

Robotics Service Bus - Bug #1846

Client disconnect can cause server abort

04/16/2014 03:10 PM - J. Moringen

Status:	Resolved	Start date:	04/16/2014
Priority:	High	Due date:	
Assignee:		% Done:	100%
Category:	C++	Estimated time:	0.00 hour
Target version:	rsb-0.9		
Description			
We assume the following (needs to be reproduced):			
<ul style="list-style-type: none">- A client disconnects because of network problems (e.g. WLAN)- The server tries to close and clean up the connection, but fails for some (yet unknown) reason- While handling the error, the server tries to print something (probably an exception message string) but fails- The server aborts			
Firstly, we have to reproduce this and find out what goes wrong.			

Associated revisions

Revision a2ace880 - 05/08/2014 05:14 PM - J. Moringen

Do not abort on failed shutdown in `src/rsb/transport/socket/BusConnection.cpp`

fixes #1846

Apparently, an exception can be thrown during socket shutdown when the remote peer requests a shutdown and the local side is not quick enough with its shutdown.

This seems like incorrect behavior of the remote peer but has to be handled anyway since e.g. a crash of the remote program would have an identical effect.

- `src/rsb/transport/socket/BusConnection.cpp` (header): updated copyright
(`BusConnection::handleReadLength`): catch exceptions thrown by shutdown, print a warning but ignore otherwise

Revision 0ff18223 - 05/08/2014 05:14 PM - J. Moringen

Never abort when printing socket exceptions in `src/rsb/transport/socket/BusConnection.cpp`

refs #1846

- `src/rsb/transport/socket/BusConnection.cpp` (`safeSocketExceptionString`):
new function; return exception message string ignoring all exceptions
(`BusConnection::performSafeCleanup`): use `safeSocketExceptionString`
(`BusConnection::handleReadLength`): likewise

History

#1 - 04/17/2014 11:18 AM - T. Korthals

Setup

Laptop

Socketserver wurde mit dem "rsb-loggercpp0.9 --scope=/ --style=detailed" gestartet.
Ein Programm wurde gestartet, welches die Sensorinformationen vom BeBot liest und die berechneten Werte an die Aktor-Scopes sendet.
Es existieren keine "Bridges", es kommuniziert also ALLES über einen Socketserver.

BeBot

Auf dem BeBot läuft kein Socketserver.
Alle Socketverbindungen laufen über den Socketserver, welcher auf dem Laptop gestartet wurde
Es wurde 4 Programme gestartet, die jeweils die Sensor-/Aktoranbindung realisieren.

Verbindung

Kabelverbindung über USB (RNDIS virtuelle Ethernetschnittstelle)

Fehlerverursachung

Während der Laufzeit musste ich mittels "rsb-sendcpp0.9" an einen Scope senden, um ein Programm zu "faken".
Grundsätzlich funktioniert dies, jedoch traten dieses Mal folgende Fehler auf:

Mehrmaliges ausführen von "rsb-sendcpp0.9":

```
$ rsb-sendcpp0.9 /tabletop/BB3/standby void
1397485237667 rsb.transport.socket.busconnection [WARN]: Dangling bus pointer when trying to dispatch incoming event; closing connection
$ rsb-sendcpp0.9 /tabletop/BB3/standby void
1397485420165 rsb.transport.socket.busconnection [WARN]: Dangling bus pointer when trying to dispatch incoming event; closing connection
$ rsb-sendcpp0.9 /tabletop/BB3/standby void
1397485762535 rsb.transport.socket.busconnection [WARN]: Dangling bus pointer when trying to dispatch incoming event; closing connection
$ rsb-sendcpp0.9 /tabletop/BB3/standby void
1397485771831 rsb.transport.socket.busconnection [WARN]: Dangling bus pointer when trying to dispatch incoming event; closing connection
$ rsb-sendcpp0.9 /tabletop/BB3/standby void
1397485780809 rsb.transport.socket.busconnection [WARN]: Dangling bus pointer when trying to dispatch incoming event; closing connection
$ rsb-sendcpp0.9 /tabletop/BB3/standby void
```

Nach der letzten Ausführung bleibt rsb-sendcpp0.9 hängen.

Ausgabe des "rsb-loggercpp0.9" (Gleichzeitig Socketserver) in gdb

```
Event
Scope      /tabletop/BB3/finish/
Id         EventId[participantId = UUID[6e519015-9a37-411c-bb69-af1b60b7f565], sequenceNumber = 2036]
Type       bytearray
Origin     6e510915-9a37-411c-bb69-af1b60b7f565
Timestamps
Create     2014-Apr-14 16:29:40.050804+??:??
Send       2014-Apr-14 16:29:40.050811+??:??
Receive    2014-Apr-14 16:29:40.762579+??:??
Deliver    2014-Apr-14 16:29:40.762589+??:??
User-Infos
rsb.wire-schema void
Payload (bytearray, length 0)
terminate called after throwing an instance of
'boost::exception_detail::clone_impl<boost::exception_detail::error_info_injector<boost::system::system_error> >'
what(): shutdown: Transport endpoint is not connected
Payload (bytearray, length 0)
-----
Event
Scope      /act/steering/
Id         EventId[participantId = UUID[a5c3ba98-52d8-4ce3-8da0-bf306745d181], sequenceNumber = 135]
Type       bytearray
Origin     a5c3ba98-52d8-4ce3-8d0a-bf036745d181
Timestamps
Create     2000-Jan-01 01:19:42.357910+??:??
```

```

Program received signal SIGABRT, Aborted.
[Switching to Thread 0x7ffff3bb0700 (LWP 8217)]
0x00007ffff5fd6f77 in __GI_raise (sig=sig@entry=6) at ../nptl/sysdeps/unix/sysv/linux/raise.c:56
56 ../nptl/sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) bt
#0 0x00007ffff5fd6f77 in __GI_raise (sig=sig@entry=6) at ../nptl/sysdeps/unix/sysv/linux/raise.c:56
#1 0x00007ffff5fda5e8 in __GI_abort () at abort.c:90
#2 0x00007ffff68e26e5 in __gnu_cxx::__verbose_terminate_handler() () from /usr/lib/x86_64-linux-gnu/libstdc++.so.6
#3 0x00007ffff68e0856 in ?? () from /usr/lib/x86_64-linux-gnu/libstdc++.so.6
#4 0x00007ffff68e0883 in std::terminate() () from /usr/lib/x86_64-linux-gnu/libstdc++.so.6
#5 0x00007ffff68e0aae in __cxa_throw () from /usr/lib/x86_64-linux-gnu/libstdc++.so.6
#6 0x00007ffff7879e91 in void boost::throw_exception<boost::system::system_error>(boost::system::system_error const&) () from /usr/lib/librsb.so.0.9
#7 0x00007ffff7879eee in boost::asio::detail::do_throw_error(boost::system::error_code const&, char const*) () from /usr/lib/librsb.so.0.9
#8 0x00007ffff787a0b2 in boost::asio::basic_socket<boost::asio::ip::tcp, boost::asio::stream_socket_service<boost::asio::ip::tcp>>::shutdown(boost::asio::socket_base::shutdown_type) () from /usr/lib/librsb.so.0.9
#9 0x00007ffff7875f7a in rsb::transport::socket::BusConnection::shutdown() () from /usr/lib/librsb.so.0.9
#10 0x00007ffff7876ee4 in rsb::transport::socket::BusConnection::handleReadLength(boost::system::error_code const&, unsigned long) () from /usr/lib/librsb.so.0.9
#11 0x00007ffff787c343 in boost::asio::detail::read_op<boost::asio::basic_stream_socket<boost::asio::ip::tcp, boost::asio::stream_socket_service<boost::asio::ip::tcp>>, boost::asio::mutable_buffers_1, boost::asio::detail::transfer_all_t, boost::_bi::bind_t<void, boost::_mfi::mf2<void, rsb::transport::socket::BusConnection, boost::system::error_code const&, unsigned long>, boost::_bi::list3<boost::_bi::value<boost::shared_ptr<rsb::transport::socket::BusConnection>>, boost::arg<1> (*)(), boost::arg<2> (*)()>>>::operator()(boost::system::error_code const&, unsigned long, int) () from /usr/lib/librsb.so.0.9
#12 0x00007ffff787c736 in boost::asio::detail::reactive_socket_recv_op<boost::asio::mutable_buffers_1, boost::asio::detail::read_op<boost::asio::basic_stream_socket<boost::asio::ip::tcp, boost::asio::stream_socket_service<boost::asio::ip::tcp>>, boost::asio::mutable_buffers_1, boost::asio::detail::transfer_all_t, boost::_bi::bind_t<void, boost::_mfi::mf2<void, rsb::transport::socket::BusConnection, boost::system::error_code const&, unsigned long>, boost::_bi::list3<boost::_bi::value<boost::shared_ptr<rsb::transport::socket::BusConnection>>, boost::arg<1> (*)(), boost::arg<2> (*)()>>>::do_complete(boost::asio::detail::task_io_service*, boost::asio::detail::task_io_service_operation*, boost::system::error_code const&, unsigned long) () from /usr/lib/librsb.so.0.9
#13 0x00007ffff788cd76 in boost::asio::detail::task_io_service::run(boost::system::error_code&) () from /usr/lib/librsb.so.0.9
#14 0x00007ffff788d4e5 in boost::asio::io_service::run() () from /usr/lib/librsb.so.0.9
#15 0x00007ffff750b95b in ?? () from /usr/lib/libboost_thread.so.1.49.0
#16 0x00007ffff72e7f6e in start_thread (arg=0x7ffff3bb0700) at pthread_create.c:311
#17 0x00007ffff609a9cd in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:113

```

#2 - 04/30/2014 07:49 AM - J. Moringen

Thanks for the detailed report, we will look into it as soon as we can.

#3 - 04/30/2014 10:03 AM - J. Moringen

- Status changed from New to In Progress
- % Done changed from 0 to 50

#4 - 04/30/2014 11:26 AM - J. Moringen

- File 0001-Do-not-abort-on-failed-shutdown-in-src-rsb-transport.patch added
- File 0002-Never-abort-when-printing-socket-exceptions-in-src-r.patch added
- Status changed from In Progress to Feedback

I would like to address this by means of the attached patches. Any opinions?

#5 - 04/30/2014 02:15 PM - J. Wienke

For the second patch: Is it clear why retrieving the exception message might fail?

#6 - 04/30/2014 06:40 PM - J. Moringen

Johannes Wienke wrote:

| *For the second patch: Is it clear why retrieving the exception message might fail?*

I could not reproduce this aspect of the problem, but Sebastian MzB. reported that printing Boost.Asio exception messages can throw exceptions when TCP connections fail in uncommon ways.

#7 - 05/08/2014 01:29 PM - J. Wienke

Weird. Anyway, seems ok to me. I would add a comment to the safe exception method message to explain why this was added.

#8 - 05/08/2014 05:30 PM - J. Moringen

- *Status changed from Feedback to Resolved*

- *% Done changed from 50 to 100*

Applied in changeset rsb-cpp|commit:a2ace8801ae5cc80c96ce27988e086d1e15542c9.

Files

0001-Do-not-abort-on-failed-shutdown-in-src-rsb-transport.patch	2.01 KB	04/30/2014	J. Moringen
0002-Never-abort-when-printing-socket-exceptions-in-src-r.patch	2.26 KB	04/30/2014	J. Moringen