

NLSR - Bug #1815

NLSR is aborting randomly; stating "Key does not exist".

08/02/2014 12:44 AM - Syed Amin

Status: Closed	Start date: 08/02/2014
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	

Description

I updated NLSR last night and since then I am getting the following error msg randomly. Any idea what's the problem.

```
~pku$ terminate called after throwing an instance of 'ndn::SecPublicInfoSqlite3::Error'
what(): Key does not exist:/ndn/pku/%C1.Router/router1/NLSR/ksk-1406964298392
./runall.sh: line 87: 5162 Aborted
```

I pasted this error from the pku node, but a few other nodes also printed the same error. As it happens randomly, I cannot mention the steps to reproduce the error.

I am using the following commits:

```
NLSR: 443ad81ceb5f38376543afe61b61764cac951b41
NFD: 68bc1e0dce68c0a0ef98fe5e3fee15a4d8b21fc8
ndn-cxx: d1f34e864909bd07ce46e4906b4fa0cd06f0b6ca
```

History

#1 - 08/02/2014 07:02 AM - Lan Wang

What is at "./runall.sh: line 87"?

#2 - 08/02/2014 10:14 AM - Syed Amin

It is the name of the script that starts nfd and nlsr, as follows (only showing the relevant lines):

```
#start nfd
echo "-->starting nfd"
HOME=/tmp/ nfd-start >> ${LOG_DIR}/nfd${TIME_STAMP}.log 2>&1
```

```
#start nlsr
echo "-->starting nlsr"
HOME=/tmp/ nlsr -f ${CONFS_DIR}/${SRC}.conf &
```

where CONFS_DIR is a dir where I have all conf files and SRC is defined as:

```
SRC=`hostname|awk -F "." '{print $1}'`
```

#3 - 08/03/2014 05:51 AM - Lan Wang

Can you check whether the /ndn/pku/%C1.Router/router1/NLSR/ksk-1406964298392 exists in /tmp?

#4 - 08/04/2014 07:32 AM - A K M Mahmudul Hoque

can you wrap around the line 151 in nlsr.cpp with try catch blocks like below?

```
try
{
    m_keyChain.deleteIdentity(m_defaultIdentity);
}
catch (std::exception& e)
{
    std::cerr << "ERROR: " << e.what() << std::endl;
}
```

Let us know if that works.

#5 - 08/04/2014 08:35 AM - Syed Amin

I am now getting segfault and one weird error from glibc:

```
*** glibc detected *** nlsr: corrupted double-linked list: 0x0952eef0 ***
```

The segfault (and this error) happens after waiting for 1 to 2 minutes.

@ Lan, I checked the keys using `ndnsec-list -k`, and the mentioned key was not there.

#6 - 08/04/2014 08:39 AM - A K M Mahmudul Hoque

Can you run un gdb and see where it is coming from?

#7 - 08/04/2014 09:13 AM - A K M Mahmudul Hoque

if you are running hyperbolic routing you may run into the problem you mentioned.

Ashlesh got the same issue when running dry-run. So I am already investigating the issue.

#8 - 08/04/2014 10:17 AM - A K M Mahmudul Hoque

For the last segmentation fault issue you can try with this patch:

<http://gerrit.named-data.net/#/c/1098/>

#9 - 08/04/2014 11:27 AM - Syed Amin

The code at emulab was compiled without debugging support due to space constraint, so I cannot not run it in debugging mode. Were you able to reproduce this error at your end?

#10 - 08/04/2014 01:13 PM - A K M Mahmudul Hoque

Yes I found the problem when testing with hyperbolic dry_run. So working on that.

#11 - 08/04/2014 02:47 PM - A K M Mahmudul Hoque

You can try with this patch:

git fetch <http://akmhoque@gerrit.named-data.net/NLSR> refs/changes/98/1098/3 && git checkout FETCH_HEAD

It should solve the issue

#12 - 08/04/2014 06:01 PM - Syed Amin

Okay the Error I initially reported goes away if I give little delay between execution of nfd and nlsr. But segfault is still there, here is the core dump:

Program terminated with signal 11, Segmentation fault.

```
#0 nlsr::HyperRoutingTableCalculator::getHyperbolicDistance (this=0x0, pnlsr=..., pMap=..., src=0, dest=0) at
../src/route/routing-table-calculator.cpp:466
466 }/namespace nlsr
(gdb) bt
#0 nlsr::HyperRoutingTableCalculator::getHyperbolicDistance (this=0x0, pnlsr=..., pMap=..., src=0, dest=0) at
../src/route/routing-table-calculator.cpp:466
#1 0x0811e11b in nlsr::HyperRoutingTableCalculator::calculatePath (this=0xbfcfb88, pMap=..., rt=..., pnlsr=...) at
../src/route/routing-table-calculator.cpp:354
#2 0x0811f0a8 in nlsr::RoutingTable::calculateHypDryRoutingTable (this=0xbfc0300, pnlsr=...) at ../src/route/routing-table.cpp:138
#3 0x081d86dd in operator() (this=0xbfcfd4) at /usr/include/boost/function/function_template.cpp:71
#4 0x08120aab in operator() (a1=..., p=, this=) at /usr/include/boost/bind/mem_fn_template.hpp:165
#5 operator(), boost::_bi::list0> (f=..., this=, a=...) at /usr/include/boost/bind/bind.hpp:313
#6 operator() (this=) at /usr/include/boost/bind/bind_template.hpp:20
#7 boost::detail::function::void_function_obj_invoker0, boost::_bi::list2boost::_bi::value<nlsr::RoutingTable*, boost::reference_wrapper<nlsr::Nlsr > >,
void>::invoke (function_obj_ptr=...) at /usr/include/boost/function/function_template.hpp:153
#8 0x081d86dd in operator() (this=0xbfcfd4) at /usr/include/boost/function/function_template.hpp:760
#9 ndn::Scheduler::onEvent (this=0xbfc00a8, error=...) at ../src/util/scheduler.cpp:180
#10 0x081d9415 in operator() (p=, this=0xbfcfd4, a1=...) at /usr/include/boost/bind/mem_fn_template.hpp:165
#11 operator(), boost::_bi::list1 > (a=, f=..., this=0xbfcfd4)
at /usr/include/boost/bind/bind.hpp:313
#12 operator()boost::system::error_code (a1=..., this=0xbfcfd4) at /usr/include/boost/bind/bind_template.hpp:47
#13 operator() (this=0xbfcfd4) at /usr/include/boost/asio/detail/bind_handler.hpp:46
#14 asio_handler_invoke, boost::_bi::list2boost::_bi::value<ndn::Scheduler*, boost::arg > >, boost::system::error_code> > (function=...) at
/usr/include/boost/asio/handler_invoke_hook.hpp:64
#15 invoke, boost::_bi::list2boost::_bi::value<ndn::Scheduler*, boost::arg > >, boost::system::error_code>, boost::_bi::bind_t, boost::_bi::list2
boost::_bi::value<ndn::Scheduler*, boost::arg > > (function=..., context=...) at /usr/include/boost/asio/detail/handler_invoke_helpers.hpp:39
#16 boost::asio::detail::wait_handler, boost::_bi::list2boost::_bi::value<ndn::Scheduler*, boost::arg > > >::do_complete (owner=0x9a84470,
base=0x9b4fd30) at /usr/include/boost/asio/detail/wait_handler.hpp:68
```

#17 0x08158b1f in complete (owner=..., this=0x9b4fd30, bytes_transferred=, ec=...) at /usr/include/boost/asio/detail/task_io_service_operation.hpp:37
#18 do_run_one (ec=..., private_op_queue=..., this_thread=..., lock=..., this=) at /usr/include/boost/asio/detail/impl/task_io_service.hpp:366
#19 boost::asio::detail::task_io_service::run (this=0x9a84470, ec=...) at /usr/include/boost/asio/detail/impl/task_io_service.hpp:146
#20 0x08152d5a in run (this=) at /usr/include/boost/asio/impl/io_service.hpp:59
#21 ndn::Face::processEvents (this=0xbfc0084, timeout=..., keepThread=44) at ../src/face.cpp:335
#22 0x080f93eb in nlsr::Nlsr::startEventLoop (this=0xbfc0084) at ../src/nlsr.cpp:260
#23 0x080a0114 in main (argc=0, argv=0x0) at ../src/main.cpp:73

#13 - 08/04/2014 09:13 PM - A K M Mahmudul Hoque

Can you try with the last patch?

git fetch <http:// Gerrit named-data.net/NLSR> refs/changes/98/1098/4 && git checkout FETCH_HEAD

#14 - 08/05/2014 11:56 AM - A K M Mahmudul Hoque

Did you try this patch? Is it working now?

git fetch <http:// Gerrit named-data.net/NLSR> refs/changes/98/1098/5 && git checkout FETCH_HEAD

#15 - 08/05/2014 02:21 PM - Syed Amin

I tested it for a few tries and it didn't segfault. Though I was not able to do thorough testing due to some problems with emulab. I am fixing those issues now, and will update you. BTW, what is the difference between patch 4 and 5.

#16 - 08/05/2014 02:43 PM - Syed Amin

There is some problem going on with emulab in creating VLANs I guess, as sometimes a few nodes stop pinging (not ndnping) others. There is a scheduled maintenance of emulab tonight, so I think I won't be able to check it further. I'll try again tomorrow morning. But the good thing is it is not segfaulted yet for the available nodes.

#17 - 08/05/2014 02:46 PM - A K M Mahmudul Hoque

Good to know that.

There is not that much difference. It was rebased on the other commit.

#18 - 08/06/2014 07:13 AM - A K M Mahmudul Hoque

The segmentation fault was coming from accessing uninitialized memory. It was triggered when NLSR was configured to calculate routing table by hyperbolic method for dry run.

#19 - 08/06/2014 09:34 AM - Syed Amin

But why it was segfaulting after Patch 3?

#20 - 08/06/2014 09:45 AM - Syed Amin

What is the cause of the actual bug report? Delaying execution of nlsr fixed that issue, so my rough guess is the problem is somewhat similar to the one that we had with shared storage, but not sure.

#21 - 08/06/2014 10:03 AM - A K M Mahmudul Hoque

I added code that introduced that error in patch 3. That was corrected in later patches.

#22 - 08/10/2014 07:07 AM - A K M Mahmudul Hoque

segmentation fault of comment 6 of this thread was resolved by following way:

When NLSR tries to get some co-ordinate but it has not received the corresponding LSA yet it was getting this segfault. So added another check if LSA exists then tries to get the co-ordinate. Which solved the issue.

#23 - 08/11/2014 09:45 AM - A K M Mahmudul Hoque

- Status changed from New to Closed