**Question Form**

Questioner: Xiaogeng Xu

Affiliation: Beijing University of Posts and Telecommunications

Operation System: Ubuntu 12.04

Boost Version: 1.48

Submit Date: 2014-05-28

**Question**

Why does the constant value of a customized parameter of *Interest* change after one-hop transmission?

1. **Question Description**

We defined a customized parameter to *Interest*, and traced this parameter during simulation. The parameter was supposed to be constant during the simulation. However, we found that the parameter was set to 0 after one-hop transmission. We wondered that the parameter was rewritten before transmission, maybe when the *Interest* is packaged.

Therefore, our question is as follows:

* 1. Whether parameter is reset before transmitting *Interest*?
  2. How to give value to a customized parameter and keep this value constant during its lifetime?
  3. If more codes need to be edit to set a parameter of Interest, which files are supposed to be edit?

1. **Problem Reproduction**

The problem can be reproduced according to the following procedure.

* 1. Announce ‘m\_icid’ as a customized parameter to *Interest* in ndn-interest.h. The red bold characters are added.

------------------------------- ndn-interest.h -----------------------------------------

private: //line 236

Ptr<Name> m\_name; ///< @brief Interest name

uint8\_t m\_scope; ///< @brief 0xFF not set, 0 local scope, 1 this host, 2 immediate neighborhood

Time m\_interestLifetime; ///< @brief InterestLifetime

uint32\_t m\_nonce; ///< @brief Nonce. not used if zero

uint8\_t m\_nackType; ///< @brief Negative Acknowledgement type

Ptr<Exclude> m\_exclude; ///< @brief Exclude filter

Ptr<Packet> m\_payload; ///< @brief virtual payload

mutable Ptr<const Packet> m\_wire;

**uint32\_t m\_icid; ///< @brief add m\_icid (Community ID) to *Interest***

------------------------------------------------------------------------------------

* 1. Two functions are defined into ndn-interest.cc. The red bold characters are added.

------------------------------- ndn-interest.cc ---------------------------------------------

Ptr<const Exclude> //line 147

Interest::GetExclude () const

{

return m\_exclude;

{

**void**

**SetCommunityId(uint32\_t icid); //set icid**

**uint32\_t**

**GetCommunityId() const; //read icid**

------------------------------------------------------------------------------------

* 1. Call ‘SetCommunityId’ in ndn-consumer.cc to set value to ‘icid’. The red bold characters are added.

------------------------------- ndn-consumer.cc --------------------------------------------

Ptr<Interest> interest = Create<Interest> (); //line 201

interest->SetNonce (m\_rand.GetValue ());

interest->SetName (nameWithSequence);

interest->SetInterestLifetime (m\_interestLifeTime);

**interest->SetCommunityId (1); // set icid=1**

------------------------------------------------------------------------------------

* 1. Add commands in ndn-forwardingstrategy.cc to print ‘nonce’ and ‘icid’ of the received *Interest*. The red bold characters are added.

------------------------------- ndn-forwardingstrategy.cc ---------------------------

void

ForwardingStrategy::OnInterest (Ptr<Face> inFace, Ptr<Interest> interest)

{

NS\_LOG\_FUNCTION (inFace << interest->GetName ()); //line 145

**std::cout<<"interest nonce: " <<interest->GetNonce() <<std::endl;**

**std::cout<<"interest community ID:"<<interest->GetCommunityId()<<std::endl; // print ‘nonce’ and ‘icid’ of the received *Interest***

m\_inInterests (interest, inFace);

------------------------------------------------------------------------------------

* 1. Run ndn-simple.cc (unchanged). The simulation result is shown as below:

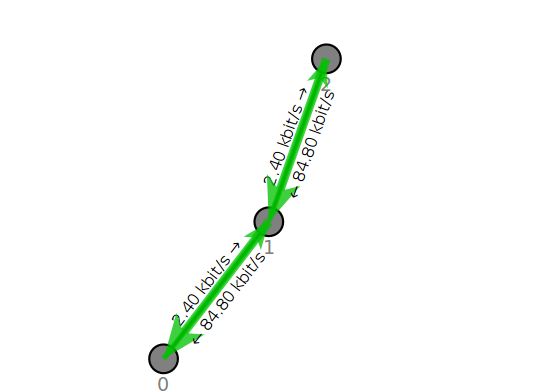


Fig.1 The Topology Output of ndn-simple.cc

|  |
| --- |
| Node 0 |
| Node 1 |
| C:\Users\Sunny\Desktop\QQ截图20140528153252.pngNode 2 |

Fig. 2 Output of nonce and community ID of the received *Interest*

Fig. 2 shows that the community ID was 1, just as being set in program. However, the community ID of the same Interest change back to 0 in both Node 2 and Node 3. We wonder that the community ID was rewritten to 0 when the Interest is re-packaged. Is it true? If it is, where is the value reset?

* 1. In order to find out the locations that related to setting values such as ‘name’, ‘nonce’, and ‘scope’ etc. We try to look for programs which include functions of ‘SetXXX’. The result feedback of our search is shown as follows. It presents that several codes do ‘SetXXX’ in our simulation platform. Does it mean that once we create a new parameter to *Interest*, we need to update all these files? Or maybe, there are one or some codes that actually set value, and we are supposed to edit those files only? Which ones?

-----------------------------Search Result-------------------------------------------------------

./src/ndnSIM/examples/custom-apps/custom-app.cc

./src/ndnSIM/examples/custom-apps/dumb-requester.cc

./src/ndnSIM/apps/ndn-consumer.cc

./src/ndnSIM/apps/ndn-consumer-zipf-mandelbrot.cc

./src/ndnSIM/test/ndnSIM-pit.cc

./src/ndnSIM/test/ndnSIM-serialization.cc

./src/ndnSIM/test/ndnSIM-fib-entry.cc

./src/ndnSIM/ndn.cxx/ndn-api-face.cc

./src/ndnSIM/model/wire/ccnb/wire-ccnb-interest.cc

./src/ndnSIM/model/wire/ndnsim.cc

------------------------------------------------------------------------------------