

Tutorial on RINASIM simulations setup

In this tutorial, we give an idea about how to set up simulations using RINASim. The scenario described here has been used in the evaluation of distributed clouds use case in the scope of the PRISTINE project.

1. The scenario : Distributed clouds use case

The scenario called “distributed clouds use case” is composed of 120 nodes in total divided into 4 different regions. There are 30 nodes per each region. Ping application has been used for communication between the nodes. The packet size is set to 1500 Bytes. Table 1 summarizes the configuration parameters and Figure 1 illustrates the scenario.

Parameter	Value
Number of VIFIB nodes	120
Number of regions	4
Number of VIFIB nodes per region	30
Application	Ping
Packet Size	1500 Bytes
Ping Starts at	140s
Ping rate	5
Simulation Time	300s for each run

Table1. Simulation Configuration Parameters.

The DIF architecture that we build here is organized as follows:

- At the bottom, a Group DIF is constructed to regroup all the VIFIB nodes inside each region.
- Three inter-group DIFs are designed to interconnect region 1/2, region 2/3 and region 3/4.
- On top, a cloud tenant DIF that contains all the nodes that are communicating.

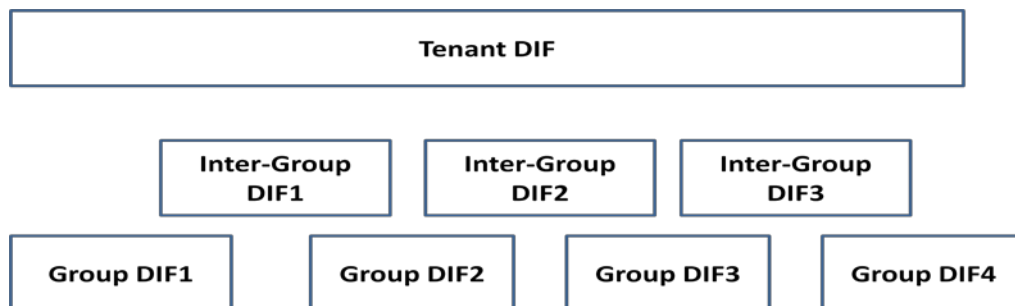


Figure 1. DIF architecture.

2. Setting up the network

a. Ned File

The ned file is where the network configuration must be done. Figure 2 illustrates the ned file of the distributed clouds scenario. In this file all the physical connections between the nodes should be defined. Please see the file DistributedCloudsLS.ned for more details.

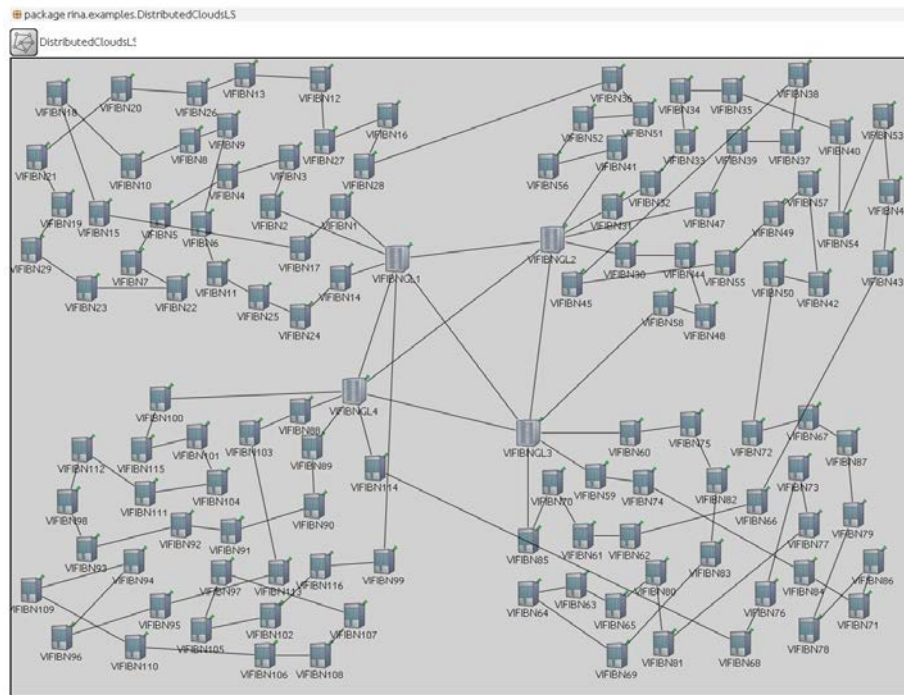


Figure 2. Ned file of the distributed clouds scenario.

b. Omnetpp.ini file

In this file all the simulation parameters are configured, first AE naming and RMT forwarding /forwarding generator/routing policies are configured. In the example, Simple table policy and Distance vector routing are used:

```

#
# Application entities naming:
#
**.VIFIBN1.AP.apName      = "App1"
**.VIFIBN2.AP.apName      = "App2"
**.VIFIBN5.AP.apName      = "App5"

# RMT Forwarding policies
**.VIFIBN*.TenantIPC.relayAndMux.ForwardingPolicyName = "SimpleTable"
**.VIFIBN*.GIPC.relayAndMux.ForwardingPolicyName = "SimpleTable"

**.VIFIBNGL*.IGIPC[*].relayAndMux.ForwardingPolicyName = "SimpleTable"

# forwarding generator policies
**.VIFIBN*.TenantIPC.resourceAllocator.pdufgPolicyName = "SimpleGenerator"
**.VIFIBN*.GIPC.resourceAllocator.pdufgPolicyName = "SimpleGenerator"

**.VIFIBNGL*.IGIPC[*].resourceAllocator.pdufgPolicyName = "SimpleGenerator"

# Routing policies
**.VIFIBN*.TenantIPC.routingPolicyName = "SimpleDV"
**.VIFIBN*.GIPC.routingPolicyName = "SimpleDV"

**.VIFIBNGL*.IGIPC[*].routingPolicyName = "SimpleDV"

```

Then the DIF addressing is configured, an example of address configuration of the group DIF 1 is illustrated here. We assign address 1 to the node 1 of the group; address 2 to the node 2 and so on:

```

#G1 DIF
**.VIFIBN1.GIPC.difName      = "G1DIF"
**.VIFIBN1.GIPC.ipcAddress    = "1"
**.VIFIBN2.GIPC.difName      = "G1DIF"
**.VIFIBN2.GIPC.ipcAddress    = "2"
**.VIFIBN3.GIPC.difName      = "G1DIF"

```

Also other groups DIF and tenant DIFs addressing should be defined. Node 1 is a part of the tenant DIF with address 1:

```

#Tenant DIF
**.VIFIBN1.TenantIPC.difName = "TenantDIF"
**.VIFIBN1.TenantIPC.ipcAddress = "1"

```

The physical connections between the nodes are represented also in Shim DIFs as described in the following. Node 1 in group 1 is connected physically to node 17. So shim1To17 is the shim Dif ensuring this connection. Node 1 has the address 1 and node 17 has the address 2:

```

#
# Server instances addressing:
#
# G1 Shims:
**.VIFIBN1.ShimIPC[0].difName      = "Shim1To17"
**.VIFIBN1.ShimIPC[0].ipcAddress    = "1"
**.VIFIBN17.ShimIPC[1].difName     = "Shim1To17"
**.VIFIBN17.ShimIPC[1].ipcAddress   = "2"

```

Then, we need to give the path to the configuration file (config.xml) that contains all the configuration parameters needed for the DIF allocator, qos cube setting and preallocated flows in the hosts, as described here:

```

#
# DIF Allocator settings
#
**.VIFIBN*.difAllocator.configData = xmldoc("config.xml", "Configuration/Node[@id='all']/DA")
#
# QoS Cube sets.
#
**.ra.qoscubesData = xmldoc("config.xml", "Configuration/QoS Cubes Set")
#
# Preallocated flow on hosts:
#
**.ra.preallocation = \
    xmldoc("config.xml", "Configuration/ConnectionSets/ConnectionSet[@id='all']/")

```

Then, finally set up the application parameters, the source node (VIFIBN1); the destination application name (App120), the application type (ping)..:

```

#
# AEs don't do anything. We're only evaluating the routing table now.
#
**.VIFIBN1.AP.applicationEntity.iae.dstApName = "App120"
**.VIFIBN1.AP.applicationEntity.iae.dstAeName = "Ping"
**.VIFIBN1.AP.applicationEntity.iae.startAt = 90s
**.VIFIBN1.AP.applicationEntity.iae.pingAt = 100s
**.VIFIBN1.AP.applicationEntity.iae.rate = 5
**.VIFIBN1.AP.applicationEntity.iae.stopAt = 1000s

```

3. Configure the DIF Allocator, Qos parameters, and pre-allocated flows:
 - a. Config.xml file

Pre-allocated flows are configured from the beginning of the file. Management and data flows in a separate way. Here, a communication between node 1 and node 120 is set up:

```

<!-- Allocate management flows. -->
<SimTime t="0">
    <Connection src="120_TenantDIF" dst="1_TenantDIF" qosReq="mgmt"/>
</SimTime>
<!-- Allocate data flows. -->

<SimTime t="10">
    <Connection src="120_TenantDIF" dst="1_TenantDIF" qosReq="1"/>

```

Then, DIF allocator configuration is done: for each DIF, the next DIF to use is specified. Here, for instance, from tenantDIF(address 1) use G1DIF (address 1)

```

<Node id="all">
    <DA>
        <APN apn="1_TenantDIF">
            <DIF difName="G1DIF" ipcAddress="1"/>
        </APN>

```

Then, from G1DIF, there are two DIFs that could be used: either Shim1To17, or Shim1ToGL1:

```

<APN apn="1 G1DIF">
    <DIF difName="Shim1To17" ipcAddress="1"/>
    <DIF difName="Shim1ToGL1" ipcAddress="1"/>
</APN>

```

Qos cubes and qos requirements configuration is illustrated in the following:

```
<QoSConfig>
  <QoSConfigSet>
    <QoSConfig id="1">
      <AverageBandwidth>12000000</AverageBandwidth>
      <AverageSDUBandwidth>1000</AverageSDUBandwidth>
      <PeakBandwidthDuration>24000000</PeakBandwidthDuration>
      <PeakSDUBandwidthDuration>2000</PeakSDUBandwidthDuration>
      <BurstPeriod>10000000</BurstPeriod>
      <BurstDuration>1000000</BurstDuration>
      <UndetectedBitError>0.01</UndetectedBitError>
      <PDUDroppingProbability>0</PDUDroppingProbability>
      <MaxSDUSize>1500</MaxSDUSize>
      <PartialDelivery>0</PartialDelivery>
      <IncompleteDelivery>0</IncompleteDelivery>
      <ForceOrder>0</ForceOrder>
      <MaxAllowableGap>0</MaxAllowableGap>
      <Delay>1000000</Delay>
      <Jitter>500000</Jitter>
      <CostTime>0</CostTime>
      <CostBits>0</CostBits>
      <ATime>0</ATime>
    </QoSConfig>
  </QoSConfigSet>
  <QoSReqSet>
    <QoSReq id="1">
      <AverageBandwidth>12000000</AverageBandwidth>
      <AverageSDUBandwidth>1000</AverageSDUBandwidth>
      <PeakBandwidthDuration>24000000</PeakBandwidthDuration>
      <PeakSDUBandwidthDuration>2000</PeakSDUBandwidthDuration>
      <BurstPeriod>10000000</BurstPeriod>
      <BurstDuration>1000000</BurstDuration>
      <UndetectedBitError>0.01</UndetectedBitError>
      <PDUDroppingProbability>0</PDUDroppingProbability>
      <MaxSDUSize>1500</MaxSDUSize>
      <PartialDelivery>0</PartialDelivery>
      <IncompleteDelivery>0</IncompleteDelivery>
      <ForceOrder>0</ForceOrder>
      <MaxAllowableGap>0</MaxAllowableGap>
      <Delay>1000000</Delay>
      <Jitter>500000</Jitter>
      <CostTime>0</CostTime>
      <CostBits>0</CostBits>
    </QoSReq>
  </QoSReqSet>
</QoSConfig>
```

Now, once we have configured our scenario, we can run the simulation.