Session 7. LTE in ns-3

김준석
Multimedia & Wireless Networking Laboratory, SNU
jskim14@mwnl.snu.ac.kr

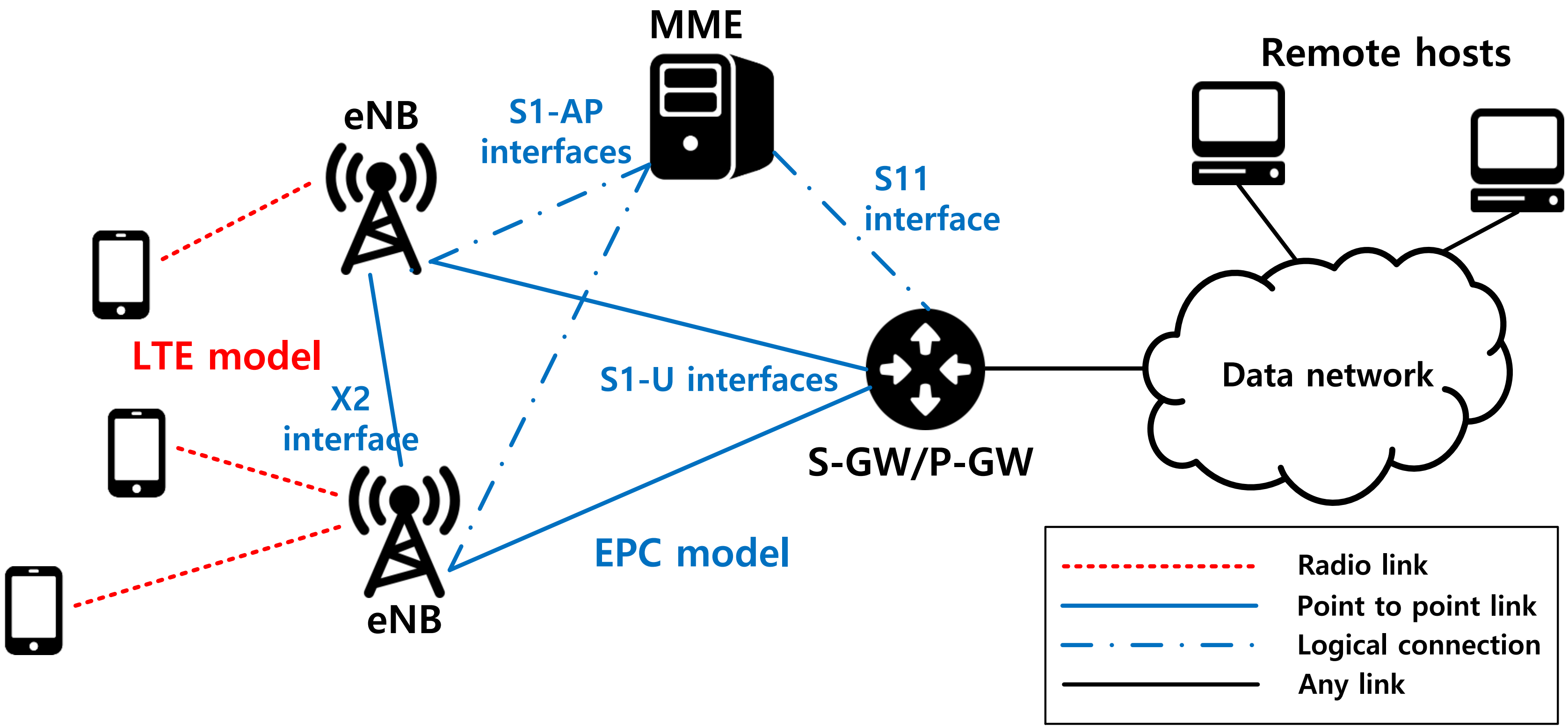
Contents

- LTE Module
  - Overview
  - LTE Protocol in ns-3
  - Function Flow
- Simulation Examples
  - Packet Transmission Example (Downlink)
  - More Exercises
    - Configurable Parameters
    - Uplink Transmission
    - Multi-cell Scenario

https://www.nsnam.org/docs/release/3.29/models/ns-3-model-library.pdf

Overview

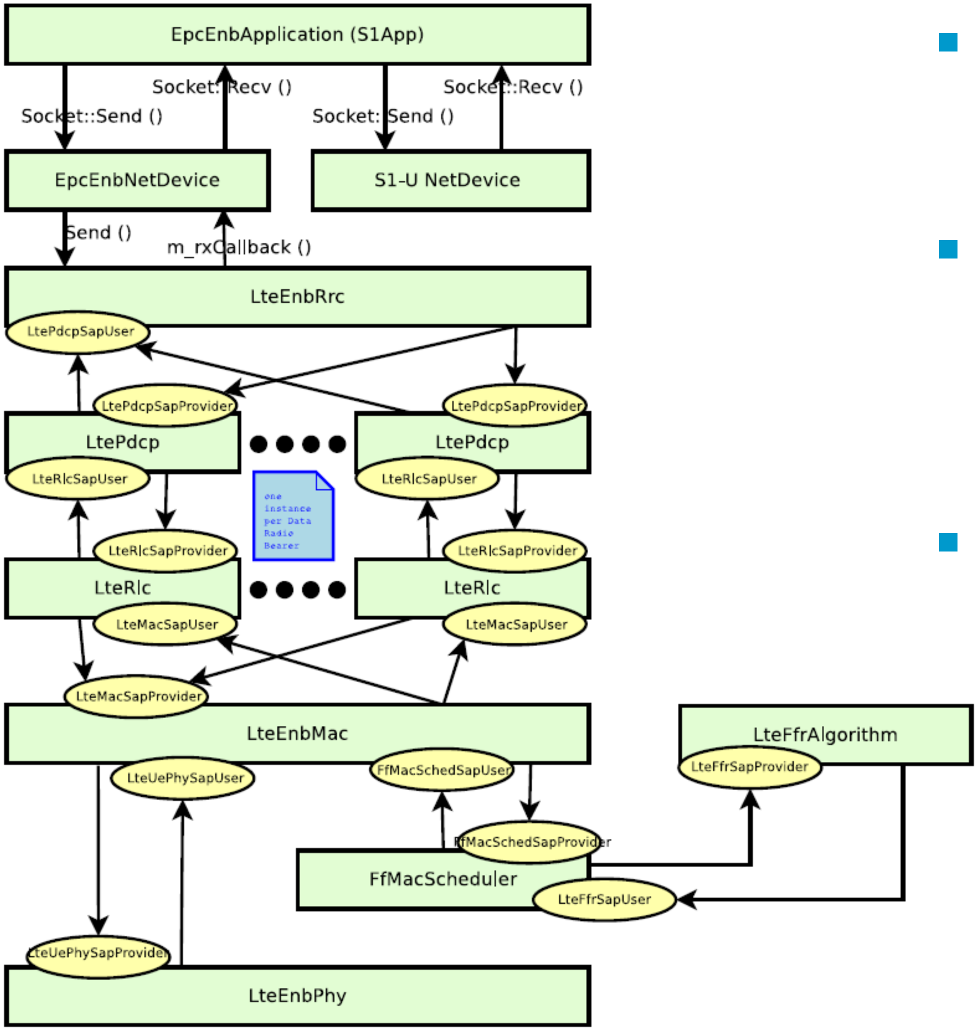
LTE-EPC Simulation Model



Overview of EPC Model

- Main object of EPC model
  - Providing means for simulation of **end-to-end IP connectivity** over LTE model
- Design choices
  - Single S-GW/P-GW node
    - Not supporting inter-S-GW mobility
  - User plane protocols used between eNBs and S-GW/P-GW → modeled accurately
  - Control plane protocols → modeled in a simplified way

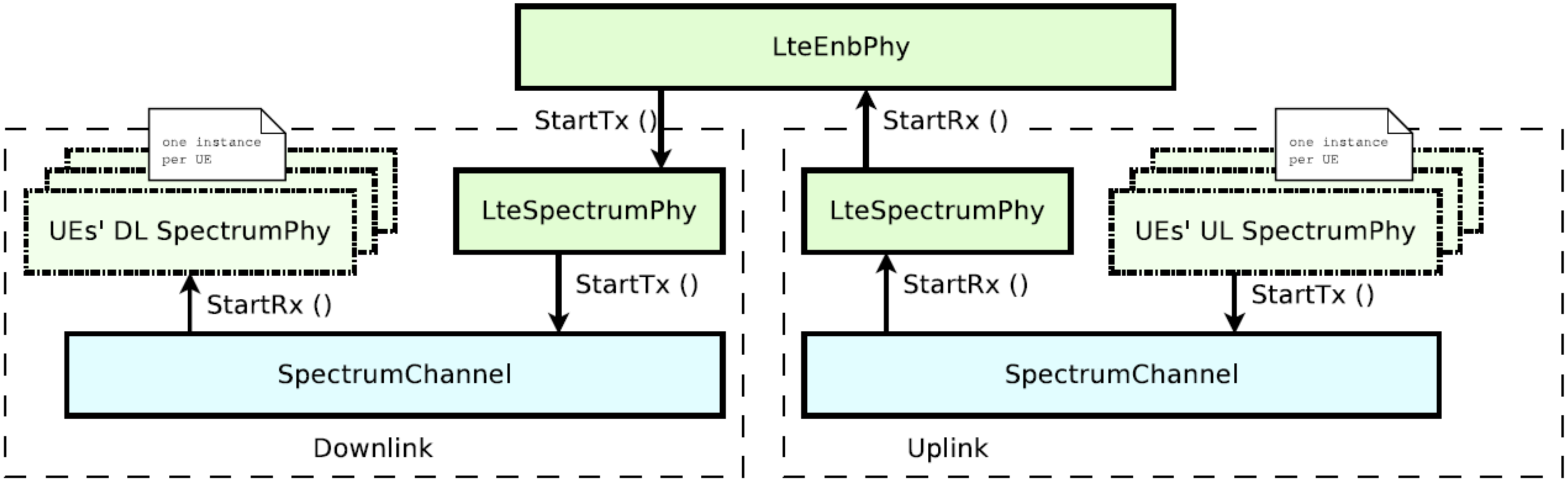
Protocol Architecture



- Protocol
  - RRC, PDCP, RLC, MAC, and PHY
- PDCP ~ RLC
  - One interface per data radio bearer (DRB)
- Service access point (SAP)
  - Provider and user
  - E.g., LtePdcpSapUser
    - RRC method called by PDCP
  - E.g., LtePdcpSapProvider
    - PDCP method called by RRC

PHY and Channel Architecture

- For eNB



  - src/lte/model/lte-spectrum-phy.cc
  - src/spectrum/model/multi-model-spectrum-channel.cc

LTE Protocol in ns-3

PDCP and RLC Model

- PDCP model
  - Transfer of Sequence number (SN) status for handover
  - Unsupported features
    - Header compression/decompression
    - Ciphering/Deciphering
- RLC model
  - TM, UM, AM modes / SM mode
  - Segmentation, Reassembly

MAC Model

- Resource allocation model
  - **Allocation type 0 / Localized mapping approach**
  - Example:
    - Downlink 3 MHz, 15 Resource blocks (RBs)
    - RBG size = 2
    - Total number of RBG = 8
    - If # of UE=3,
    - Resource allocation information

      UE 1 (10101001) UE 2 (01000010) UE 3 (00010100)

Channel Bandwidth (MHz)	Maximum Number of Resource Blocks (Transmission Bandwidth Configuration)	Maximum Occupied Bandwidth (MHz)
1.4	6	1.08
3	15	2.7
5	25	4.5
10	50	9.0
15	75	13.5
20	100	18.0

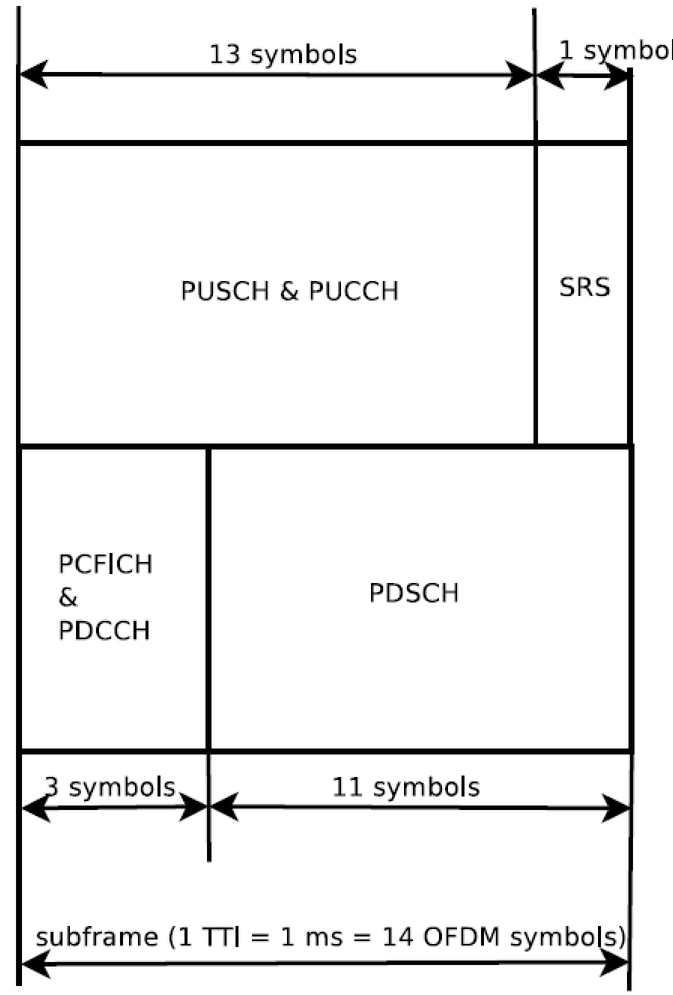
RBG

	1		2		3		4		5		6		7		8
RB	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

System Bandwidth N_{RB}^{DL}	RBG Size (P)
≤10	1
11 – 26	2
27 – 63	3
64 – 110	4

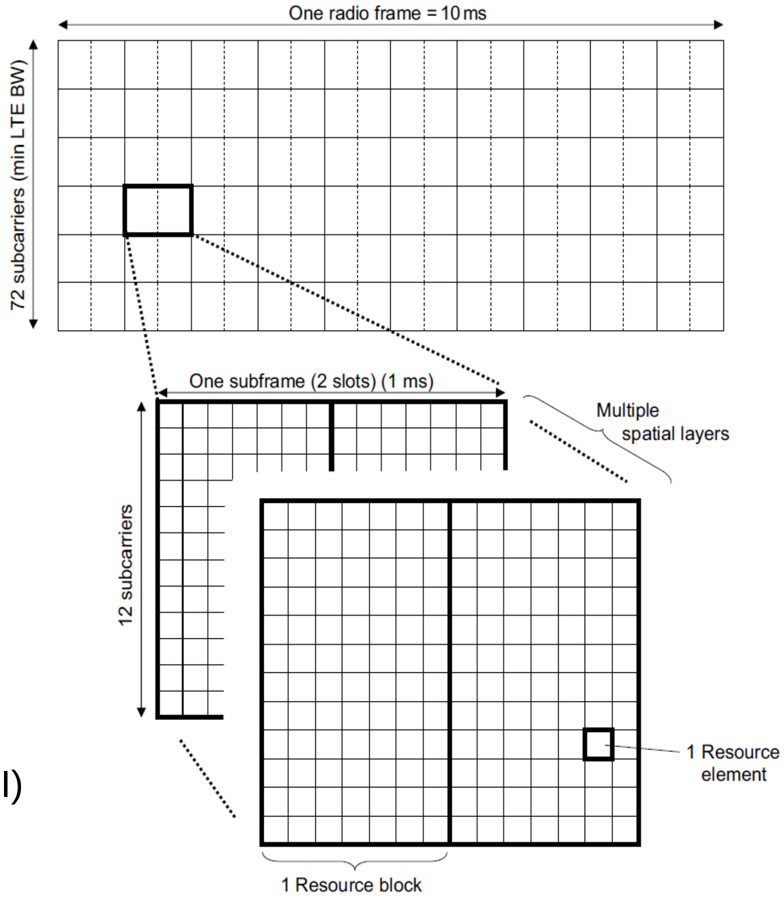
Scheduler and PHY Model

- Implemented schedulers
  - Downlink
    - Round Robin (RR) , Proportional Fair (PF), etc
  - Uplink
    - Round Robin (RR)
- PHY
  - Only **FDD** is modeled / **CQI feedback**
  - Granularity
    - Frequency domain: 1 Resource Block (RB)
    - Time domain: 1 Transmission Time Interval (TTI)
  - Supports different bandwidths per eNB



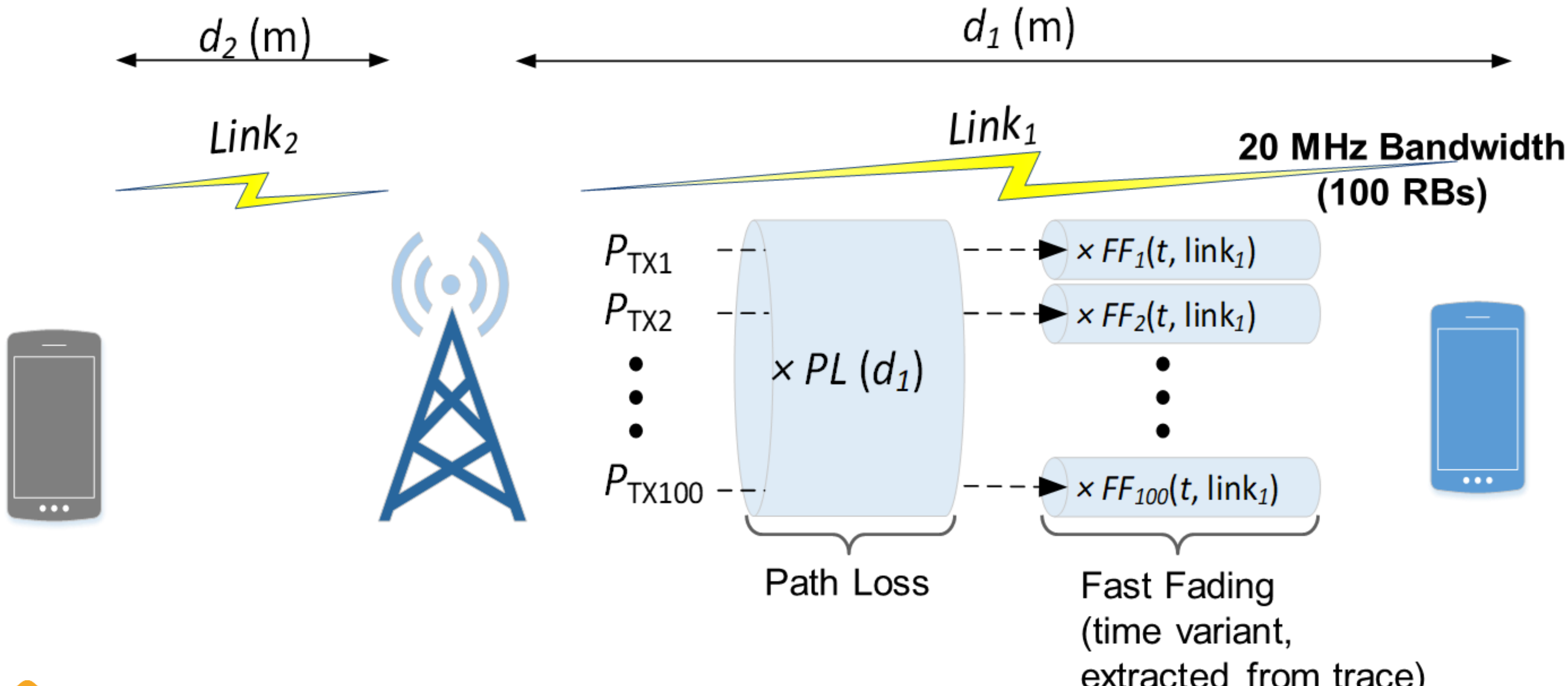
Scheduler and PHY Model

- Implemented schedulers
  - Downlink
    - Round Robin (RR) , Proportional Fair (PF), etc
  - Uplink
    - Round Robin (RR)
- PHY
  - Only **FDD** is modeled / **CQI feedback**
  - Granularity
    - Frequency domain: 1 Resource Block (RB)
    - Time domain: 1 Transmission Time Interval (TTI)
  - Supports different bandwidths per eNB



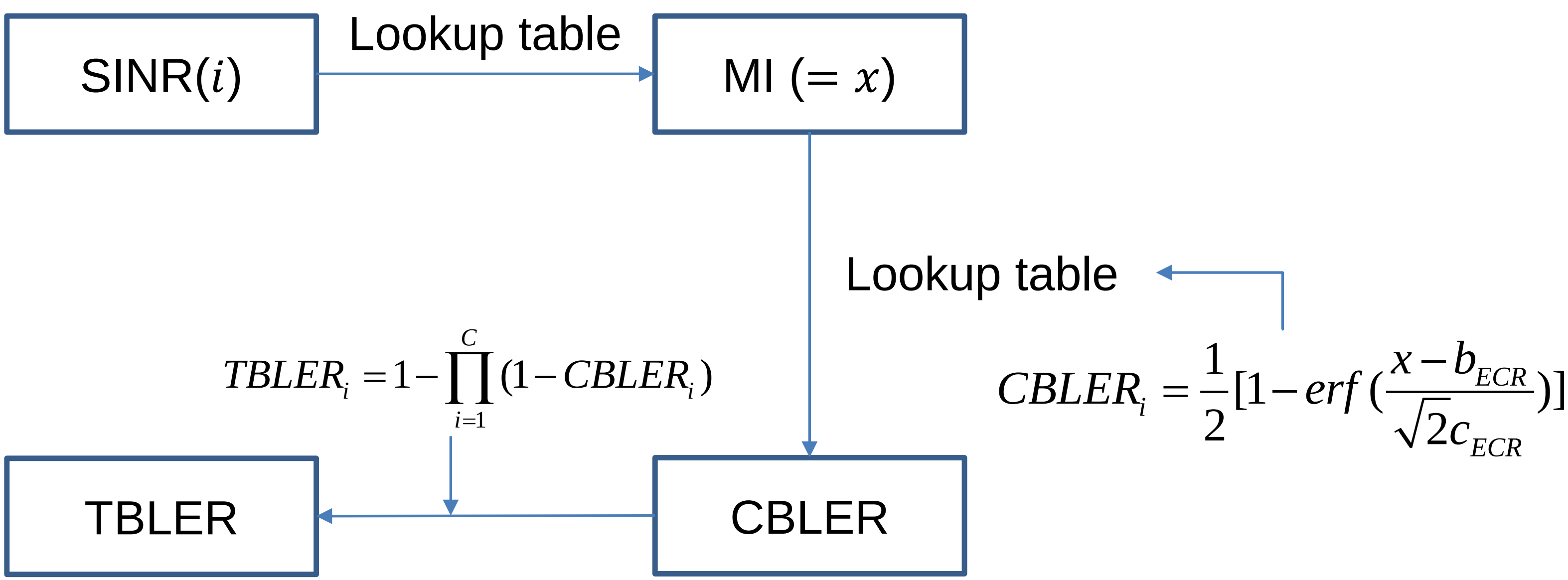
Channel Model

- LTE fast fading model in ns-3
  - Applying fast fading to each Resource Block (RB)
    - Extracting the fast fading value from the trace at each simulation time
    - Representing frequency response



PHY Model

- Error rate model
  - Link-to-system mapping (LSM) technique
    - Mutual information effective SINR mapping (MIESM)
  - Use SINR values, MCS, Coding block (CB) size, and HARQ info.



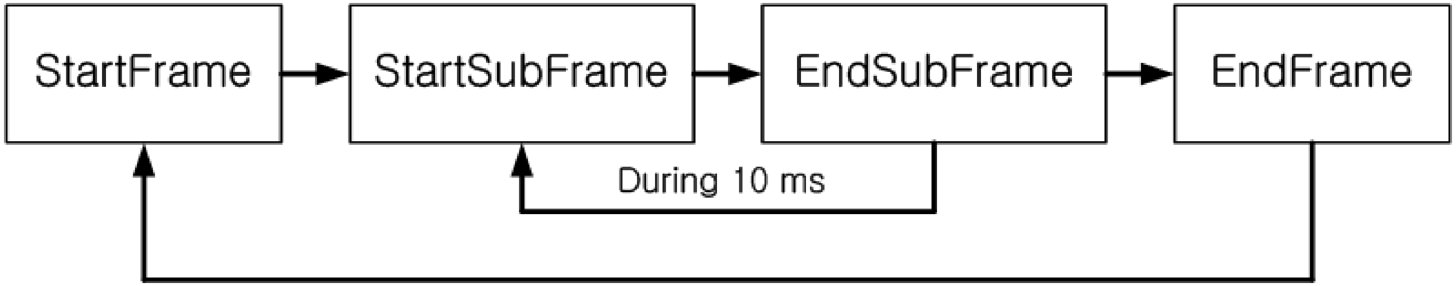
$$TBLER_i = 1 - \prod_{i=1}^{C} (1 - CBLER_i)$$

$$CBLER_i = \frac{1}{2}[1 - erf(\frac{x - b_{ECR}}{\sqrt{2} c_{ECR}})]$$

Function Flow

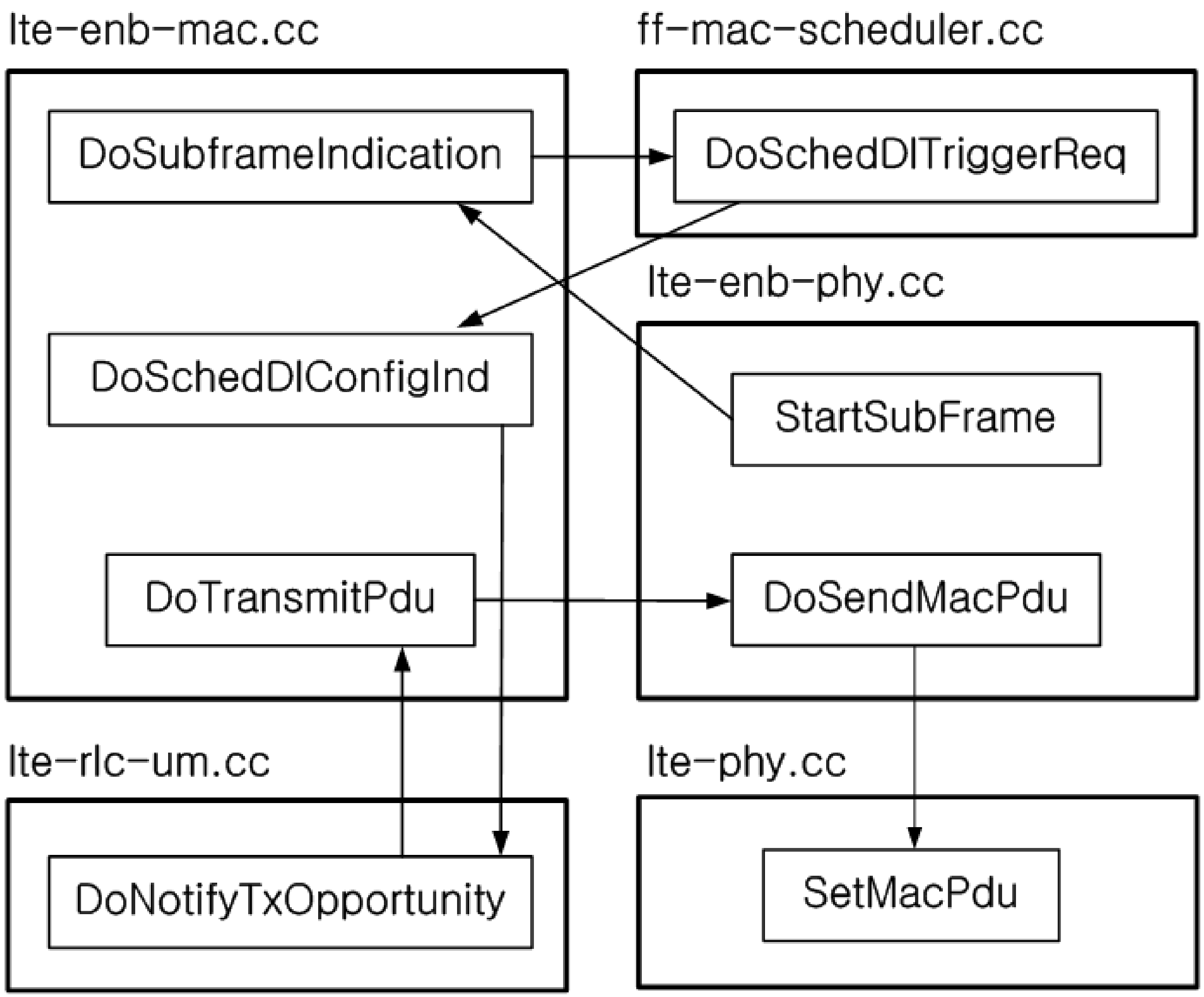
Frame Structure

- Source code (src/lte/model/lte-enb-phy.cc)
  - Function call flow
    - 1 Frame = 10 Subframes

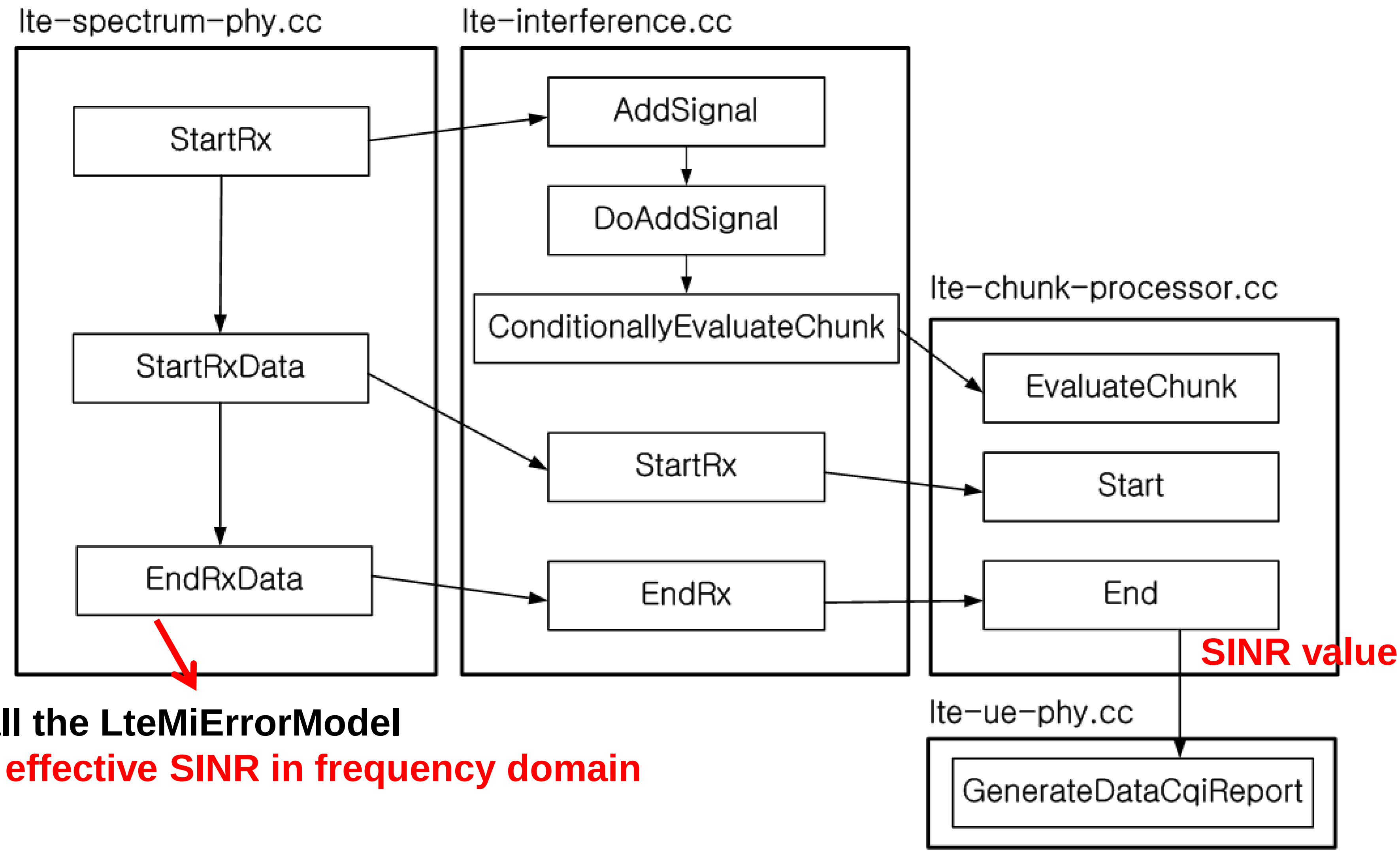


  - Function of StartSubFrame
    - Process the current burst of control messages
    - Send data frame
    - Trigger the MAC

Data Flow (Downlink)



Interference



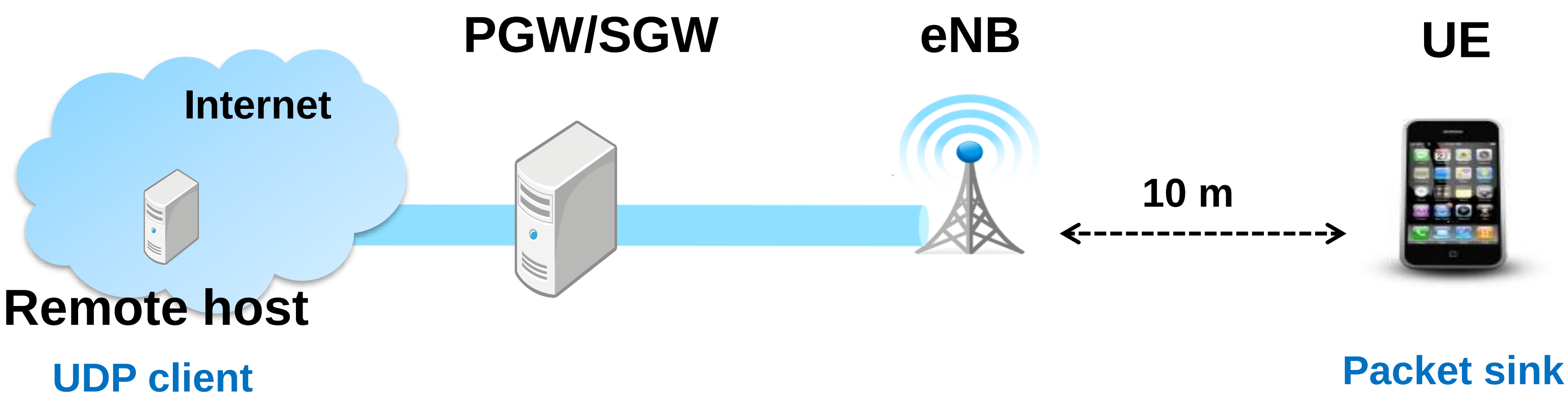
Call the LteMiErrorModel

MI effective SINR in frequency domain

Simulation Examples

Packet Transmission Example

- Simulation environment
  - Remote host generates UDP packet
    - Packet size: 1460 bytes
  - Downlink bandwidth: 20 MHz (100 RBs)



Simulation Code(1/9)

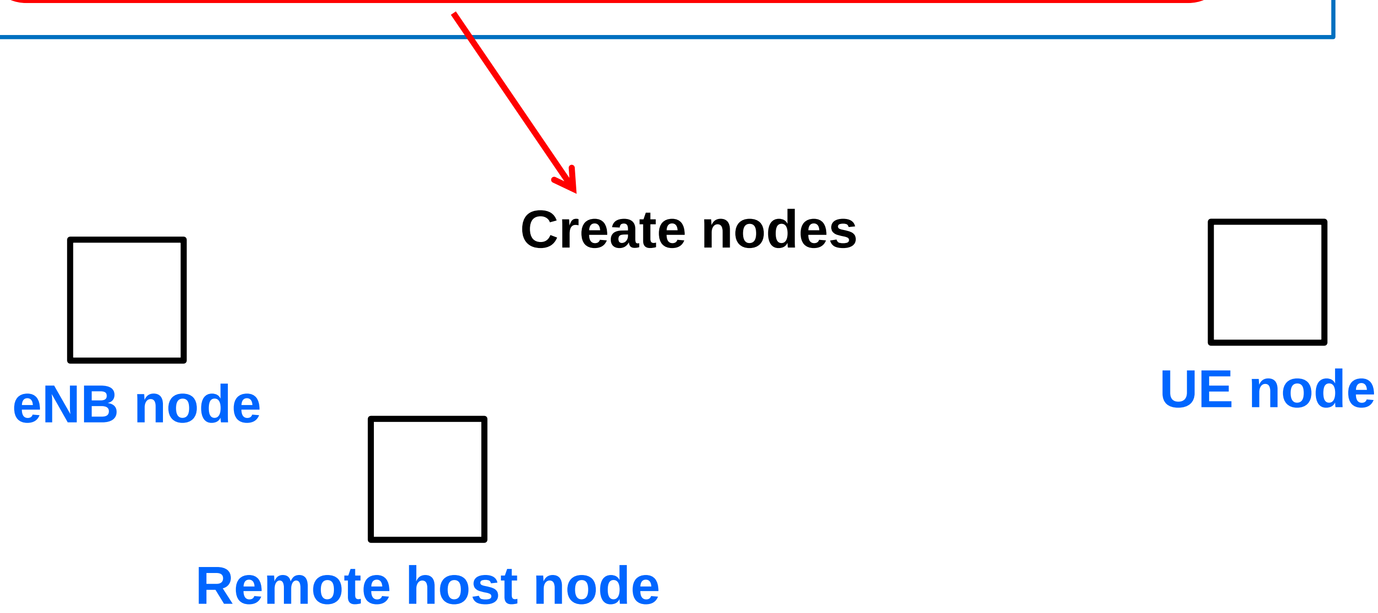
- lte-downlink.cc
  - Define variables

```
NS_LOG_COMPONENT_DEFINE ("LteDownlink");

using namespace ns3;

int main (int argc, char *argv[])
{
  double ltePktInv = 0.0001;   // full pumping LTE packet
  double lteStartTimeSec = 1; // application start time
  double lteStopTimeSec = 2;  // application end time
  double downlinkRb = 100;    // the number of resource blocks used for downlink
```

Simulation Code(2/9)

- Create node

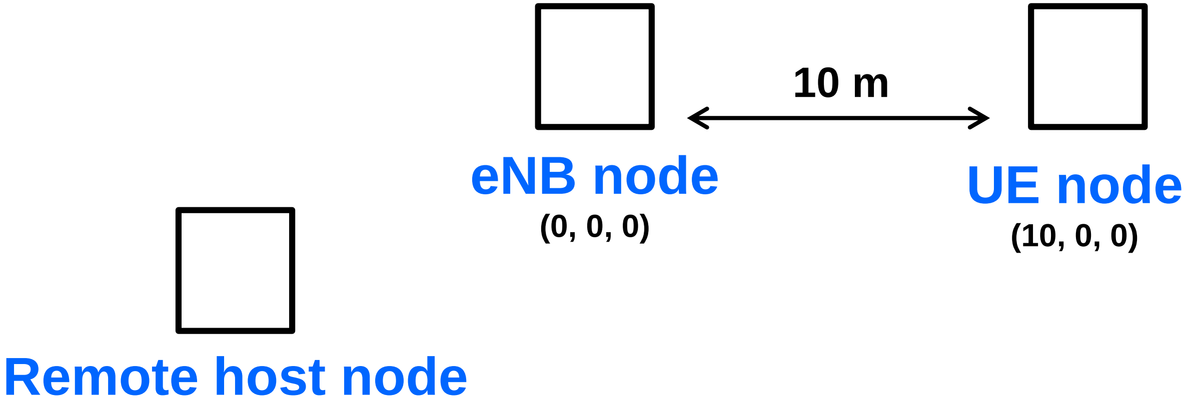
```
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

NodeContainer remoteHostNode, enbNode, ueNode;
remoteHostNode.Create (1);
enbNode.Create (1);
ueNode.Create (1);
```



Simulation Code(3/9)

- Allocate position

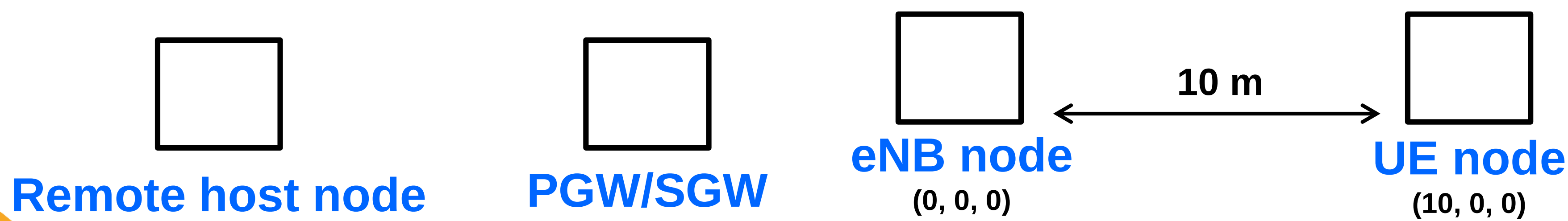
```
Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
positionAlloc->Add (Vector(0, 0, 0)); // for LTE eNB
positionAlloc->Add (Vector(10, 0, 0)); // for LTE UE

MobilityHelper mobility;
mobility.SetPositionAllocator (positionAlloc);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (enbNode);
mobility.Install (ueNode);
```



Simulation Code(4/9)

- LTE and EPC configuration
  - Using LteHelper and EpcHelper

```
//LteHelper
Ptr<LteHelper> lteHelper = CreateObject<LteHelper> ();
lteHelper->SetAttribute ("PathlossModel", StringValue
("ns3::LogDistancePropagationLossModel"));
lteHelper->SetEnbDeviceAttribute ("DlBandwidth", UintegerValue (downlinkRb));
lteHelper->SetEnbDeviceAttribute ("UlBandwidth", UintegerValue (downlinkRb));

//EpcHelper
Ptr<PointToPointEpcHelper> epcHelper = CreateObject<PointToPointEpcHelper> ();
lteHelper->SetEpcHelper (epcHelper);
Ptr<Node> pgw = epcHelper->GetPgwNode ();

//Wired configuration (Remote host and PGW/SGW)
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("1Gbps"));
pointToPoint.SetDeviceAttribute ("Mtu", UintegerValue (1500));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("0.00001ms"));
```



Simulation Code(5/9)

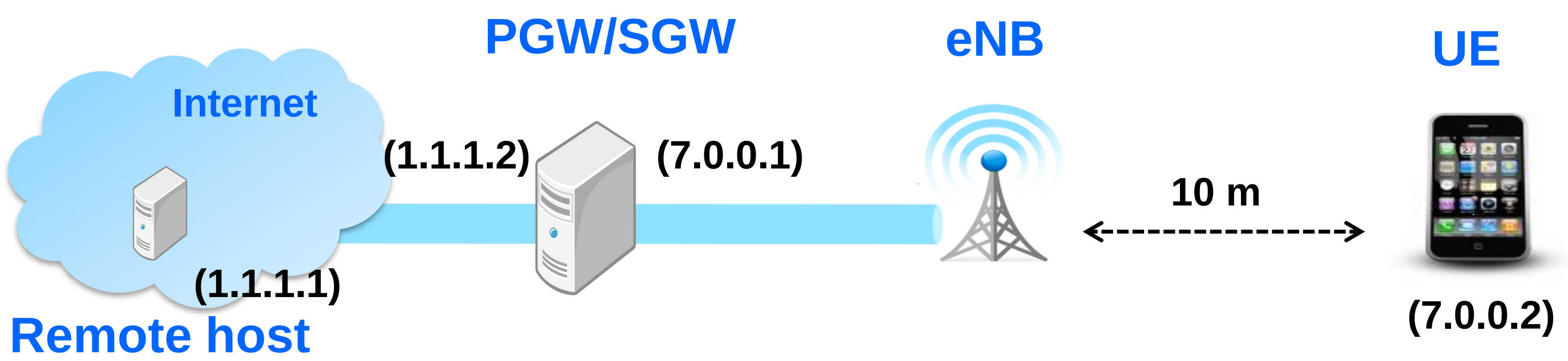
- Assign IP address to EPC

```
//Temporal containers
NetDeviceContainer p2pDevs, enbDev, ueDevs;

//Remote host– PGW/SGW
p2pDevs = pointToPoint.Install (remoteHostNode.Get(0), pgw);
enbDev = lteHelper->InstallEnbDevice (enbNode);
ueDevs = lteHelper->InstallUeDevice (ueNode);

// IP address assignment for remote host and PGW/SGW
InternetStackHelper internet;
internet.Install (remoteHostNode);

Ipv4AddressHelper ipv4;
ipv4.SetBase ("1.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer internetIpIfaces = ipv4.Assign (p2pDevs);

//Routing setting from EPC to LTE
Ipv4StaticRoutingHelper ipv4RoutingHelper;
Ptr<Ipv4StaticRouting> remoteHostStaticRouting = ipv4RoutingHelper.GetStaticRouting
((remoteHostNode.Get(0))->GetObject<Ipv4> ());
remoteHostStaticRouting->AddNetworkRouteTo (Ipv4Address ("7.0.0.0"), Ipv4Mask
("255.0.0.0"), 1);
```

Simulation Code(6/9)

- Assign IP address to LTE

```
//IP address assignment for UE
internet.Install (ueNode);

Ipv4InterfaceContainer ueIfs = epcHelper->AssignUeIpv4Address (NetDeviceContainer (ueDevs));
Ptr<Ipv4StaticRouting> ueStaticRouting = ipv4RoutingHelper.GetStaticRouting (ueNode.Get(0)->GetObject<Ipv4> ());
ueStaticRouting->SetDefaultRoute (epcHelper->GetUeDefaultGatewayAddress (), 1);

// Automatic attachment of a UE device to a suitable cell
lteHelper->Attach (ueDevs.Get (0));
```



Simulation Code(7/9)

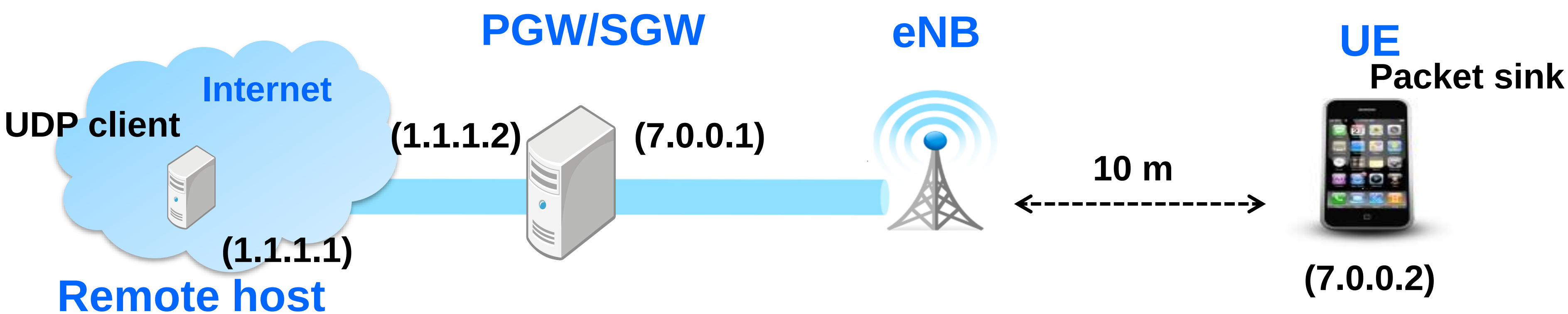
- Application

```
uint16_t dlPort = 10000;  //Setting the downlink UDP port

ApplicationContainer clientApps;   //Making an application container
ApplicationContainer serverApps;

//Set the destination IP address, port etc.
UdpClientHelper dlClientHelper (ueIfs.GetAddress (0), dlPort);

dlClientHelper.SetAttribute ("MaxPackets", UintegerValue (64707202));
dlClientHelper.SetAttribute ("Interval", TimeValue (Seconds (ltePktInv)));
dlClientHelper.SetAttribute ("PacketSize", UintegerValue (1460));

//Install the UDP client application to remote server
clientApps.Add (dlClientHelper.Install (remoteHostNode.Get(0)));

//Install the packet sink to UE
PacketSinkHelper dlPacketSinkHelper ("ns3::UdpSocketFactory", InetSocketAddress
(Ipv4Address::GetAny (), dlPort));
serverApps.Add (dlPacketSinkHelper.Install (ueNode.Get(0)));
```

Simulation Code(8/9)

- Activate dedicated bearer

```
//Setting the Traffic Flow Template (TFT)
Ptr<EpcTft> tft = Create<EpcTft> ();
EpcTft::PacketFilter dlpf;
dlpf.localPortStart = dlPort;
dlpf.localPortEnd = dlPort;
tft->Add (dlpf);

//Setting the bearer
EpsBearer bearer (EpsBearer::NGBR_VIDEO_TCP_DEFAULT);
lteHelper->ActivateDedicatedEpsBearer (ueDevs.Get (0), bearer, tft);
```



Simulation Code(9/9)

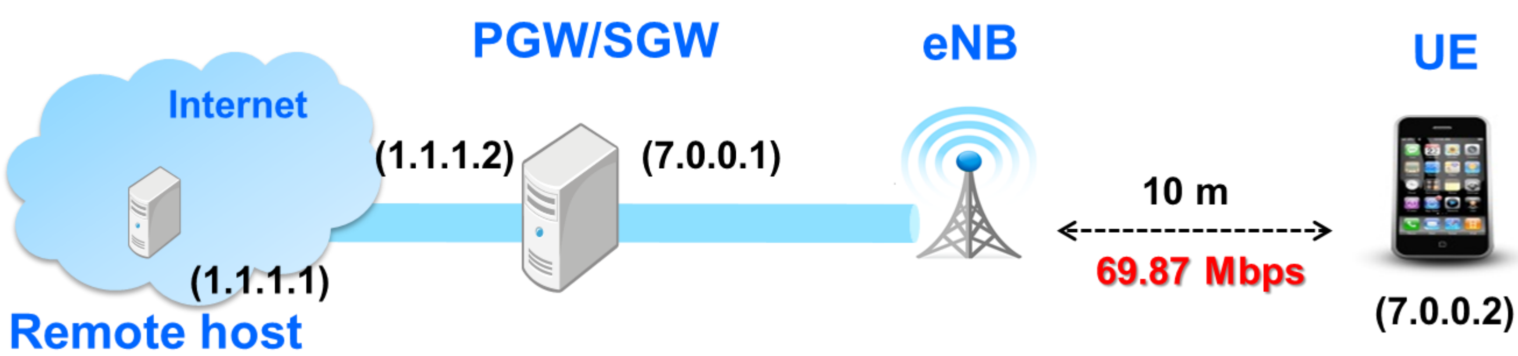
- Simulation running and Calculate the throughput

```
//Setting the simulation time
serverApps.Start (Seconds (lteStartTimeSec));
clientApps.Start (Seconds (lteStartTimeSec));
Simulator::Stop (Seconds(lteStopTimeSec));
Simulator::Run ();

//Tracing
Ptr<PacketSink> sink = serverApps.Get (0)->GetObject<PacketSink> ();
double lteThroughput = sink->GetTotalRx () * 8.0 / (1000000.0*(stopTimeSec -
lteStartTimeSec));

NS_LOG_UNCOND (“UE(" << ueIfs.GetAddress(0) <<“) lte throughput: "
<< lteThroughput);

Simulator::Destroy ();

return 0;
```

Packet Transmission Examples

- Simulation Results
  - ./waf --run scratch/lte-downlink

```
Build commands will be stored in build/compile_commands.json
'build' finished successfully (13.856s)
UE(7.0.0.2) lte throughput: 69.8698 Mbps
```



  - Is the above value correct?

Bandwidth: 20 MHz
Modulation: 64QAM
MIMO: without MIMO (SISO)

Maximum throughput: 75.376 Mbps

Configurable Parameters

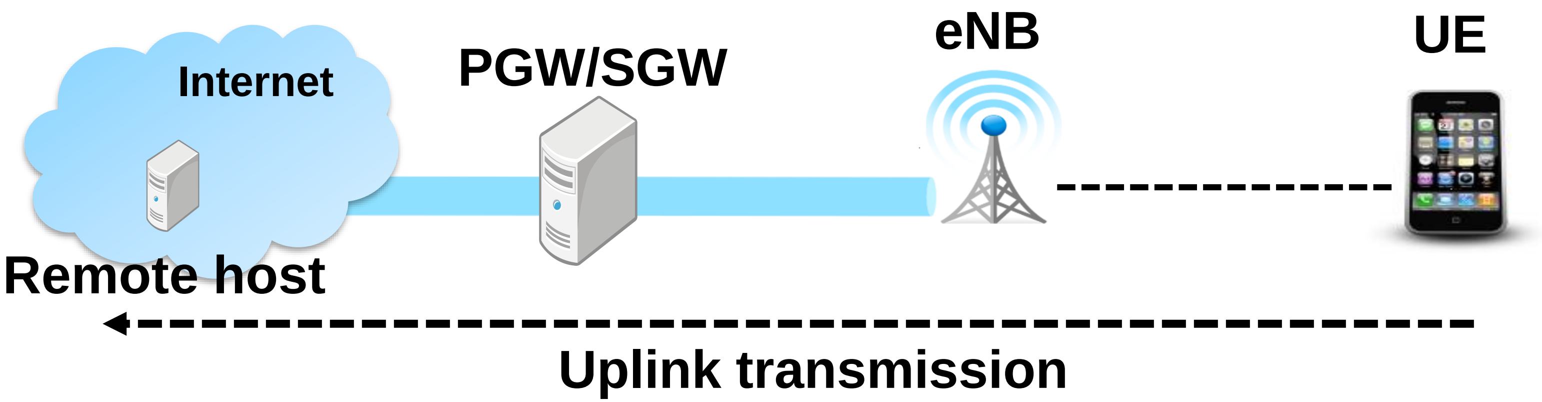
- Change parameters what you want !

```
double ltePktInv = 0.0001;
double downlinkRb = 100;
double distance = 10;

CommandLine cmd;

cmd.AddValue ("downlinkRb", "number of resource blocks in downlink",downlinkRb);
cmd.AddValue ("distance", "distance between eNB and UE [m]",distance);
cmd.AddValue ("ltePktInv", "packet generation interval of UDP [s]",ltePktInv);
cmd.AddValue ("dlEarfcn", "downlink frequency",dlEarfcn);
cmd.AddValue ("lteTxPower", "lte eNB transmission power [dBm]",lteTxPower);
cmd.Parse (argc, argv);
```

- Observe throughput value

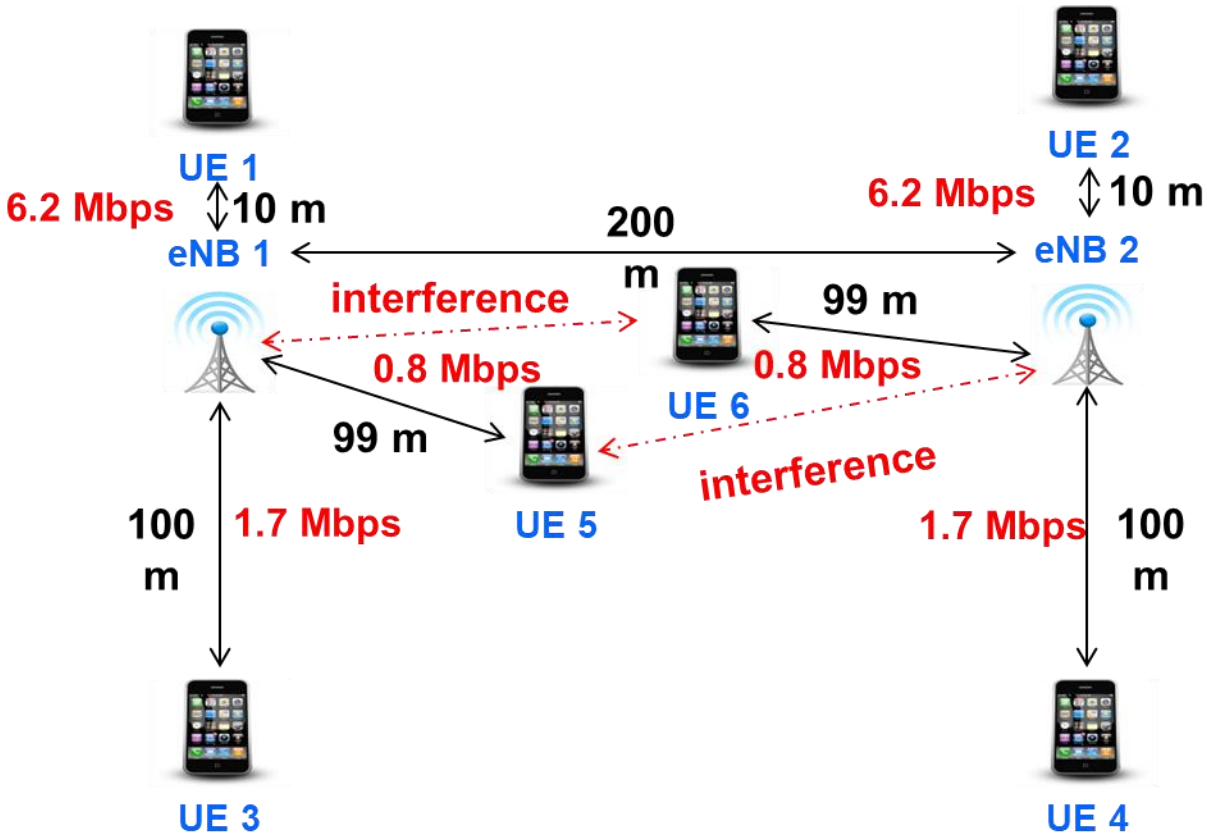
Uplink Transmission

- Uplink transmission
  - lte-uplink-skel.cc



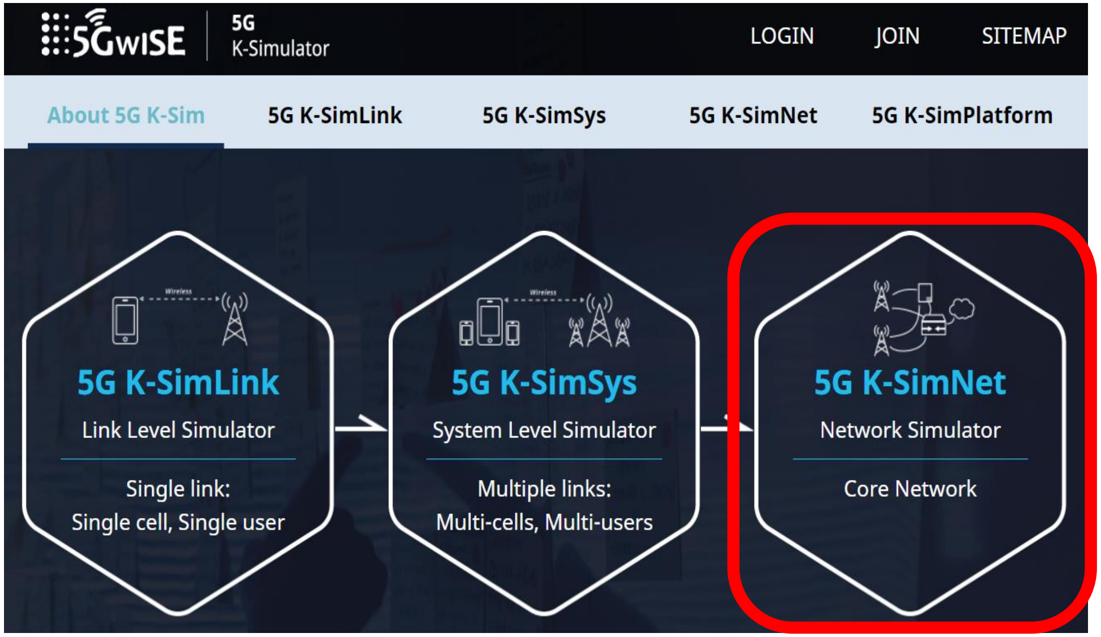
Multi-cell Scenario

- Multi-cell downlink transmission
  - lte-multicell-skel.cc



5G Simulator (5G K-SimNet)

- http://5gopenplatform.org/main/index.php



- Reference: ns-3 mmwave by NYU and Univ. of Padova
  - https://wireless.engineering.nyu.edu/e2e-network-simulation/

How to Use LTE Source?

- src/lte/model/~
- src/lte/helper/~
- Follow all procedures
  - lteHelper→InstallEnbDevice
  - lteHelper→InstallUeDevice
- Implement specific feature (e.g., handover, scheduling, …)
  - Make scratch file
  - Print log messages

Thank You !