 801U)
Wed Dec 6 2017 09:07:51 EST [HealthShare Modules:Core:14.02.2415 + Linkage Engine:15.03.9901]

Then the configuration of the operation:

Nom de la classe	EnsLib.HL7.Operation.TCPOperation
Description	Accepts HL7 Messages and forwards them to a remote IP address, reading a response HL7 Message object returned from the remote IP address if the GetReply setting is enabled.
Nom de classe adapt	EnsLib.HL7.Adapter.TCPOutboundAdapter
Description adapt	Outbound adapter for HL7 v2 over TCP
Partenaire métier	<input type="text"/> 🔍

Adresse IP
172.17.229.138

Port
9002

▼ Param connexion

Rester connecté
-1

Délai de connexion
20

Nouvel essai de reconnexion
0


Obtenir réponse
☒

Délai de réponse
80

Délai de lecture
40

Config SSL

▼



Interface locale


▼

Tramage

MLLP

▼

▼ Paramètres suppl

Planifier


Taille de pool

Actions code de réponse

Aucun échec si déconnecté
☐

Enregistrer les réponses

Séparateurs

Rechercher dans la classe de table

Codage par défaut des caractères

Intervalle de nouvel essai

Délai d'échec

Envoyer SuperSession
☐

Délai vitesse

NB high numbers for Read- and Response-Timeouts come from "long" transfers occurring sometimes, e.g. HL7 messages with about 600 segments, successfully transferred if we let them this "long" time.

Some messages in particular block the queue, and we don't understand why because they seem formatted the same way than other passed messages, neither they are the longest ones nor have special characters inside. Removing those messages from the queue allows the flow to be active anew, until another blocking message arrive.

At network level, we notice resets during the transfer of the blocking messages. Below an example with the reset occurring always after the transfer of the 5th chunk.

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Wireshark capture reset of the connection short-paging

Echier Editor View Alter Capture Analyzer Statistics Applications Telephony Wireless QoS Aids

Apply a display filter <<Ctrl>

No.	Time	Source	Destination	Protocol	Length	Info
11	18.724078	172.17.152.84	172.17.229.138	TCP	66	57104 → 9002 [SYN, ECH, CHR] Seq=0 Win=0192 Len=0 MSS=1460 WS=256 SACK_PERM=1
12	18.724241	172.17.229.138	172.17.152.84	TCP	66	9002 → 57104 [SYN, ACK, ECH] Seq=0 Ack=1 Win=0192 Len=0 MSS=1460 WS=256 SACK_PERM=1
13	18.726121	172.17.152.84	172.17.229.138	TCP	60	57104 → 9002 [ACK] Seq=1 Ack=1 Win=1051136 Len=0
14	18.737800	172.17.152.84	172.17.229.138	TCP	1514	57104 → 9002 [ACK] Seq=1 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
15	18.737806	172.17.152.84	172.17.229.138	TCP	1514	57104 → 9002 [ACK] Seq=1461 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
16	18.737903	172.17.229.138	172.17.152.84	TCP	54	9002 → 57104 [ACK] Seq=1 Ack=2921 Win=2102272 Len=0
17	18.738013	172.17.152.84	172.17.229.138	TCP	1514	57104 → 9002 [ACK] Seq=2921 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
18	18.738013	172.17.152.84	172.17.229.138	TCP	1514	57104 → 9002 [ACK] Seq=4301 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
19	18.738025	172.17.229.138	172.17.152.84	TCP	54	9002 → 57104 [ACK] Seq=1 Ack=5841 Win=2102272 Len=0
20	18.739757	172.17.152.84	172.17.229.138	TCP	1514	57104 → 9002 [ACK] Seq=5841 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
21	18.739966	172.17.152.84	172.17.229.138	TCP	54	57104 → 9002 [RST] Seq=5841 Win=0 Len=0
22	18.750511	172.17.229.138	172.17.152.84	TCP	54	9002 → 57104 [ACK] Seq=1 Ack=7302 Win=2102272 Len=0
23	18.743259	172.17.152.84	172.17.229.138	TCP	66	57554 → 9002 [SYN, ECH, CHR] Seq=0 Win=0192 Len=0 MSS=1460 WS=256 SACK_PERM=1
24	18.743336	172.17.229.138	172.17.152.84	TCP	66	9002 → 57554 [SYN, ACK, ECH] Seq=0 Ack=1 Win=0192 Len=0 MSS=1460 WS=256 SACK_PERM=1
25	18.750766	172.17.152.84	172.17.229.138	TCP	60	57554 → 9002 [ACK] Seq=1 Ack=1 Win=1051136 Len=0
26	18.756213	172.17.152.84	172.17.229.138	TCP	1514	57554 → 9002 [ACK] Seq=1 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
27	18.756293	172.17.152.84	172.17.229.138	TCP	1514	57554 → 9002 [ACK] Seq=1461 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
28	18.756304	172.17.229.138	172.17.152.84	TCP	54	9002 → 57554 [ACK] Seq=1 Ack=2921 Win=2102272 Len=0
29	18.756423	172.17.152.84	172.17.229.138	TCP	1514	57554 → 9002 [ACK] Seq=2921 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
30	18.756430	172.17.152.84	172.17.229.138	TCP	1514	57554 → 9002 [ACK] Seq=4301 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
31	18.756438	172.17.229.138	172.17.152.84	TCP	54	9002 → 57554 [ACK] Seq=1 Ack=5841 Win=2102272 Len=0
32	18.758067	172.17.152.84	172.17.229.138	TCP	1514	57554 → 9002 [ACK] Seq=5841 Ack=1 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
33	18.761240	172.17.152.84	172.17.229.138	TCP	54	57554 → 9002 [RST] Seq=5841 Win=0 Len=0
34	18.764609	172.17.229.138	172.17.152.84	TCP	54	9002 → 57554 [ACK] Seq=1 Ack=7301 Win=2102272 Len=0
35	178.761739	172.17.152.84	172.17.229.138	TCP	66	57902 → 9002 [SYN, ECH, CHR] Seq=0 Win=0192 Len=0 MSS=1460 WS=256 SACK_PERM=1
36	178.761821	172.17.229.138	172.17.152.84	TCP	66	9002 → 57902 [SYN, ACK, ECH] Seq=0 Ack=1 Win=0192 Len=0 MSS=1460 WS=256 SACK_PERM=1

Flags: 0x010 (ACK)
 Window size value: 4106
 [Calculated window size: 1051136]
 [Window size scaling factor: 256]
 Checksum: 0xbab7 [Unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 [SQ/NACK analysis]

```

0000  00 50 5d ad 2d ee 00 c8 8b 57 d2 40 00 00 05 02  |PV (~ ~ M @ E~|
0010  05 dc 3c 42 00 7d 06 e5 d5 ac 11 90 54 ac 11   |...~T...~|
0020  e5 8a df 10 23 2a 90 cd ef f9 5e 03 84 50 10   |...~V~P~|
0030  18 0a 97 97 30 38 5e 5e 41 0d 4e 54 45 7c      |...~P+A NTE]|
0040  31 7c 7c 47 45 4e 52 41 4c 2e 32 32 30 2e      |L[GENE RAL.220.|
0050  53 61 6e 07 20 63 6f 6d 70 6c 65 74 
```

In the logs of the operation, we find disconnections and reconnections (usually it works while the message in the log is "Discarding received non-HL7 data..."). The receiving party can set up to 10 connections at a time but it seems that everyone becomes blocked as soon as the blocking message occurs :

[illegible]

The main question is : why does the operation not just send the message completely, instead of resetting the connection each time ?

Best regards,

Mathieu

[#Business Operation](#) [#Caché](#) [#Ensemble](#) [#HealthShare](#)

Source

URL: <https://community.intersystems.com/post/connection-reset-while-transferring-hl7-through-tcoperation>