

ndnSIM - Task #2371

Port TcpFace and UdpFace to the new code base

01/11/2015 06:32 PM - Alex Afanasyev

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	model	Estimated time:	0.00 hour
Target version:			
Description			

History

#1 - 02/16/2015 04:41 PM - Alex Afanasyev

- Status changed from New to In Progress

(Assigned to Steven Collison)

#2 - 02/16/2015 08:35 PM - Alex Afanasyev

- Assignee set to Steven Collison

#3 - 02/18/2015 08:28 AM - Steven Collison

- I plan on trying to port ndn-udp-face first since it's simpler
- I'll use ndn-net-device-face as a baseline

Questions:

- To receive data, should I be registering a receive callback with the underlying transport mechanism(ns3 socket?)
- What is the point of TcpBoundaryHeader in ndn-tcp-face? Is it used for debugging or do we actually need to add an extra header ? (<https://github.com/NDN-Routing/ndnSIM/blob/master-v1/plugins/ip-faces/ndn-tcp-face.cc>)

#4 - 02/18/2015 09:17 AM - Spyros Mastorakis

Hello Steven,

1) To my understanding, the answer is yes, you will need to use the ns3 socket class, but you will have to replace the existing Face class with the Face class of NFD and, in general, overload the methods of the nfd::Face class. For example, the RegisterProtocolHandlers and UnRegisterProtocolHandlers methods do not exist anymore.

2) To my understanding again by looking at the source code, Alex has made some tricks (or "magic" as he likes to call it :) and actually sends two NS3 packets. The first one contains the segment length at its header and the second one is the actual TCP segment:

<https://github.com/NDN-Routing/ndnSIM/blob/master-v1/plugins/ip-faces/ndn-tcp-face.cc#L172-L177>

I guess that he did that to make the sending the receiving process easier.

@Alex: feel free to correct me, if I have said something wrong. Actually, I have not worked enough with these TCP, IP and UDP faces in order to be 100% sure that the things I have said are fully correct.

#5 - 02/18/2015 11:04 AM - Alex Afanasyev

- To receive data, should I be registering a receive callback with the underlying transport mechanism(ns3 socket?)
Yes, though I don't really understand this question. With TCP face, you need to communicate through tcp tunnel, not raw network interface like in NetDeviceFace.
- What is the point of TcpBoundaryHeader in ndn-tcp-face? Is it used for debugging or do we actually need to add an extra header ? (<https://github.com/NDN-Routing/ndnSIM/blob/master-v1/plugins/ip-faces/ndn-tcp-face.cc>)
This could be my question for you in cs118 :) This additional header was useful with old packet format (as there was no indication about message length in the packet format itself). With new packet format, it is no longer needed.

#6 - 02/18/2015 09:31 PM - Steven Collison

I noticed NetDeviceFace increments a FwHopCountTag before sending a packet. Should the Udp/Tcp Faces also increment the hop count?
<https://github.com/NDN-Routing/ndnSIM/blob/master/model/ndn-net-device-face.cpp#L84-L87>

#7 - 02/18/2015 09:48 PM - Alex Afanasyev

Yes. FwHopCount needs to be increased.

#8 - 02/18/2015 11:43 PM - Steven Collison

- File *ndn-udp-face.cc* added
- File *ndn-udp-face.hpp* added
- % Done changed from 0 to 20

Here's a starting implementation for UdpFace

#9 - 02/18/2015 11:57 PM - Alex Afanasyev

Don't upload code to redmine. Upload to our Gerrit (gerrit.named-data.net)

#10 - 02/19/2015 12:42 PM - Steven Collison

- File deleted (*ndn-udp-face.cc*)

#11 - 02/19/2015 12:42 PM - Steven Collison

- File deleted (*ndn-udp-face.hpp*)

#12 - 05/29/2015 02:01 PM - Spyros Mastorakis

- Assignee changed from Steven Collison to Yuanzhi Gao

#13 - 01/26/2016 01:27 PM - Alex Afanasyev

- Assignee deleted (*Yuanzhi Gao*)

#14 - 01/26/2016 01:27 PM - Alex Afanasyev

- Status changed from *In Progress* to *New*
- % Done changed from 20 to 0

#15 - 02/01/2016 06:04 PM - Alex Afanasyev

- Target version deleted (*2.1*)

To follow up on this issue. There is no immediate plan to implement this feature. Effects of overlay simulations (such as varying overlay link bandwidths, cross traffic, etc.) is better simulated directly using Loss and Delay models on p2p or mcast links.