[NFD] Get the list of nodes and associated links inside strategy

Anju K James

Thu 10/10/2019 5:32 PM

To: nfd-dev@lists.cs.ucla.edu <nfd-dev@lists.cs.ucla.edu>
Cc: Dan Ameme <danameme@nmsu.edu>

Hello All,

We are using ndnSIM 3.7 to simulate the network in Ubuntu 18.04.2 LTS and stuck with the following requirement.

Requirement: Get the list of all nodes in topology and all the links associated with each node inside strategy.(NFD/daemon/fw).

Example:

```
+----------+      1Mbps      +--------+      1Mbps      +----------+
| consumer | <------------> | router | <------------> | producer |
+----------+        10ms     +--------+        10ms     +----------+
```

In the above point to point topology, I would like to get the following result.

GetNodeList() : 0,1,2
GetLinks(0) : 0-1
GetLinks(1) : 1-0, 1-2
GetLinks(2) : 2-1

I have tried the following three options. But could not meet the requirement. Could you please suggest a method to resolve this issue?

option 1: ns3::ndn::GlobalRouter -> GetIncidencies()

```
+    for (ns3::NodeList::Iterator node = ns3::NodeList::Begin(); node != ns3::NodeList::End(); node++) {
+        ns3::Ptr<ns3::ndn::GlobalRouter> source = (*node)->GetObject<ns3::ndn::GlobalRouter>();
+
+        if (source == 0) {
+            std::cout << "\n===Node: " << (*node)->GetId() << " does not export GlobalRouter interface";
+            continue;
+        }
+        else{
+            //TODO: source is always 0. So not able to get the graph edges.
+            std::cout << "\n****Node: " << (*node)->GetId() << " export GlobalRouter interface";
+            ns3::ndn::GlobalRouter::IncidencyList graphEdges = source->GetIncidencies();
+        }
+    }
+
```

Problem : ns3::ndn:: GlobalRouter object is always zero and could not call GetIncidencies() api.

Option 2: ns3::AnnotatedTopologyReader::GetLinks()

```
+    std::cout << "Linksize:       " << ns3::AnnotatedTopologyReader::GetLinks().size();
+    ns3::NodeContainer nodes = ns3::AnnotatedTopologyReader::GetNodes();
+    std::cout << "# of nodes:      " << ns3::AnnotatedTopologyReader::GetNodes().GetN();
```

Problem: Returns zero as result from all APIs.

Option3: Indirect method using ns3::ndn::NetDeviceTransport

```
+    fib::NextHopList::const_iterator it = nexthops.end();
+    for (fib::NextHopList::const_iterator it = nexthops.begin(); it != nexthops.end(); ++it) {
+
+        Face& outFace = it->getFace();
+        // if (outFace.getId() == inFace.getId()) {
+        //    continue;
+        // }
+        std::cout <<"\n+++++++++++Found next hop: " << it->getFace().getId();
+        auto transportIn = dynamic_cast<ns3::ndn::NetDeviceTransport*>(outFace.getTransport()); //Get Transport for outFace
+        if (transportIn != nullptr) {
+            ns3::Ptr<ns3::PointToPointNetDevice> ndOut = transportIn->GetNetDevice()->GetObject<ns3::PointToPointNetDevice>();
+            ns3::Ptr<ns3::Channel> channelOut = ndOut->GetChannel();
+            ns3::Ptr<ns3::PointToPointChannel> ppChannelOut = ns3::DynamicCast<ns3::PointToPointChannel>(channelOut);
+            ns3::Ptr<ns3::NetDevice> ndOut1 = ppChannelOut->GetDevice(0);
+            ns3::Ptr<ns3::NetDevice> ndOut2 = ppChannelOut->GetDevice(1);
+            currentNodeId = std::to_string(ndOut1->GetNode()->GetId()) + " " + std::to_string(ndOut2->GetNode()->GetId());
+            std::cout <<"\n+++++++++++ currentNodeId : " << currentNodeId << std::endl;
+        }
+
```

Problem: I could get the current nodeId. But not the list of all nodes and associated links.

Regards,
Anju K. James |