ns-3 Direct Code Execution (DCE) UMIP Manual

Release 1.10

Direct Code Execution project

Mar 09, 2018

Contents

- Doxygen: Documentation of the public APIs

Contents:

CHAPTER 1

Introduction

The UMIP (Usagi-Patched Mobile IPv6 stack) support on DCE enables the users to reuse routing protocol implementations of UMIP. UMIP now supports Mobile IPv6 (RFC3775), Network Mobility (RFC3963), Proxy Mobile Ipv6 (RFC5213), etc, and can be used these protocols implementation as models of network simulation. It reduces the time of re-implementation of the model, and potentially improve the result of the simulation since it already "actively running" in the real world.

Unlike the quagga support of DCE, UMIP support requires Linux kernel integration of DCE (ns-3-linux) because of the dependencies of UMIP, interacts with Linux kernel.

1.1 Current Status (2012/6/1)

UMIP support on DCE does not fully support all the environment that DCE has. The following shows the limited availability of each protocol.

	Advanced Mode (ns-3-linux)	Remarks
Mobile IPv6	OK	
NEMO	OK	
Proxy Mobile IPv6	(N/A)	under dev.
Dual-Stack MIP v6	(N/A)	under dev.

CHAPTER 2

Getting Started

2.1 Prerequisite

UMIP support on DCE requires several packages: autoconf, automake, flex, git-core, wget, g++, libc-dbg, bison, indent, pkgconfig, libssl-dev, libsysfs-dev

You need to install the correspondent packages in advance.

```
$ sudo apt-get install git-core (in ubuntu/debian)
```

or

```
$ sudo yum install git (in fedora)
```

2.2 Building ns-3, DCE, DCE-Quagga, and DCE-UMIP

To install ns-3-dce-umip, you can use **bake** as an installation tool as follows.

```
$ hg clone http://code.nsnam.org/bake bake
$ export BAKE_HOME=`pwd`/bake
$ export PATH=$PATH:$BAKE_HOME
$ export PYTHONPATH=$PYTHONPATH:$BAKE_HOME
```

then build ns-3-dce with umip:

```
mkdir dce
cd dce
bake.py configure -e dce-linux-|version| -e dce-umip-|version|
bake.py download
bake.py build
```

note that dce-umip-1.10 is the DCE umip module version 1.10. If you would like to use the development version of the module, you can specify **dce-umip-dev** as a module name for bake.

For more information about ns-3-dce core, please refer the DCE manual.

Then you can try an example of ns-3-dce-umip as follows:

```
$ cd source/ns-3-dce
$ ./test.py -s dce-umip
...
PASS: TestSuite dce-umip 11.474 s
1 of 1 tests passed (1 passed, 0 skipped, 0 failed, 0 crashed, 0 valgrind errors)
```

You can see the above PASSed test if everything goes fine. Congrats!

2.3 Examples

2.3.1 Basic

```
$ cd source/ns-3-dce
$ ./waf --run dce-umip-nemo
```

if everything goes fine, you would see the encapsulated ICMP echo request and reply packets from the generated pcap file (dce-umip-nemo-***.pcap).

```
 $ tcpdump -r dce-umip-nemo-3-0.pcap -n proto ipv6
 reading from file dce-umip-nemo-3-0.pcap, link-type IEEE802_11 (802.11)
 09:00:20.000212 IP6 2001:1:2:7:200:ff:fe00:5 > 2001:1:2:3::1: IP6␣
↪2001:1:2:5:200:ff:fe00:8 > 2001:1:2:6::7: ICMP6, echo request, seq 1, length 64
 09:00:20.000834 IP6 2001:1:2:3::1 > 2001:1:2:7:200:ff:fe00:5: IP6 2001:1:2:6::7 >␣
↪2001:1:2:5:200:ff:fe00:8: ICMP6, echo reply, seq 1, length 64
 09:00:21.000722 IP6 2001:1:2:7:200:ff:fe00:5 > 2001:1:2:3::1: IP6␣
↪2001:1:2:5:200:ff:fe00:8 > 2001:1:2:6::7: ICMP6, echo request, seq 2, length 64
 09:00:21.001345 IP6 2001:1:2:3::1 > 2001:1:2:7:200:ff:fe00:5: IP6 2001:1:2:6::7 >␣
↪2001:1:2:5:200:ff:fe00:8: ICMP6, echo reply, seq 2, length 64
 09:00:25.000769 IP6 2001:1:2:7:200:ff:fe00:5 > 2001:1:2:3::1: IP6␣
↪2001:1:2:5:200:ff:fe00:8 > 2001:1:2:6::7: ICMP6, echo request, seq 6, length 64
 09:00:25.001392 IP6 2001:1:2:3::1 > 2001:1:2:7:200:ff:fe00:5: IP6 2001:1:2:6::7 >␣
↪2001:1:2:5:200:ff:fe00:8: ICMP6, echo reply, seq 6, length 64
 09:00:26.000816 IP6 2001:1:2:7:200:ff:fe00:5 > 2001:1:2:3::1: IP6␣
↪2001:1:2:5:200:ff:fe00:8 > 2001:1:2:6::7: ICMP6, echo request, seq 7, length 64
 09:00:26.001438 IP6 2001:1:2:3::1 > 2001:1:2:7:200:ff:fe00:5: IP6 2001:1:2:6::7 >␣
↪2001:1:2:5:200:ff:fe00:8: ICMP6, echo reply, seq 7, length 64

(snip)
$
```

You will also see the packet exchange between Home Agent and Mobile Router from the generated pcap file.

```
09:00:06.780000 IP6 (hlim 63, next-header unknown (60) payload length: 80)
2001:1:2:7:200:ff:fe00:5 > 2001:1:2:3::1: DSTOPT (padn)(homeaddr: 2001:1:2:3::1000)
mobility: BU seq#=26515 AH lifetime=300(padn)(alt-CoA:␣
↪2001:1:2:7:200:ff:fe00:5)(padn)(type-0x06: len=18)
(snip)
09:00:07.784000 IP6 (hlim 64, next-header Routing (43) payload length: 40)
2001:1:2:3::1 > 2001:1:2:7:200:ff:fe00:5: srcrt (len=2, type=2, segleft=1, rsv=0x0,␣
↪[0]2001:1:2:3::1000)
```

```
mobility: BA status=0 seq#=26515 lifetime=296(padn)
(snip)
09:00:07.788943 IP6 (hlim 63, next-header unknown (60) payload length: 32)
2001:1:2:7:200:ff:fe00:5 > 2001:1:2:3::1: DSTOPT (padn)(homeaddr:␣
↪2001:1:2:3::1000)[bad icmp6 cksum f70e!]
ICMP6, mobile router solicitation, length 8, id 0xacc7
09:00:07.788943 IP6 (hlim 64, next-header Routing (43) payload length: 64)
2001:1:2:3::1 > 2001:1:2:7:200:ff:fe00:5: srcrt (len=2, type=2, segleft=1, rsv=0x0,␣
↪[0]2001:1:2:3::1000) [bad icmp6 cksum f70e!]
ICMP6, mobile router advertisement, length 40, id 0xacc7
          prefix info option (3), length 32 (4): 2001:1:2:3::/64, Flags [onlink,␣
↪auto], valid time 298s, pref. time 148s
            0x0000:  40c0 0000 012a 0000 0094 0000 0000 2001
            0x0010:  0001 0002 0003 0000 0000 0000 0000
```

Binding Update (BU) and Binding Acknowledgment packets are exchanged, follows mobile router advertisement generated by UMIP implementation.

2.4 Configuration Manual

In order to use UMIP in ns-3, users need to define in the scenario via ns3::Mip6dHelper.

```
#include "ns3/mip6d-helper.h"

int main (int argc, char *argv[])
{
  Mip6dHelper mip6d;

  // Home Agent configuration
  mip6d.AddHaServedPrefix (ha.Get (0), Ipv6Address ("2001:1:2::"), Ipv6Prefix (48));
  mip6d.EnableHA (ha);
  mip6d.Install (ha);

  // Mobile Router (NEMO) configuration
  for (uint32_t i = 0; i < mr.GetN (); i++)
    {
      mip6d.AddMobileNetworkPrefix (mr.Get (i), Ipv6Address (mnps->at (i).c_str ()),␣
↪Ipv6Prefix (64));
      mip6d.AddHomeAgentAddress (mr.Get (i), Ipv6Address ("2001:db8:deaf:beaf::1"));
      mip6d.AddHomeAddress (mr.Get (i), Ipv6Address ("2001:1:2:3::1000"), Ipv6Prefix␣
↪(64));
      mip6d.AddEgressInterface (mr.Get (i), "sim0");
    }
  mip6d.EnableMR (mr);
  mip6d.Install (mr);
}
```

CHAPTER 3

Modifying DCE UMIP

3.1 Customizing Helper

At this moment, only a limited configuration of UMIP is implemented in the Mip6dHelper. For example, if you wanna configure the "route optimization" for triangle route by Mobile IPv6, you do have to extend Mip6dHelper (mip6d-helper.cc) to generate the following configuration for example.

```
DoRouteOptimizationMN enabled
```

3.2 Customizing Binary

If you wanna extend the protocol by modifying the source code of mip6d, your extended binary should be located at the directory "ns-3-dce/build/bin_dce".

CHAPTER 4

FAQ

(TBA)