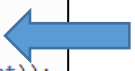
Network Coding Simulation in User Plane using NS3

- We observed that the GTP-U type of communication in NS3 is working only towards the upload direction.
- Communication from Enodes-to-SGW is point-to-point link type communication, which is defined in NS3.
  (We don't think, the download (From SGW-to-Enodeb) is using the GTP-U)
- The steps that we followed are below.

Step-1:

- We modified the function that is responsible for sending the GTP-U packets in Enodeb side.
- The function is in the Lena EPC model of the NS3.
  - **File Name**: epc-enb-application.cc
  - **URL Link**: https://www.nsnam.org/doxygen/epc-enb-application_8cc_source.html
  - **Name of the Function:** EpcEnbApplication::SendToS1uSocket

```
void
EpcEnbApplication::SendToS1uSocket (Ptr<Packet> packet, uint32_t teid)
{
  NS_LOG_FUNCTION (this << packet << teid <<  packet->GetSize ());
  GtpuHeader gtpu;
  gtpu.SetTeid (teid);
  // From 3GPP TS 29.281 v10.0.0 Section 5.1
  // Length of the payload + the non obligatory GTP-U header
  gtpu.SetLength (packet->GetSize () + gtpu.GetSerializedSize () - 8);
  packet->AddHeader (gtpu);
  uint32_t flags = 0;
  m_s1uSocket->SendTo (packet, flags, InetSocketAddress (m_sgwS1uAddress, m_gtpuUdpPort));
}
```



Modification in here, Before sending

Step-2:

- We modified the function that is responsible for receiving the GTP-U packets in SGW side.
- The function is in the Lena EPC model of the NS3.
  - **File Name**: epc-sgw-application.cc
  - **URL Link**: https://www.nsnam.org/doxygen/epc-sgw-application_8cc_source.html
  - **Name of the Function:** EpcSgwApplication::RecvFromS1uSocket

```
void
EpcSgwApplication::RecvFromS1uSocket (Ptr<Socket> socket)
{
  NS_LOG_FUNCTION (this << socket);
  NS_ASSERT (socket == m_s1uSocket);
  Ptr<Packet> packet = socket->Recv ();
  GtpuHeader gtpu;
  packet->RemoveHeader (gtpu);
  uint32_t teid = gtpu.GetTeid ();

  SendToS5uSocket (packet, m_pgwAddr, teid);
}
```

Step-3:

- We added packet loss to the point-to-point link.
- The function is in the src and poin-to-point model of the NS3.
    - **File Name**: point-to-point-helper.cc
    - **URL Link**: https://www.nsnam.org/doxygen/point-to-point-helper_8cc_source.html
    - **Name of the Function:** as an attribute

Step-4:

- We modified the .cc file to perform TCP test. (default, it is running UDP test.)
- The function is in the Lena EPC model of the NS3.
    - **File Name**: lena-simple-epc.cc
    - **URL Link**: https://www.nsnam.org/doxygen/lena-simple-epc_8cc_source.html
    - **Name of the Function:** main function

```
for (uint32_t u = 0; u < ueNodes.GetN (); ++u)
  {
    if (!disableDl)
      {
        PacketSinkHelper dlPacketSinkHelper ("ns3::UdpSocketFactory", InetSocketAddress (Ipv4Address::GetAny (), dlPort));
        serverApps.Add (dlPacketSinkHelper.Install (ueNodes.Get (u)));

        UdpClientHelper dlClient (ueIpIface.GetAddress (u), dlPort);
        dlClient.SetAttribute ("Interval", TimeValue (interPacketInterval));
        dlClient.SetAttribute ("MaxPackets", UintegerValue (1000000));
        clientApps.Add (dlClient.Install (remoteHost));
      }

    if (!disableUl)
      {
        ++ulPort;
        PacketSinkHelper ulPacketSinkHelper ("ns3::UdpSocketFactory", InetSocketAddress (Ipv4Address::GetAny (), ulPort));
        serverApps.Add (ulPacketSinkHelper.Install (remoteHost));

        UdpClientHelper ulClient (remoteHostAddr, ulPort);
        ulClient.SetAttribute ("Interval", TimeValue (interPacketInterval));
        ulClient.SetAttribute ("MaxPackets", UintegerValue (1000000));
        clientApps.Add (ulClient.Install (ueNodes.Get(u)));
      }

    if (!disablePl && numNodePairs > 1)
      {
        ++otherPort;
        PacketSinkHelper packetSinkHelper ("ns3::UdpSocketFactory", InetSocketAddress (Ipv4Address::GetAny (), otherPort));
        serverApps.Add (packetSinkHelper.Install (ueNodes.Get (u)));

        UdpClientHelper client (ueIpIface.GetAddress (u), otherPort);
        client.SetAttribute ("Interval", TimeValue (interPacketInterval));
        client.SetAttribute ("MaxPackets", UintegerValue (1000000));
        clientApps.Add (client.Install (ueNodes.Get ((u + 1) % numNodePairs)));
      }
  }
```

- Modification in here,
- Test is UDP in default
- Convert it to TCP test.
- Can not be done by simply writing TCPclientHelper

Step-5:

- Copy modified lena-simple-epc.cc to /Scratch
- Compile with WAF