

RINASim

## **BASICS**

**Vladimír Veselý**

Brno University of Technology, Czech Republic

[veselyv@fit.vutbr.cz](mailto:veselyv@fit.vutbr.cz)

# Agenda

- 1) Introduction
- 2) OMNeT++ Handbook
- 3) RINASim overview
- 4) Interactive Demo
- 5) Conclusion



# 1) Introduction

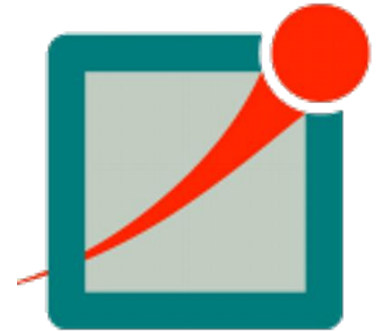
*How to install?*

*Where to get it?*

*Documentation and useful links.*

# Requirements

- ◆ OMNeT++ discrete event simulator
  - ◆ Windows, Linux, FreeBSD environment
  - ◆ Free for non-commercial purposes
  - ◆ Supported on version 5.0
    - ◆ The last release compatible with OMNeT++ 4.6 is <https://github.com/kvetak/RINA/releases/tag/August2016-v4>
  - ◆ No other libraries or frameworks needed
    - ◆ Potential cooperation with INET framework



# VM Install

## ◆ Out-of-the box virtual machine

- ◆ OMNeT++ 5.0 with the newest RINASim

- ◆ Generic OVA appliance

  - ◆ MintLinux

- ◆ Created on VMWare Workstation

  - ◆ *...should work also on VirtualBox and Qemu*

## ◆ Download from

<http://nes.fit.vutbr.cz/ivesely/vm/RINASim.zip>

# Windows Installation

◆ Uses MINGW32\_NT-10.0-WOW

◆ Cookbook

1) Download OMNeT++

<https://omnetpp.org/omnetpp/send/30-omnet-releases/2307-omnetpp-50-windows>

2) ./configure && make

<http://omnetpp.org/doc/omnetpp/InstallGuide.pdf>

3) Download RINASim

<https://github.com/kvetak/RINA/archive/master.zip>

4) Import RINASim project

OMNeT++

Category: OMNeT++ Releases

Older versions

OMNeT++ 5.0 (Windows)

Release 5.0 is a result of development of new features compared to the last 4 x

Create new file

Upload files

Find file

Clone or download

Clone with HTTPS ?

Use SSH

Use Git or checkout with SVN using the web URL.

<https://github.com/kvetak/RINA.git>



Open in Desktop

Download ZIP

# Installation Linux

◆ gcc 4.9.2 with C++11 support

◆ Cookbook

## 1) Requirements

build-essential gcc g++ bison flex perl tcl-dev tk-dev libxml2-dev zlib1g-dev default-jre  
doxygen graphviz libwebkitgtk-1.0-0

## 2) Download OMNeT++

<https://omnetpp.org/omnetpp/send/30-omnet-releases/2305-omnetpp-50-linux>

## 3) ./configure && make

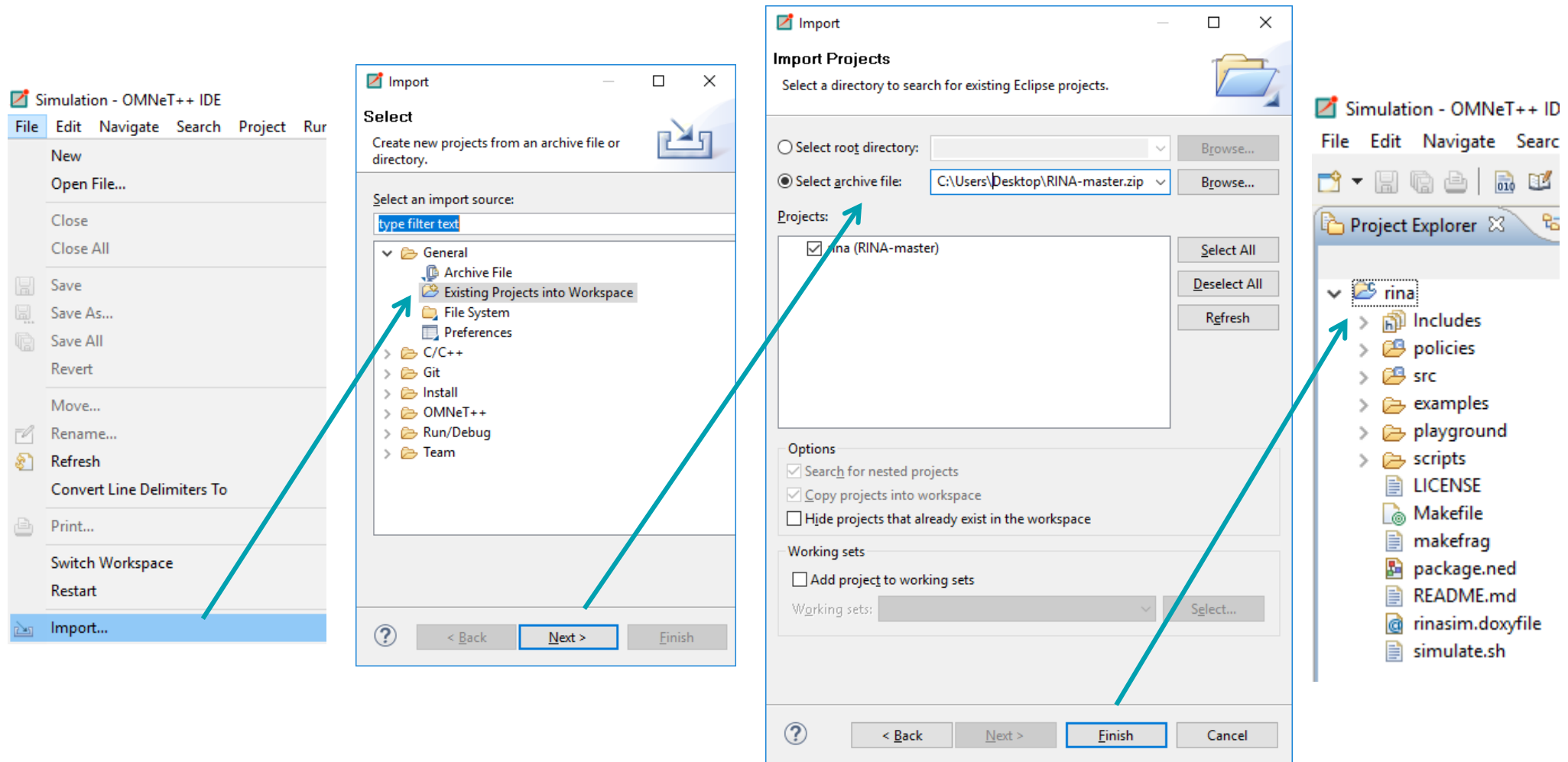
<http://omnetpp.org/doc/omnetpp/InstallGuide.pdf>

## 4) Download RINASim

<https://github.com/kvetak/RINA/archive/master.zip>

## 5) Import RINASim project

# Importing RINASim Project





# Navigation

- ◆ **/src** ... RINASim core source codes
  - ◆ **/Common** ... common or shared ADTs
  - ◆ **/CS** ... high-level nodes
  - ◆ **/DAF** ... DAF components
  - ◆ **/DIF** ... DIF components
- ◆ **/policies** ... programmable set of policies
- ◆ **/examples** ... accompanied scenarios
- ◆ **/playground** ... unmaintained/experimental scenarios
- ◆ **/out** ... compiled project binaries
- ◆ **/doc** ... Doxygen source code documentation
- ◆ **/scripts** ... devel batches for SLOC and fingerprinting

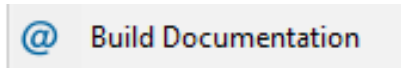
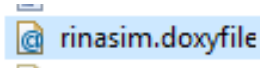
# Documentation

## ◆ Doxygen

◆ <http://nes.fit.vutbr.cz/ivesely/doxy>

◆ Important ADTs, ongoing work

◆ Build it in folder **/doc**



## ◆ PRISTINE D2.4 and D2.6

◆ <http://ict-pristine.eu/?p=472>

◆ <http://ict-pristine.eu/?p=772>



## ◆ Nasty details about RINASim are in my dissertation

- ◆ VESELÝ, Vladimír. *A NEW DAWN OF NAMING, ADDRESSING AND ROUTING ON THE INTERNET*. Brno, 2016. Available from: <http://www.fit.vutbr.cz/study/DP/PD.php?id=515>. PhD. Thesis. Brno University of Technology, Faculty of Information Technology. 2016-04-12. Supervisor Švéda Miroslav.

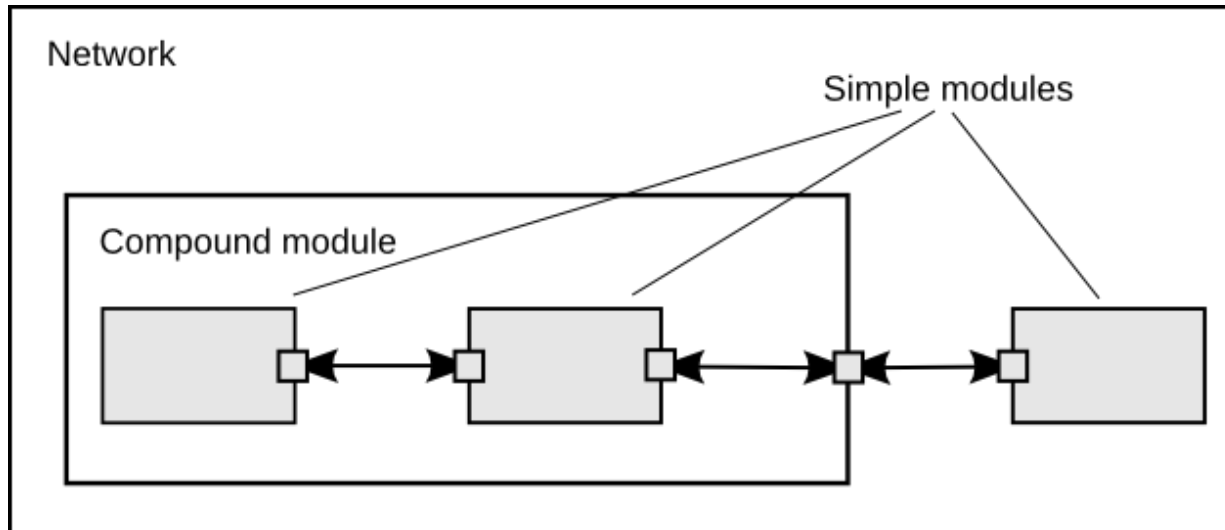


## 2) OMNeT++ Handbook

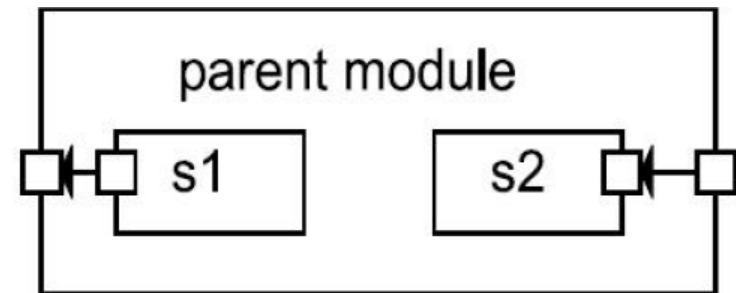
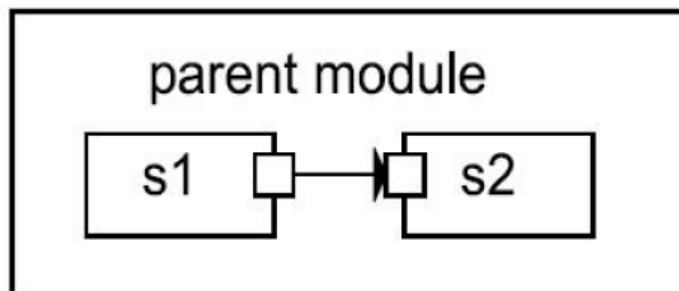
*The fast and the furious introductory course for OMNeT++ discrete event simulator*

# Simulation Module

## Modules



## Gates (input, output, inout), messages



# Languages

## ◆ NED

- ◆ to define models and interconnections

- ◆ \*.ned

## ◆ C++

- ◆ to implement model behavior

- ◆ \*.h and \*.cc

## ◆ Message definition

- ◆ to deploy easily C++ message classes

- ◆ \*.msg

# Module's Hierarchy

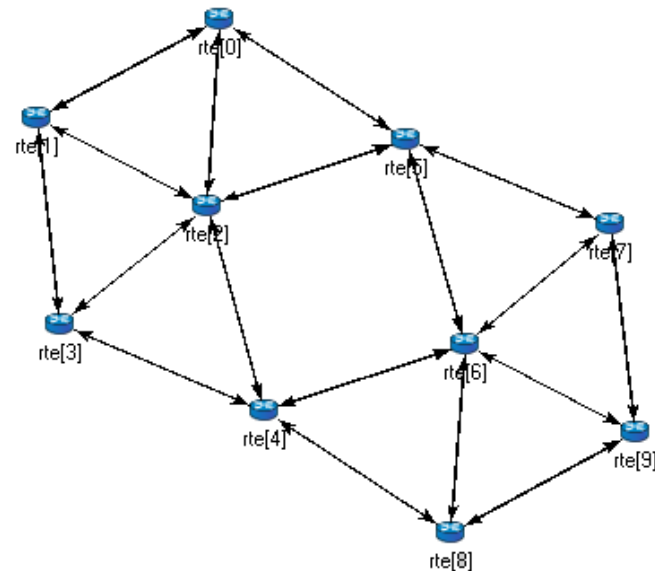
## Simple

```
simple TestModul
{
  parameters:
    @display("i=block/queue");
  gates:
    input in;
    output out;
}
```

## Compound

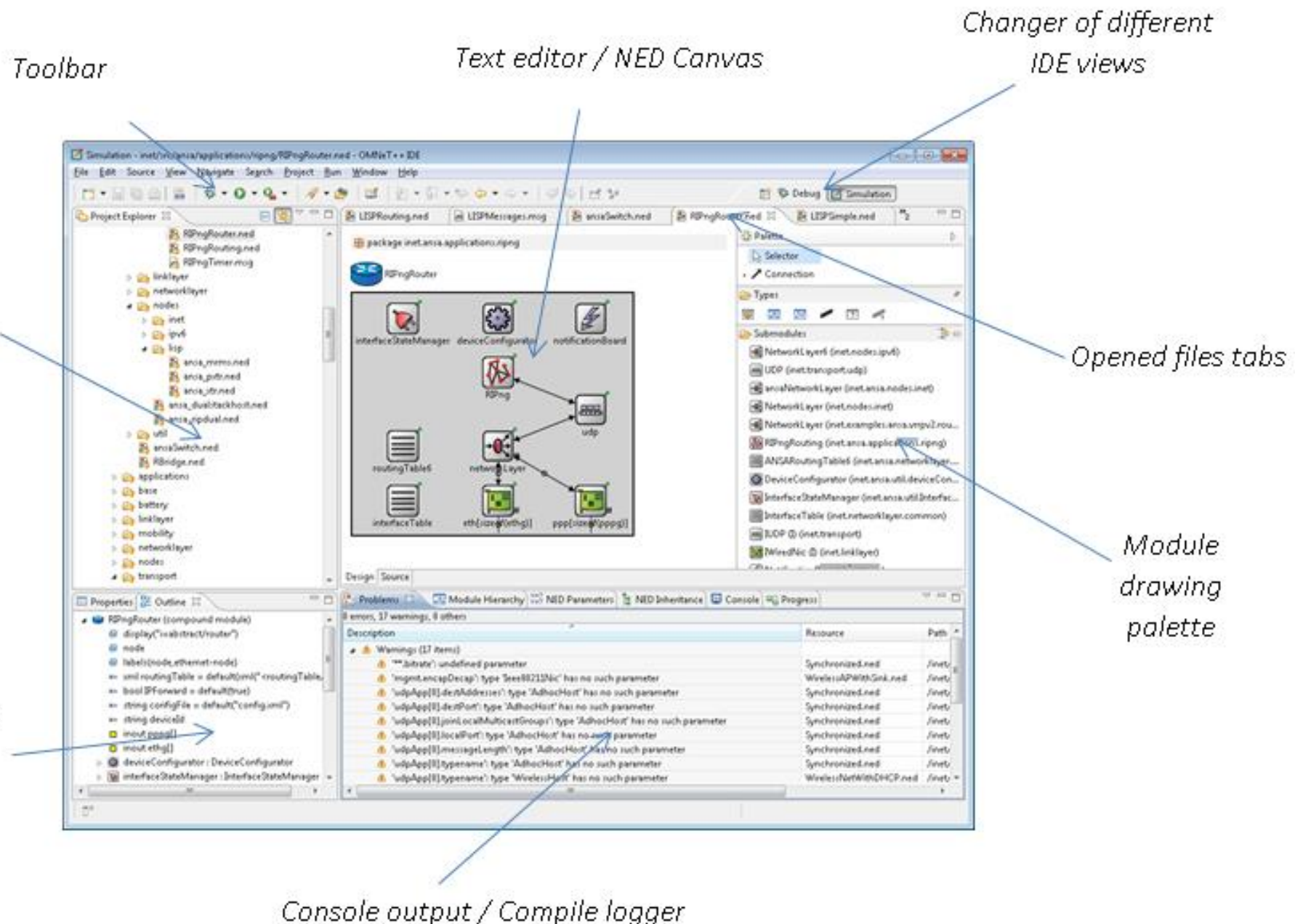
```
module Router
{
  parameter:
    @display("i=block/router");
  gates:
    inout SerialInterface[];
    inout EthInterface[];
  submodules:
    tcp: TCP;
    ip: IP;
    layer1: physicalLayer;
  connections:
    tcp.ipIn <-- ip.tcpOut;
    tcp.ipOut --> ip.tcpIn;
    layer1.ipIn <-- ip.llIn;
    layer1.ipOut --> ip.llOut;
}
```

## Network



```
//
// A network
//
network Network
{
  submodules:
    node1: Node;
    node2: Node;
    node3: Node;
    ...
  connections:
    node1.port++ <--> {datarate=100Mbps;} <--> node2.port++;
    node2.port++ <--> {datarate=100Mbps;} <--> node4.port++;
    node4.port++ <--> {datarate=100Mbps;} <--> node6.port++;
    ...
}
```

# IDE



# Simulation

Control buttons

Detached graphical component detail

Scheduled messages

The screenshot displays the OMNeT++ simulation environment. At the top, a menu bar and toolbar are visible. Below the toolbar, a timeline shows the simulation progress. The main window is divided into several panes:

- Scheduled messages:** A list of messages scheduled for execution, including PING-0 through PING-4, StopCommunication, and various RcvrInactivityTimer events.
- Component parameters:** A table showing the parameters for the selected component, (IPProcess) Demo.hostA.ipcProcess0.
- Console log:** A log of events and messages, including the creation of resources and the receipt of a control PDU.
- Network diagram:** A visual representation of the network topology, showing hosts, switches, and the internal structure of a host.

Class	Name	Info
cPar	ipcAddress	"1"
cPar	diffName	"Layer01"
cPar	apName	"1_Layer01"
cPar	relay	false
cPar	routingPolicyName	"DummyRouting"
cPar	onWire	false
cGate	southlo\$[0]	<-- <parent>.medium\$0
cGate	southlo\$[0]	<-- <parent>.medium\$0
cGate	northlo\$[0]	<-- ipcProcess1.southlo\$[0]
cGate	northlo\$[0]	<-- ipcProcess1.southlo\$[0]
cGate	northlo_14935\$0	<-- ipcProcess1.southlo_ipcProcess0_1
cGate	northlo_35665\$0	<-- ipcProcess1.southlo_ipcProcess0_3
cGate	northlo_35665\$0	<-- ipcProcess1.southlo_ipcProcess0_3
FlowAllocator	flowAllocator	id=27
RelayAndMux	relayAndMux	id=28

Component parameters

Console log



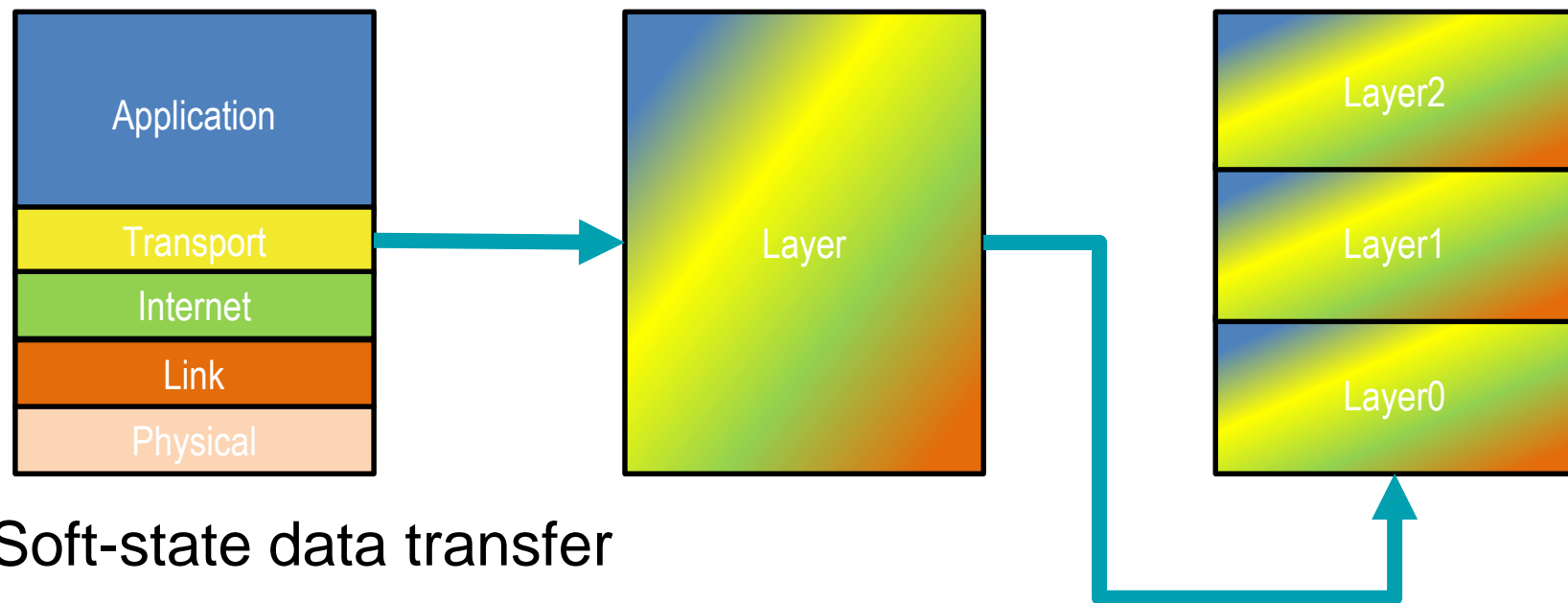


### 3) RINASim Overview

*How are things stitched together?  
Components and their purpose*

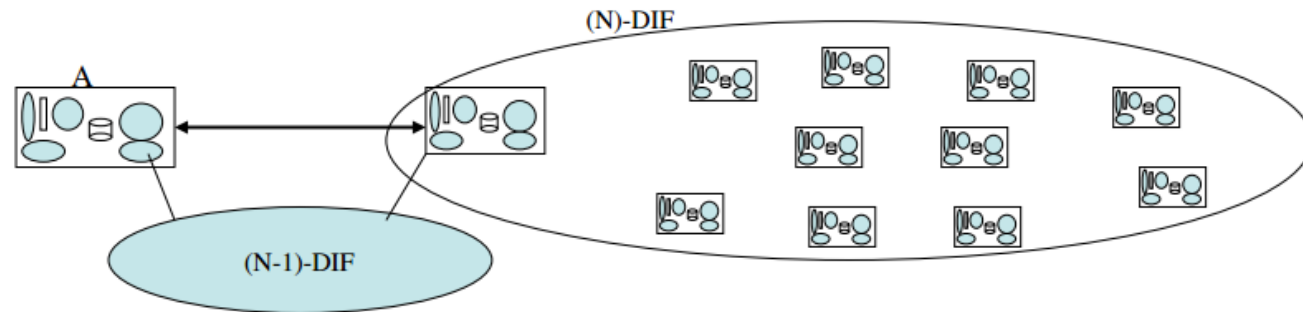
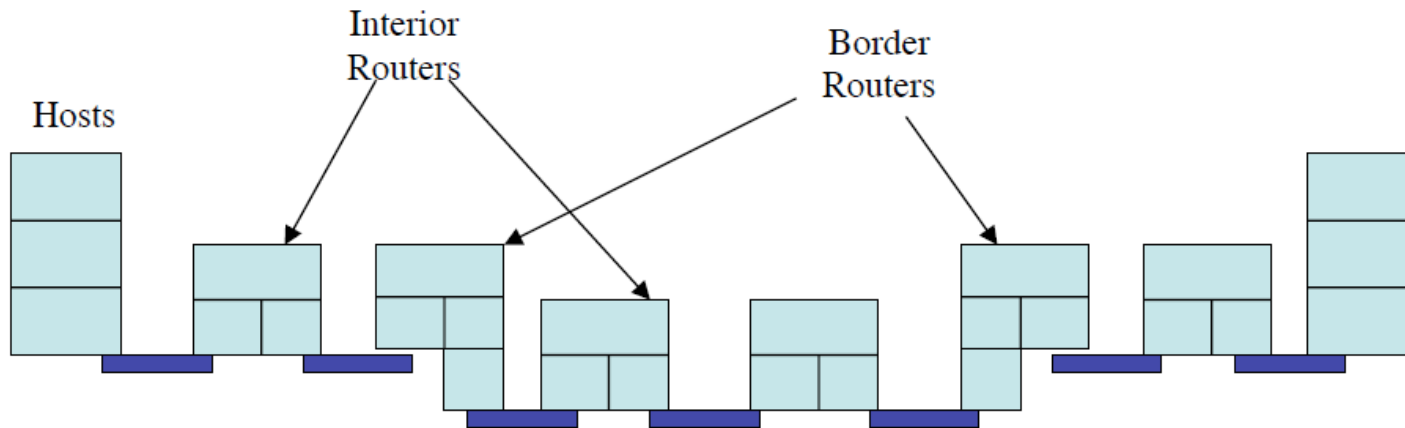
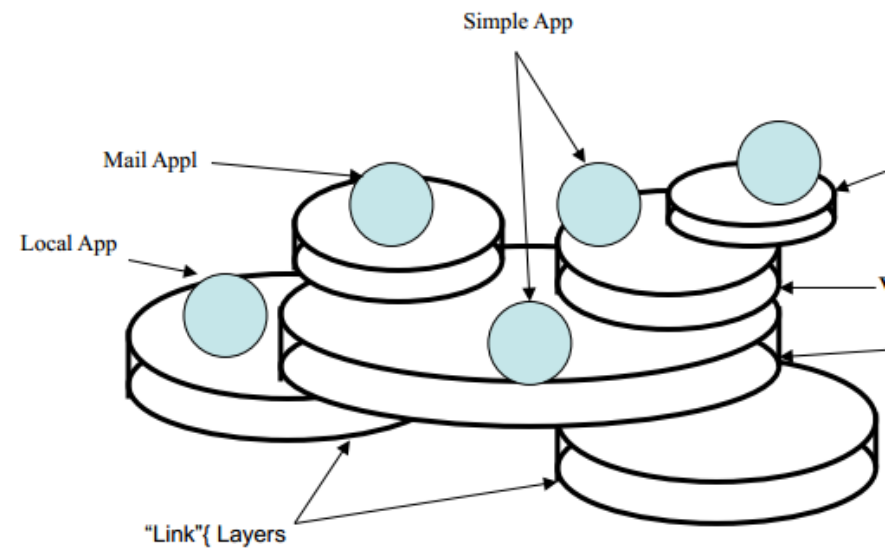
# TCP/IP vs RINA

- One generic layer for interprocess communication called **DIF** that limits scope



- 1) Soft-state data transfer
- 2) Complete addressing and naming
- 3) Single generic application protocol
- 4) Split between fixed mechanism and programmable policy

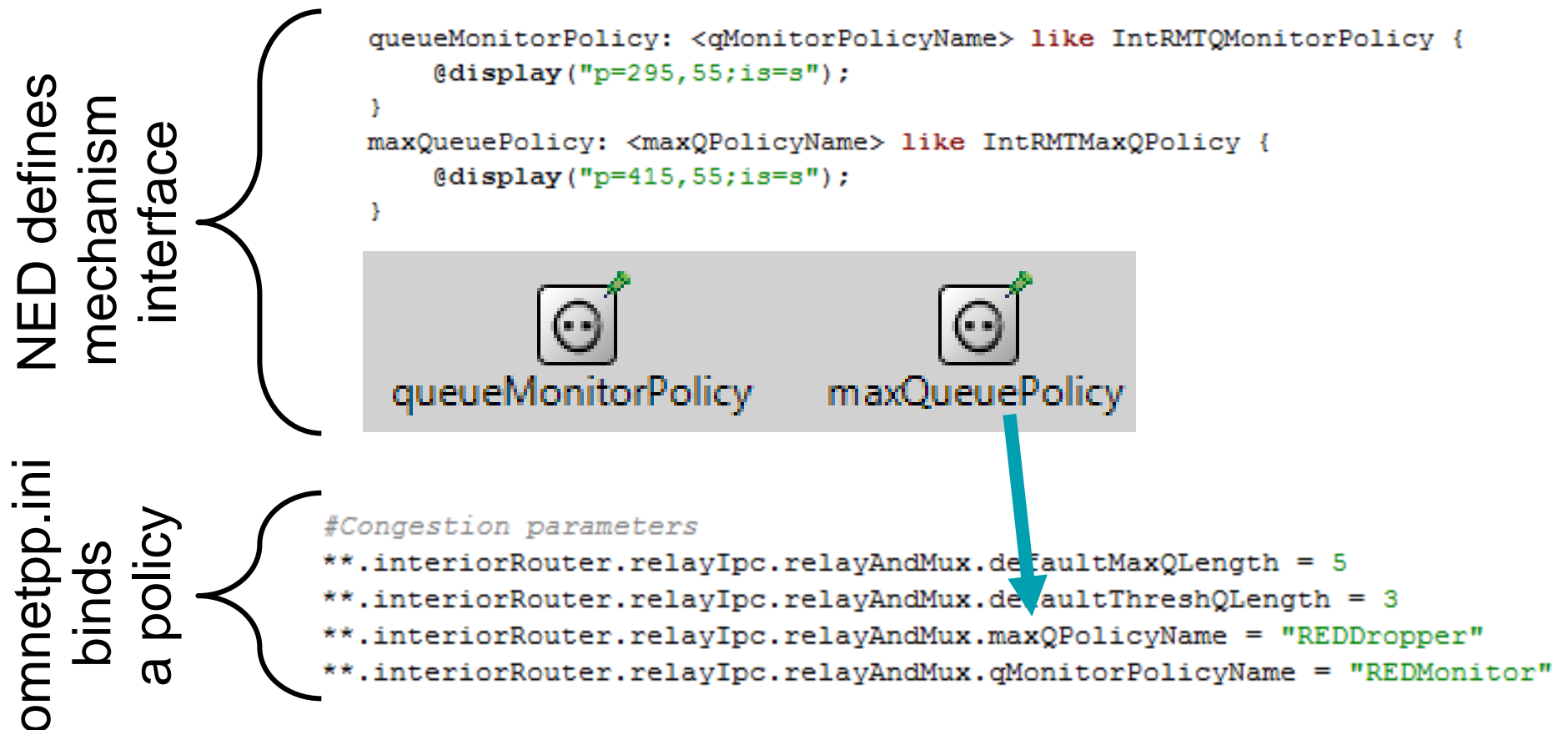
# RINA Network



Pictures are taken from John Day's presentation at <http://www.slideshare.net/ictpristine/rina-introduction-part-ii>

# Design

- Split between mechanism and policy
  - Modules as interfaces



# Configuring Parameters ①

- ◆ *How to setup your simulation?*
- ◆ Statically preconfigured
  - ◆ NED parameters in `omnetpp.ini`
  - ◆ More complex configuration (lists) in `config.xml`
- ◆ On-the-fly
  - ◆ not supported everywhere

# Configuring Parameters ②

omnetpp.ini

```
#Static addressing
**.hostA.ipcProcess0.ipcAddress = "1"
**.hostA.ipcProcess0.difName = "Layer0"
**.hostB.ipcProcess0.ipcAddress = "2"
**.hostB.ipcProcess0.difName = "Layer0"
#QoS settings
**.ra.qoscubesData = xmldoc("config.xml", "Configuration/QoS CubesSet")
```

config.xml

```
<Configuration>
  <Host id="host">
    <DA>
      <Directory>
        <APN apn="SourceA">
          <DIF difName="Layer0" ipcAddress="1" />
        </APN>
        <APN apn="DestinationB">
          <DIF difName="Layer0" ipcAddress="2" />
        </APN>
      </Directory>
    </DA>
    <QoS CubesSet>
      <QoSCube id="QoSCube_WithoutDTCP">
        <AverageBandwidth>10000000</AverageBandwidth>
        <AverageSDUBandwidth>1000</AverageSDUBandwidth>
        <PeakBandwidthDuration>20000000</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>
      </QoSCube>
    </QoS CubesSet>
  </Host>
</Configuration>
```

(list<QoSCube>) ...2.hostA.ipcProcess0.resourceAllocator.ra.this->QoS Cubes

Fields Contents (0)

this->QoS Cubes (list<QoSCube>)

elements[3] (QoSCube)

[0] = QoSCube Id> QoSCube\_WithoutDTCP

- average BW = 10000000 bit/s, average SDU BW = 1000 SDU/s
- peak BW duration = 20000000 bit/s, peak SDU BW duration = 2000 SDU/s
- burst period = 10000000 usecs, burst duration = 1000000 usecs
- undetected bit errors = 0.01%, PDU dropping probability = 0%
- max SDU Size = 1500 B
- partial delivery = no, incomplete delivery = no
- force order = no
- max allowed gap = 0 SDUs
- delay = 1000000 usecs, jitter = 500000 usecs
- cost-time = 0 \$/ms, cost-bits = 0 \$/Mb
- A-Time = 0ms
- resiliency factor = do-not-care

[1] = QoSCube Id> QoSCube\_WithDTCP \ average BW = 10000000 bit/s, average SDU BW = 1000 SDU/s

[2] = QoSCube Id> MGMT-QoS Cube \ average BW = 12000 bit/s, average SDU BW = 12 SDU/s

running simulation

# Policy Implementations

## ▼ policies

### DAF

### ▼ DIF

### ▼ EFCP

### ▼ DTCP

- > ECN
- > ECNSlowDown
- > LostControlPDU
- > NoOverridePeak
- > NoRateSlowDown
- > RateReduction
- > RcvFCOverrun
- > RcvrAck
- > RcvrControlAck
- > RcvrFC
- > ReceivingFC
- > ReconcileFC
- > RxTimerExpiry
- > SenderAck
- > SendingAck
- > SndFCOverrun
- > TxControl

### ▼ DTP

- > InitialSeqNum
- > RcvrInactivity
- > RTTEstimator
- > SenderInactivity

### > EFCPPolicy.cc

### > EFCPPolicy.h

### ▼ FA

- > AllocateRetry
- > MultilevelQoS
- > NewFlowRequest

### ▼ RA

- > AddressComparator
- > PDUFG
- > QueueAlloc
- > QueueIDGen

### ▼ RMT

- > MaxQueue
- > Monitor
- > PDUForwarding
- > Scheduler

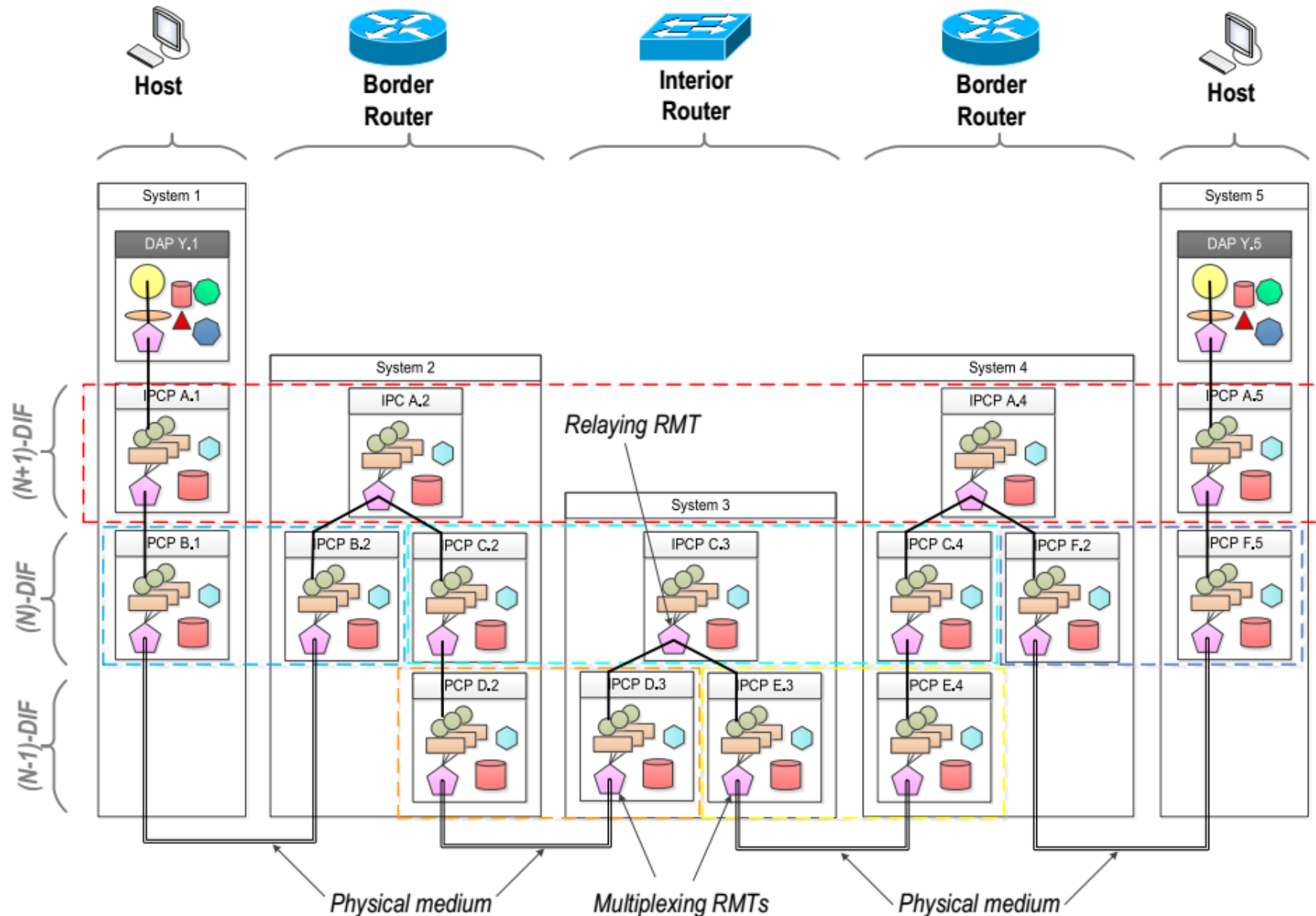
### ▼ Routing

- > common
- > DCRouting
- > DomainRouting
- > DummyRouting
- > PortsLoadRouting
- > SimpleRouting
- > TDomainRouting
- > TSimpleRouting

### ▼ SDUProtection

- > DummyProtection
- > FixedDelay

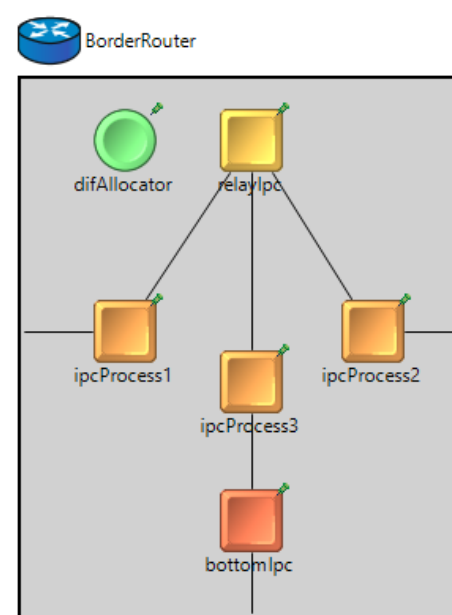
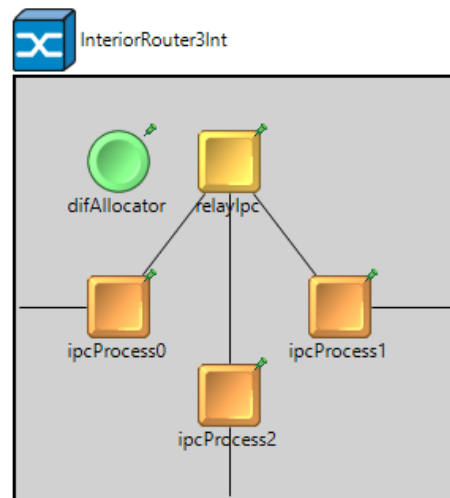
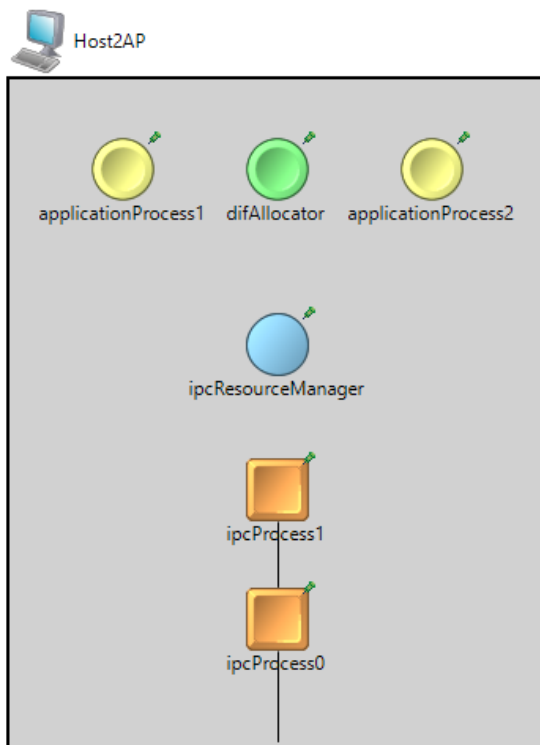
# Computation Systems ①





# Computation Systems ②

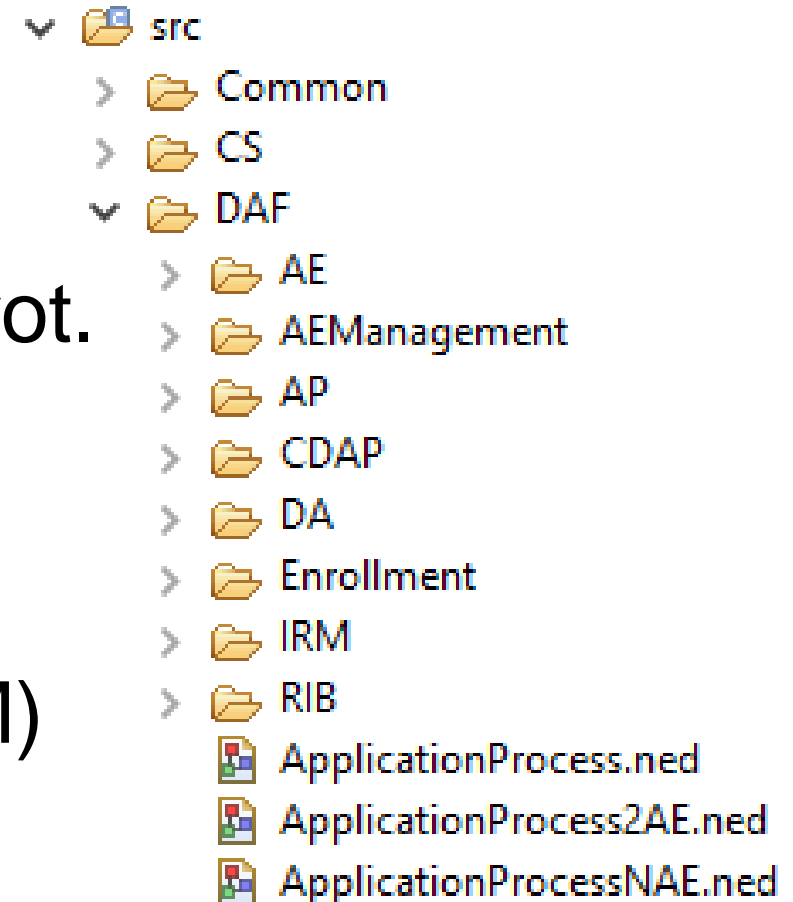
## Hosts and Routers



- src
  - Common
    - CS
      - BorderRouter.ned
      - BorderRouter2n1nm.ned
      - BorderRouter2n1nmLeft.ned
      - BorderRouter3n1nm.ned
      - DC\_AS.ned
      - DC\_CR.ned
      - DC\_Server.ned
      - DC\_TOR.ned
      - DC\_VM.ned
      - Host1AP\_Adm.ned
      - Host1AP.ned
      - Host1APNInt.ned
      - Host2AP.ned
      - HostNAP\_Adm.ned
      - HostNAP.ned
      - HostNAPNAE.ned
      - InteriorRouter2Int.ned
      - InteriorRouter3Int.ned
      - InteriorRouter4Int.ned
      - InteriorRouterNInt\_Adm.ned
      - InteriorRouterNInt.ned
      - InteriorRouterNMInt.ned
      - Shim\_M\_Adm.ned
      - TestHost1AP.ned
      - ViFIB\_MGM.ned
      - ViFIB.ned
      - ViFIBNode.ned
      - ViFIBNodeGL1.ned
      - ViFIBNodeGL2.ned

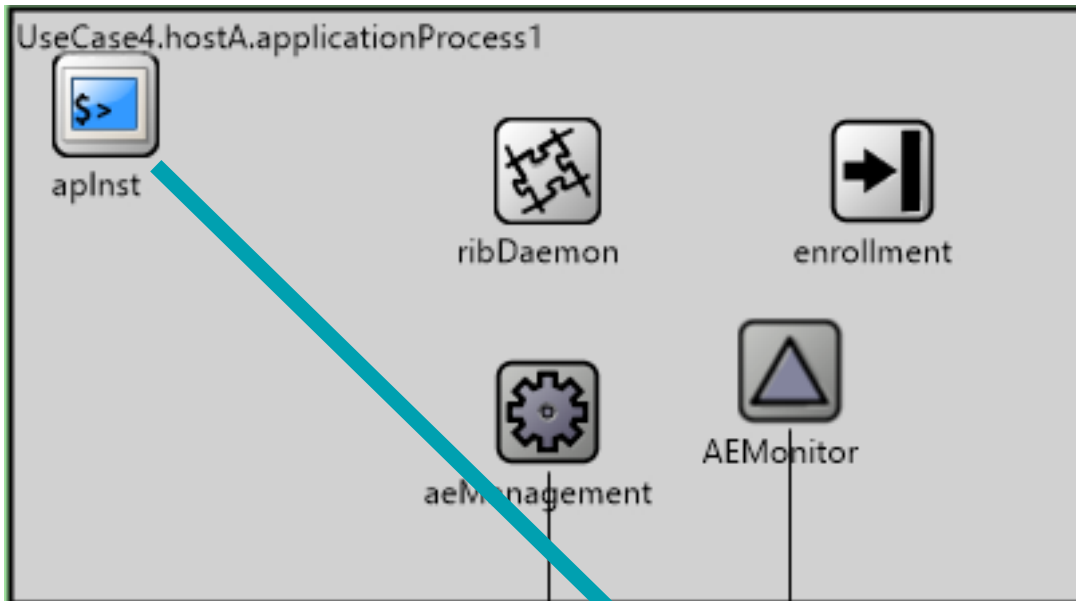
# DAF Components

- ◆ Application Process (AP)
- ◆ Application Entity (AE)
- ◆ Common Distribute Apod. Prot.
- ◆ DIF Allocator (DA)
- ◆ DAF Enrollment
- ◆ IPC Resource Manager (IRM)
- ◆ DAF RIB Daemon















# Application Process

- Contains a single AP instance



(ApplicationProcess) UseCase4.hostA.applicationProcess1

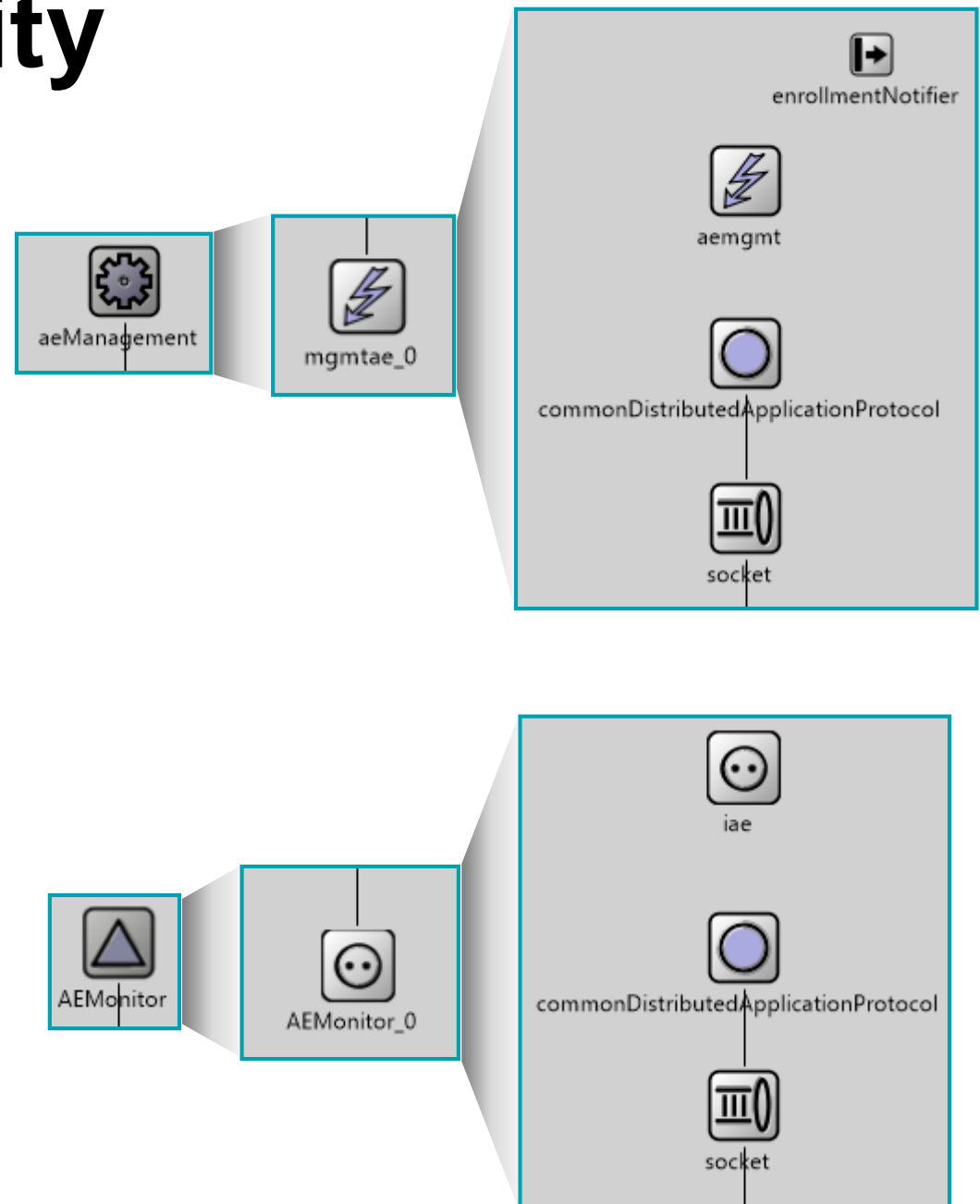
Fields	Contents (12)	
Class	Name	Info
 cPar	apName	"SourceA"
 cPar	apInstance	"0"
 cPar	apType	"APping"
 cGate	southlo\$i[0]	<-- ipcResourceManager.northlo\$0[0]
 cGate	southlo\$i[1]	<-- ipcResourceManager.northlo\$0[1]
 cGate	southlo\$0[0]	--> ipcResourceManager.northlo\$i[0]
 cGate	southlo\$0[1]	--> ipcResourceManager.northlo\$i[1]
 APping	aplnt	id= 14
 DAFRIBDaemon	ribDaemon	id=15
 DAFEnrollmentModule	enrollment	id= 16
 ApplicationEntity	aeManagement	id=17
 ApplicationEntity	AEMonitor	id=576

- Application code is written using CDAP API

```
if ( !strcmp(msg->getName(), "start") ) {  
    invokeId = getNewInvokeId();  
    a_open(invokeId, par("dstApName").stringValue(), "0", "AEMonitor", "-1");  
}  
else if (!strcmp(msg->getName(), "stop")) {  
    a_close(conID);  
}
```

# Application Entity

- AP has AE instances
  - suffix ***\_instanceID***
  - AEManagementModule instance for exchange of management CDAP messages
- Another AEs delivering IPC for AP instance



# Common Distributed App Protocol

- Simulation module used by AE and RIBd

## CDAP

- Sends/Receives CDAP messages

## CDAPSplitter

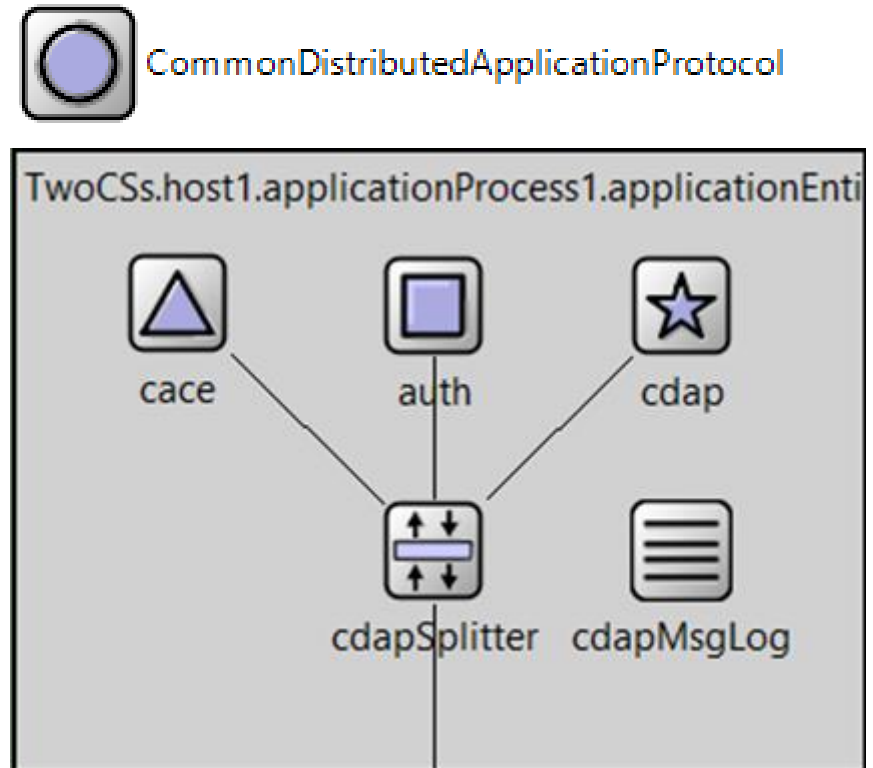
- Delegates CDAP message to appropriate module

## CDAPMsgLog

- Statistic collector

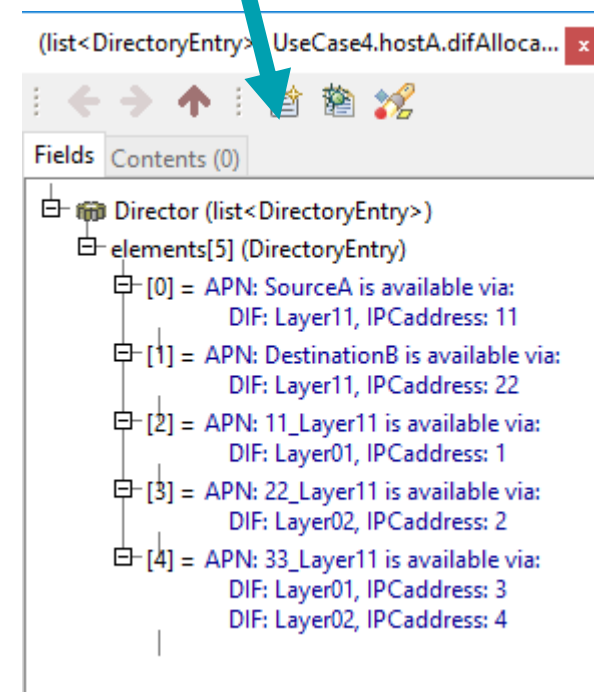
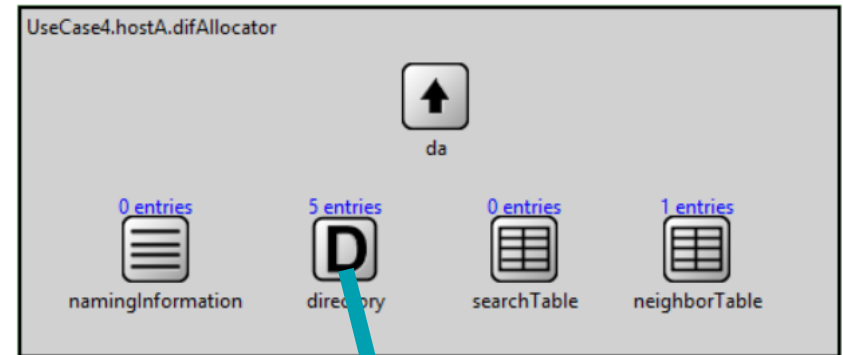
## CACE + AUTH

- Used by Enrollment or during authentication phase



# DIF Allocator

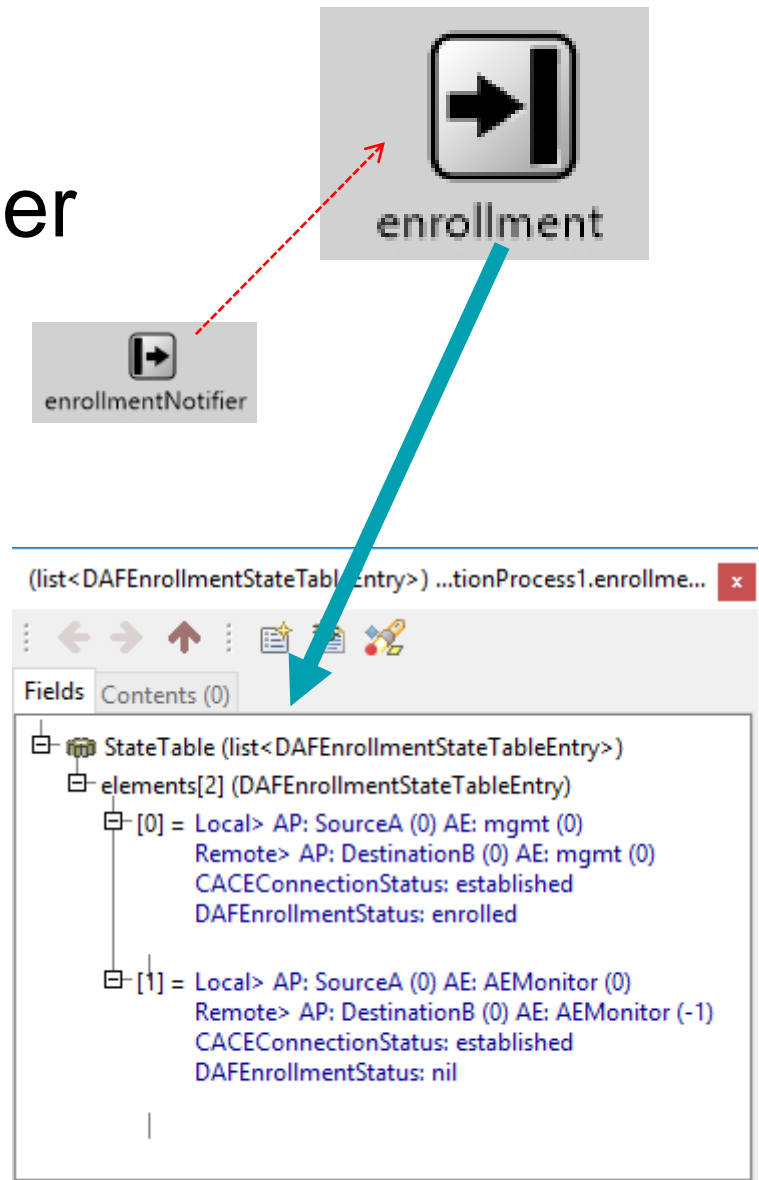
- ◆ *Knows how AP/IPCP is reachable via which IPCP*
- ◆ Configured externally
- ◆ „Unique“ naming scheme
  - ◆ whatever string allowing different interpretations
  - ◆ IPCP name is concatenation of unambiguous address and DIF name



Name	Info
ipAddress	"11"
difName	"Layer11"
apName	"11_Layer11"

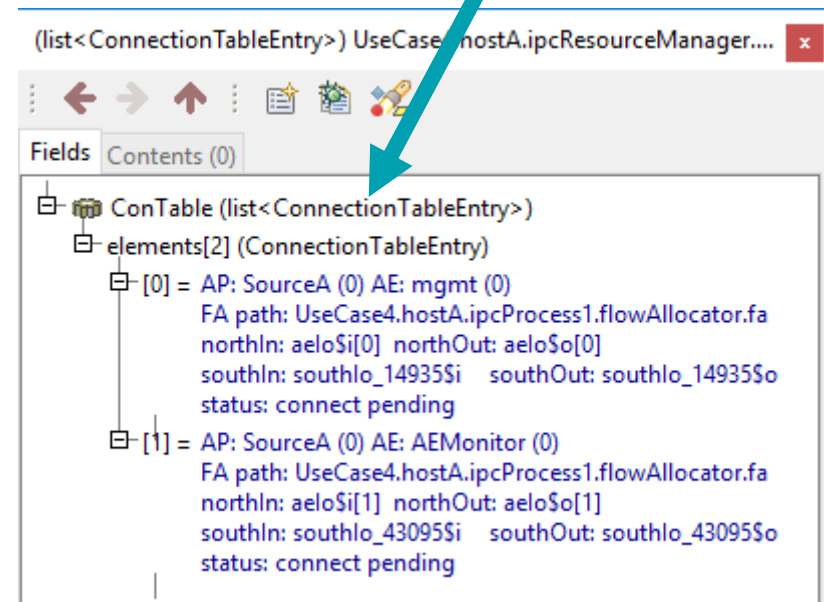
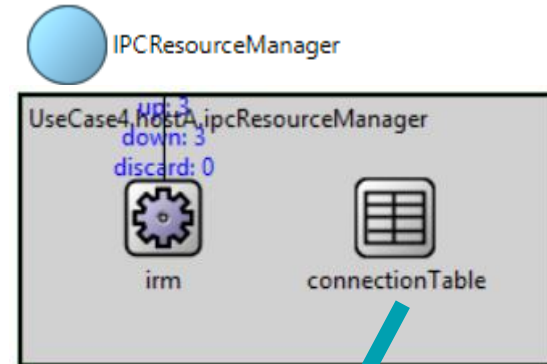
# DAF Enrollment

- Notified by Enrollment Notifier within AE
- Maintains state about AE's enrollment



# IPC Resource Manager

- Interconnects APs with IPCs
- Passes messages from applications to DIFs





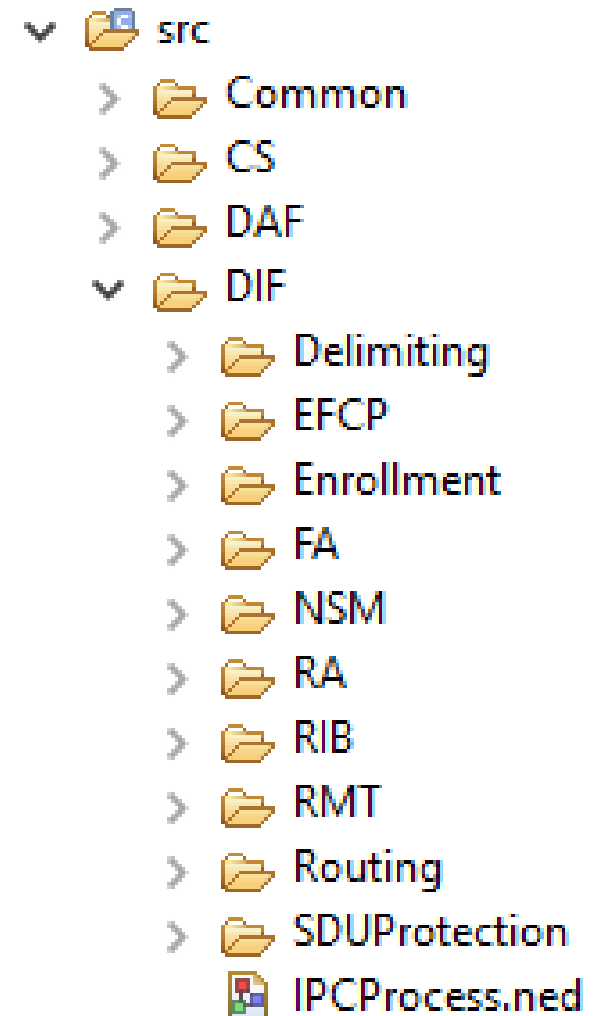
# DAF RIB Daemon

- ◆ Shared database for AP instance functionality
- ◆ Its functionality is not bound to other concrete components comparing to IPCP's RIBd

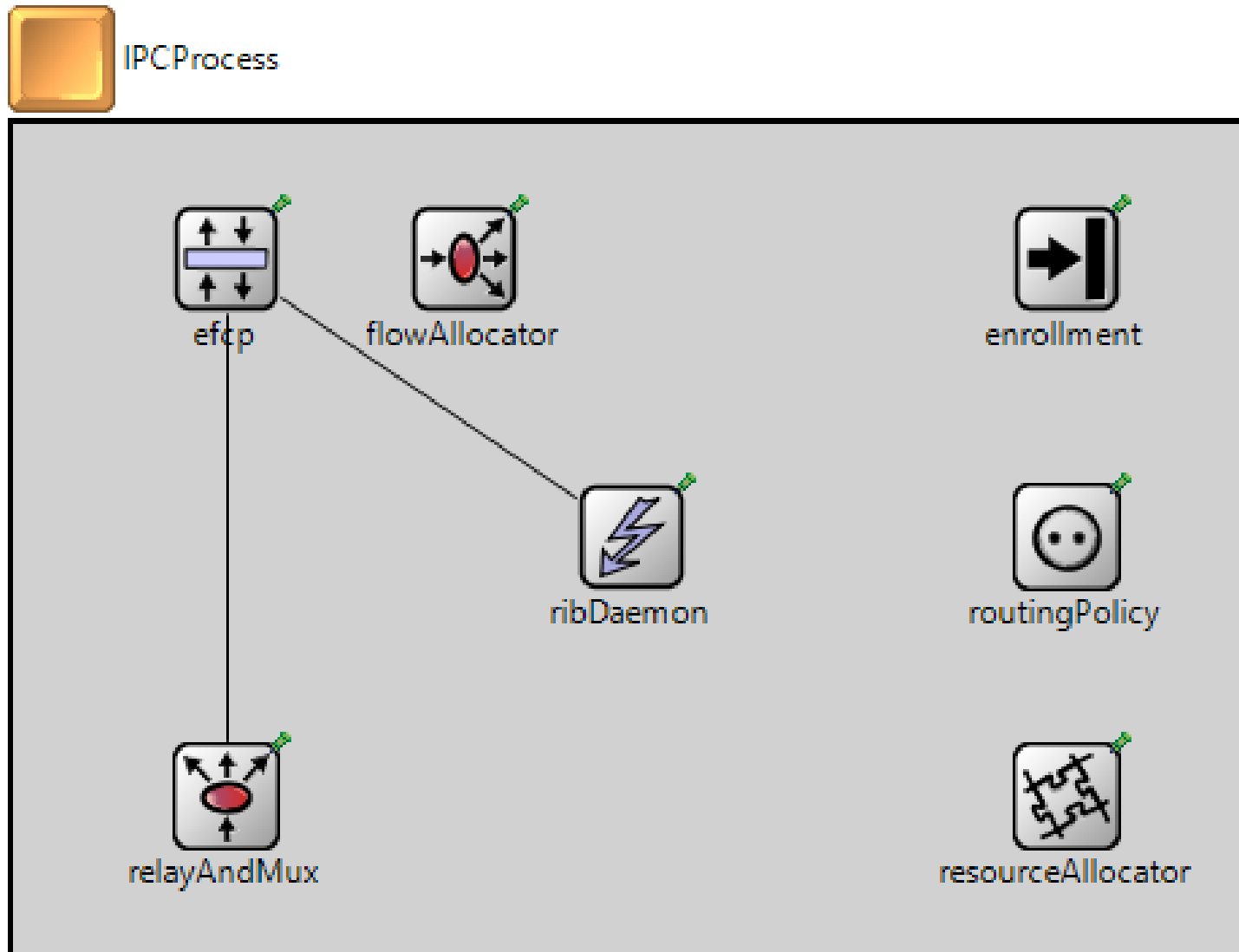


# DIF Components

- ◆ Flow Allocator (FA)
- ◆ Error and Flow Control Prot.
- ◆ Delimiting
- ◆ Enrollment
- ◆ RIB Daemon
- ◆ SDU Protection
- ◆ Relaying and Multiplexing Task
- ◆ Resource Allocator (RA)
- ◆ Routing

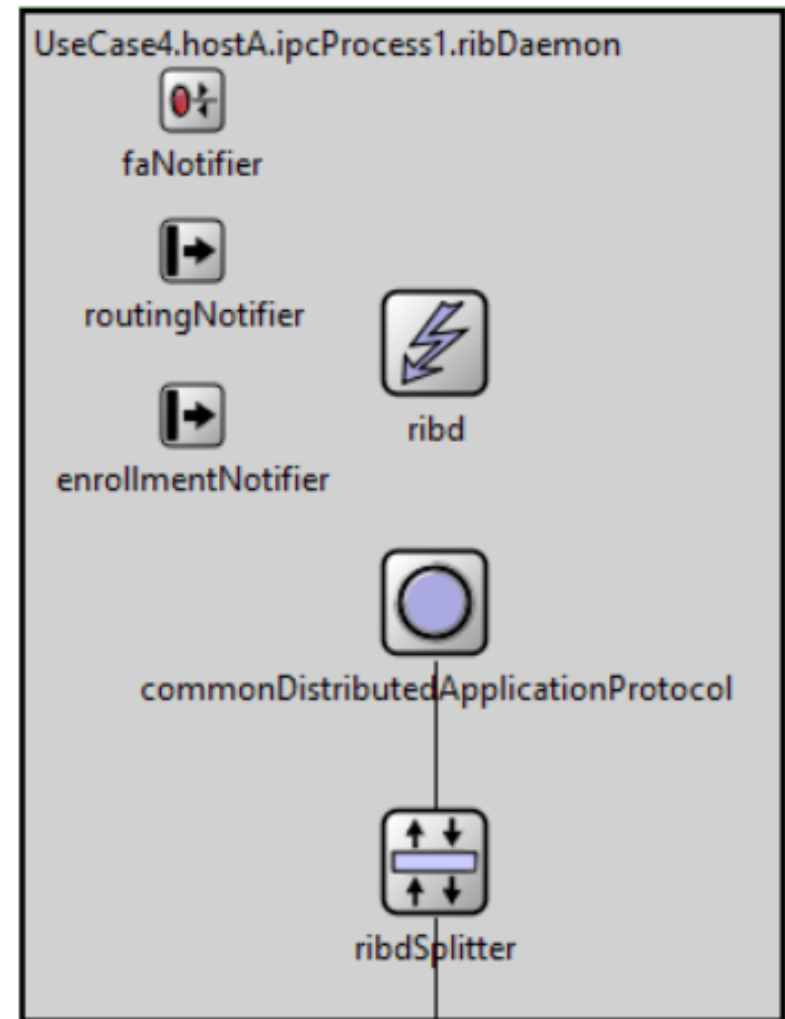


# IPC Process



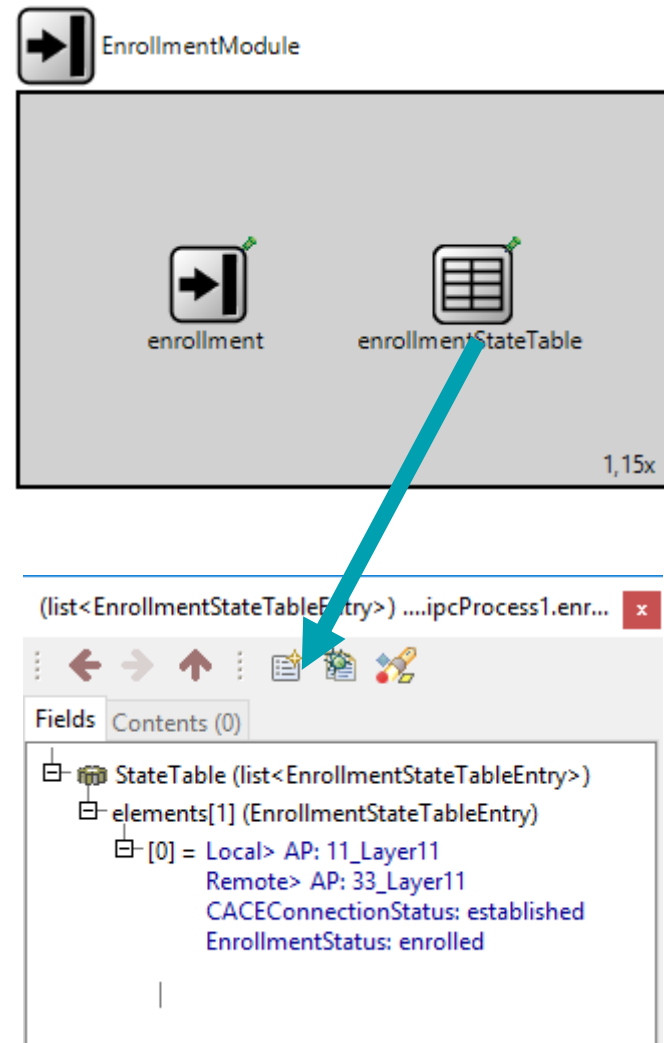
# RIB Daemon

- Generates / Processes IPCP management messages
- The main IPCP's AE
- RIBd**
  - Core functionality
- CDAP**
  - Socket-like message sender/receiver
- Notifiers**
  - FA, Routing, Enrollment hooks to RIBd
- RIBdSplitter**
  - Passes CDAP msgs to/from appropriate EFCPI
  - Placeholder for socket behavior



# DIF Enrollment

- ◆ Maintains enrollment status for a given DIF
- ◆ Used when allocating connections between IPCPs



# Flow Allocator

- Manages flow lifecycle
- FA**
  - Core functionality
- FAI\_portId\_cepld**
  - Instance
- NFlowTable**
  - Information about all (N)-DIF connections
- NewFlowRequestPolicy**
  - Score or Min compare
- AllocateRetryPolicy**
  - Upon treshhold reach
- QoSComparerPolicy**
  - For multi QoSCube routing purposes

The screenshot shows a network configuration tool window titled "(std::list<FAI\_TableEntry>) ...Relay.host1.ipcProcess1.fl...". The window has a "Fields" tab and a "Contents (0)" section. The main content area displays a tree structure of flow table entries. The first entry is "FaiTable (std::list<FAI\_TableEntry>)", which contains a sub-entry "FaiTable[2] (FAI\_TableEntry)". This sub-entry has two elements: "[0]" and "[1]". The "[0]" element is a flow entry with the following details: "STATUS: allocation positive, transfer \ FAI> (FAI)fai\_57043\_20757", "SRC> RIBd of AP: 11\_Layer11", "address: 11(Layer11)", "neighbor: 11(Layer11)", "port: 57043", "cep: 20757", "DST> RIBd of AP: 22\_Layer11", "address: 22(Layer11)", "neighbor: 33(Layer11)", "port: 38984", "cep: 6921", "Hop Count: 16", "Retries: 0/3", "DDT: no", "Chosen RA's QoS cube>MGMT-QoS Cube", "QoS Requirements List> average BW = 2048 bit/s, average SDU BW = 10 SDU/s, peak BW duration = 4096 bit/s, peak SDU BW duration = 20 SDU/s, burst period = 10000 usecs, burst duration = 10000 usecs, undetect. bit errors = 0%, PDU dropping probability = 0%, max SDU Size = 1500 B, partial delivery = no, incomplete delivery = no, force order = yes, max allowed gap = 0 SDUs, delay = 0 usecs, jitter = 0 usecs, cost-time = 0 \$/ms, cost-bits = 0 \$/Mb, Created at: 10, invld: 42613001, Deleted at: 0, invld: 0".

# Error and Flow Control Protocol

## ◆ EFCP

- ◆ Manages EFCP instances

## ◆ EFCPTable

- ◆ Table of known EFCPIs

## ◆ Delimiting\_portId

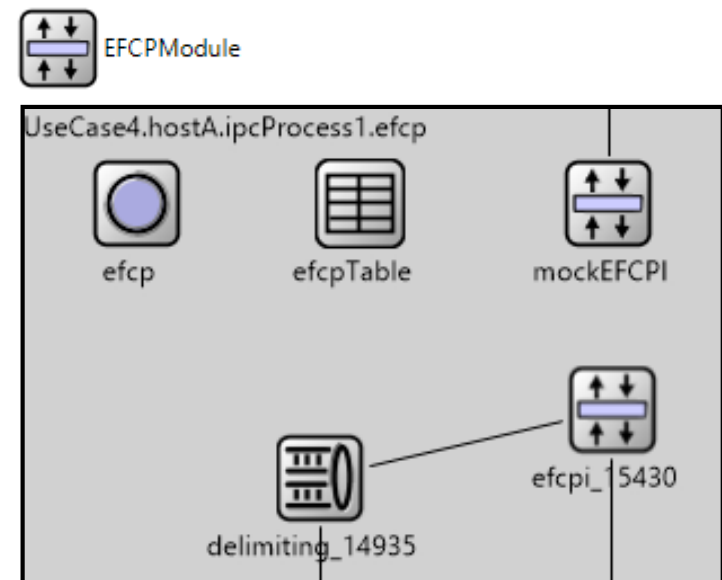
- ◆ Creates SDUs from incoming PDUs

## ◆ EFCPI\_cepId

- ◆ Provides DTP and DTCP services

## ◆ MockEFCPI

- ◆ Provides unreliable communication for IPCP management messages
- ◆ Simple en/decapsulater between SDUs and PDUs



# EFCP Instance

## DTP

- Actual Data Transfer

## DTCP

- Handles Flow Control and ReXmission

## DTPState

- Holds all DTP related variables

## DTCPState

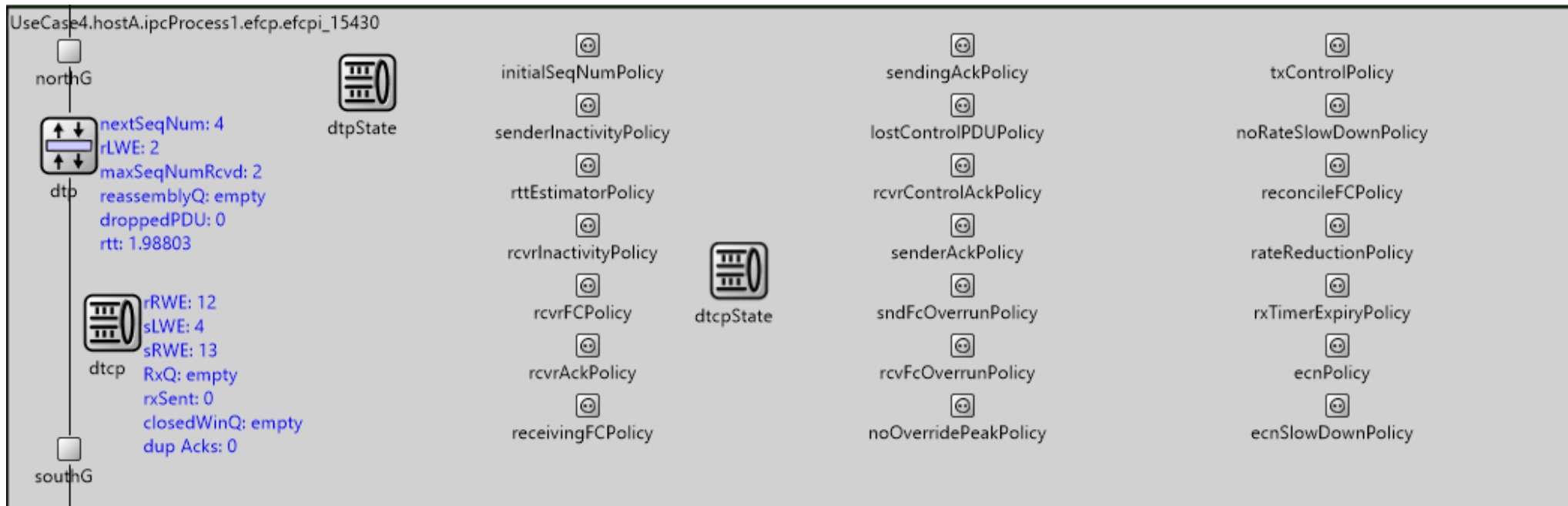
- Holds all DTCP related variables

(DTPState) UseCase4.hostA.ip...

Fields	Contents (2)
Class	Name Info
cPar	rtt 2
cPar	mpl 50

(DTCPState) UseCase4.hostA.ipcProcess1....

Fields	Contents (11)
Class	Name Info
cPar	initialSenderCredit 10
cPar	rcvCredit 10
cPar	aTime 0
cPar	nextSenderControlSe 1
cPar	dataReXmitMax 3
cPar	maxClosedWinQueLe 4
cPar	timeUnit 1000
cPar	sendingTimeUnit 1000
cPar	rcvBufferPercentThre 75
cPar	initialSendingRate 1
cPar	configRcvrRate 1



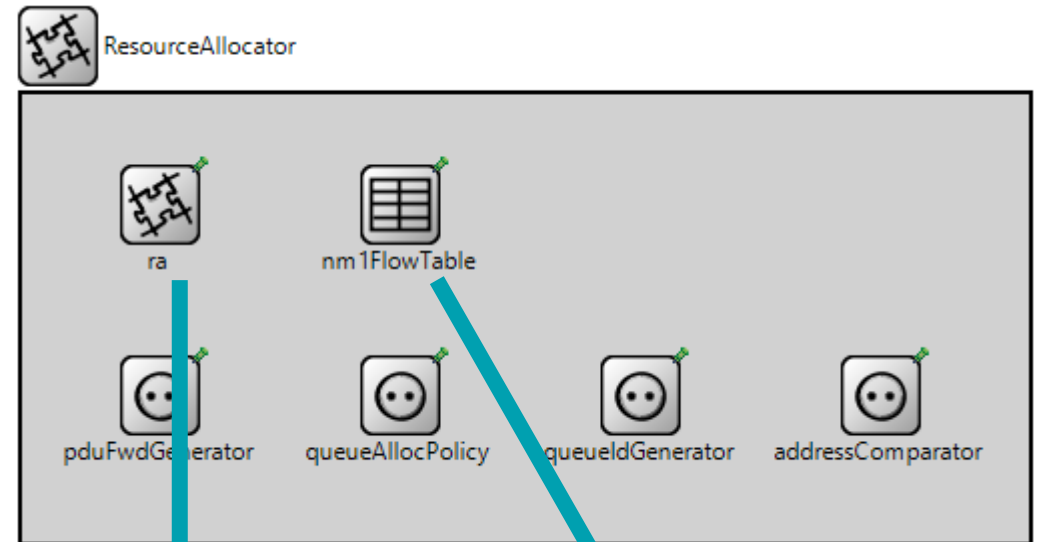
## EFCP policies

- Triggered during various DTP states



# Resource Allocator

- Provides access to (N-1)-DIFs and their resources
- **RA**
  - Core functionality
  - Manages IPCP's QoS Cubes
- **NM1FlowTable**
  - Information about current (N-1)-flows
- **PDUFwdGenerator**
  - Forwarding information management
- **QueueAllocPolicy**
  - How and when should RMT queues be allocated?
- **QueueIdGenerator**
  - In which RMT queue should a PDU be stored?
- **AddressComparator**
  - Syntax and comparison of addresses

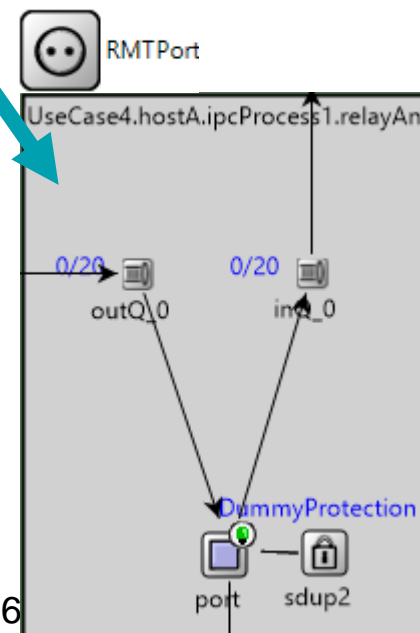
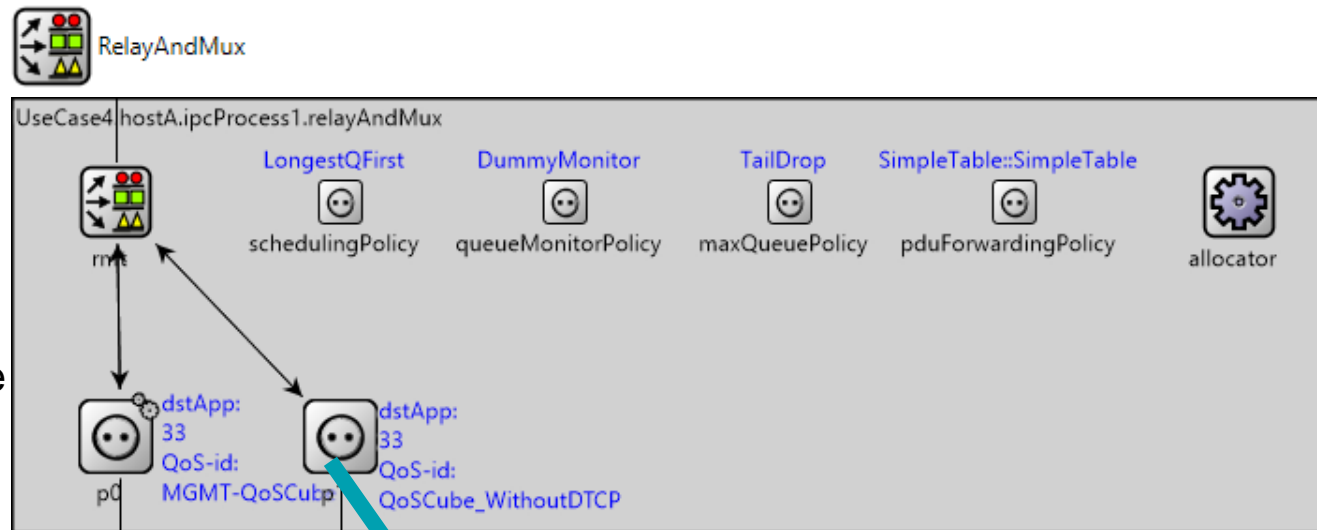


```
(std::list<QoS Cube> ...) hostA.ipcProcess1.resourceAlloca...
Fields Contents (0)
this->QoS Cubes (std::list<QoS Cube>)
  this->QoS Cubes[3] (QoS Cube)
    [0] = QoS Cube Id> 1 \ average BW = 12000000 bit/s \ average
    [1] = QoS Cube Id> 2 \ average BW = 12000000 bit/s \ average
    [2] = QoS Cube Id> MGMT-QoS Cube
          average BW = 2048 bit/s
          average SDU BW = 10 SDU/s
          peak BW duration = 4096 bit/s
          peak SDU BW duration = 20 SDU/s
          burst period = 10000 usecs
          burst duration = 10000 usecs
          undetect. bit errors = 0%
          PDU dropping probability = 0%
          max SDU Size = 1500 B
          partial delivery = no
          incomplete delivery = no
          force order = yes
          max allowed gap = 0 SDUs
          delay = 0 usecs
          jitter = 0 usecs
          cost-time = 0 $/ms
          cost-bits = 0 $/Mb
          A-Time = 0ms
```

```
(list<NM1FlowTableItem>) ...hostA.ipcProcess1.resourceAllocator.nm1FlowTable.f...
Fields Contents (0)
flows (list<NM1FlowTableItem>)
  elements[2] (NM1FlowTableItem)
    [0] = SRC> AP: 11_Layer11
          address: 1(Layer01), neighbor: 1(Layer01)
          port: 55026
          cep: 46884
          DST> AP: 33_Layer11
          address: 3(Layer01), neighbor: 3(Layer01)
          port: 48600
          cep: 39512
          Hop Count: 16
          Retries: 0/3
          DDT: no
          Chosen RA's QoS cube: MGMT-QoS Cube (aggregated)
          QoS Requirements List>
            average BW = 12000 bit/s, average SDU BW = 10 SDU/s
            peak BW duration = 24000 bit/s, peak SDU BW duration = 20 SDU/s
            burst period = 10000 usecs, burst duration = 10000 usecs
            undetect. bit errors = 0%, PDU dropping probability = 0%
            max SDU Size = 1500 B
            partial delivery = no, incomplete delivery = no
            force order = yes
            max allowed gap = 0 SDUs
            delay = 0 usecs, jitter = 0 usecs
            cost-time = 0 $/ms, cost-bits = 0 $/Mb
            resiliency factor = 0 usecs
          FA path: UseCase4.hostA.ipcProcess0.flowAllocator.fa
          status: established
          RMT port: port
    [1] = SRC> AP: 11_Layer11 \ address: 1(Layer01), neighbor: 1(Layer01) \ po
```

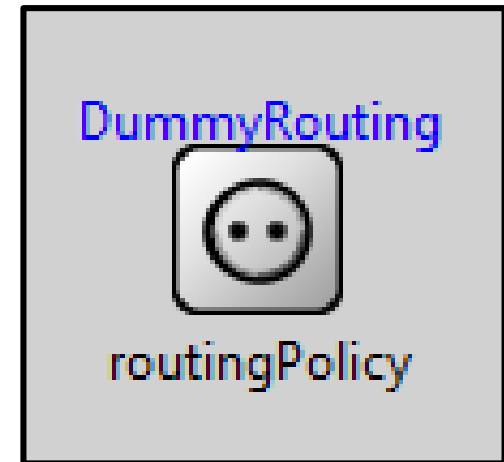
# Relaying and Multiplexing Task

- Relays incoming/outgoing PDU to proper destination (either an EFCP instance or an (N-1)-flow)
- **RMT**
  - The core PDU forwarder
- **SchedulingPolicy**
  - When a PDU needs to be sent/received, which queue should it be taken from?
- **QueueMonitorPolicy**
  - Keeping information about port/queue states
- **MaxQueuePolicy**
  - What should happen to a queue when it overflows?
- **PDUForwardingPolicy**
  - Where should be PDU relayed based on a given header?



# Routing

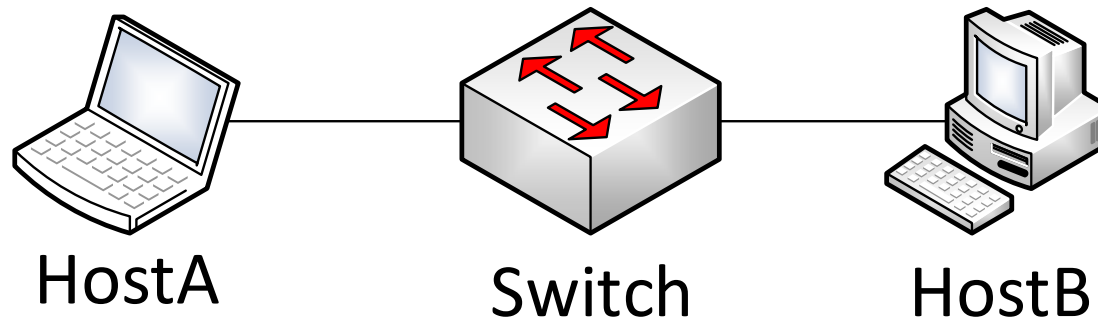
- ◆ The policy computing optimal paths to other destinations by given metrics
- ◆ Usually some sort of routing algorithm exchanging information with other members of a DIF



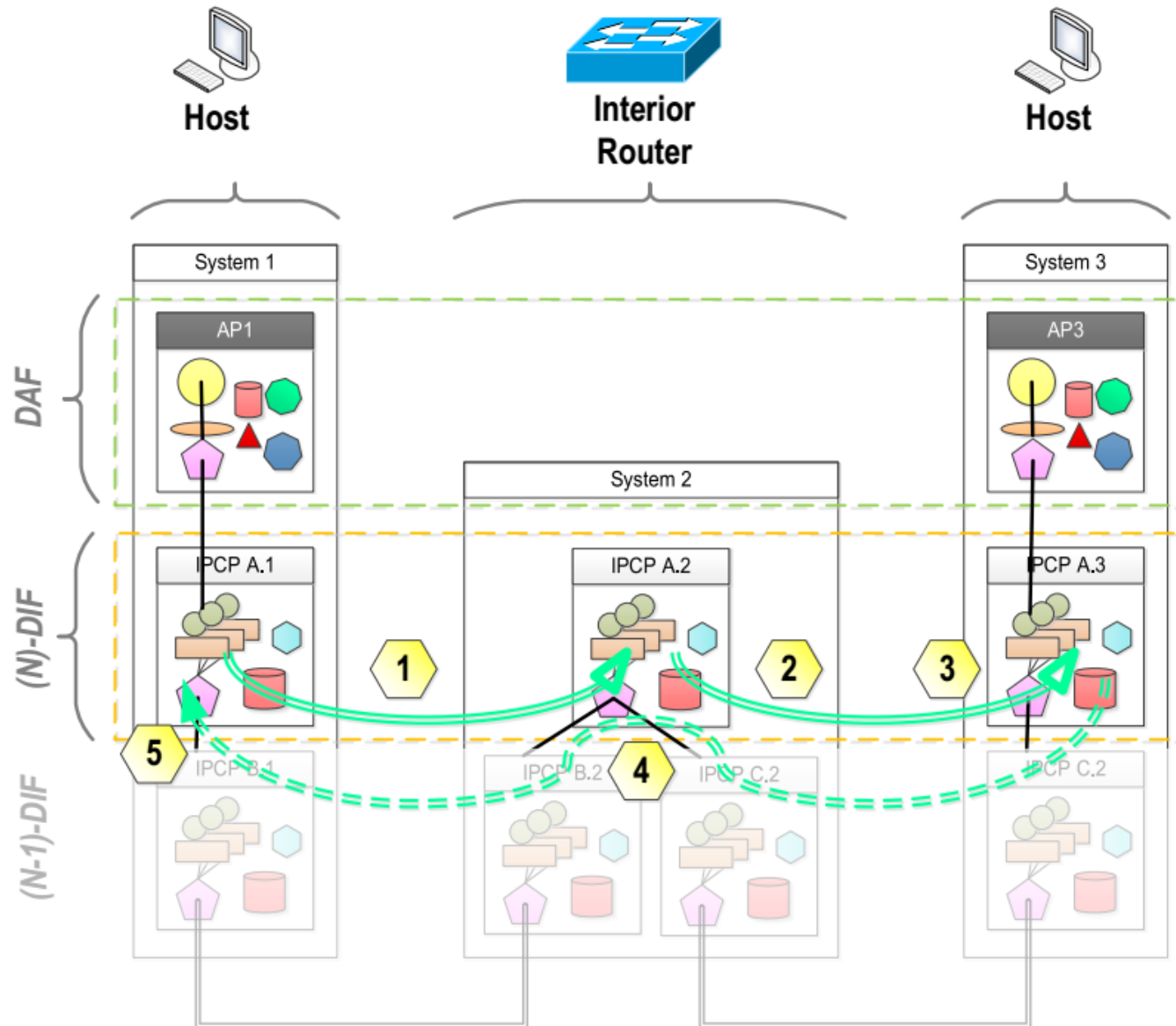


# Interactive Demo

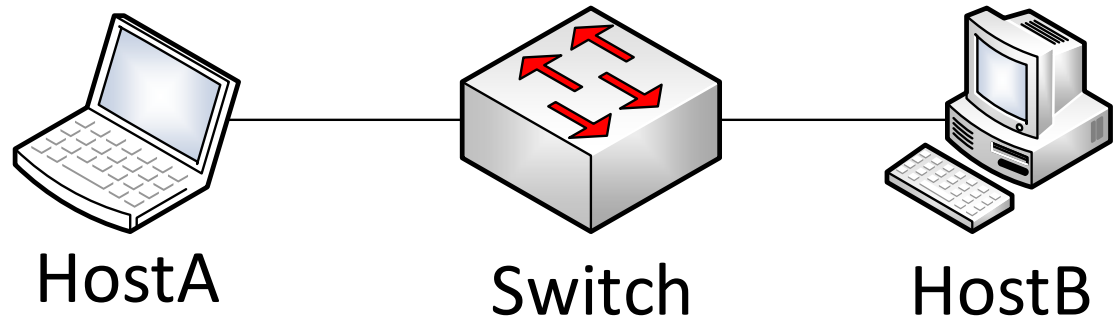
*How IPC works between two hosts interconnected to a common node?*



# Topology



# Cookbook



## ◆ Topology

- ◆ 2 × host with single AP
- ◆ 1 × interior router
- ◆ 2 × datarate channel between

## ◆ Task

- 1) Setup network
- 2) Schedule simulation
- 3) Run

## ◆ Goal

- ◆ To observe IPC between two hosts interconnected by a interior router

# 1) Setup network

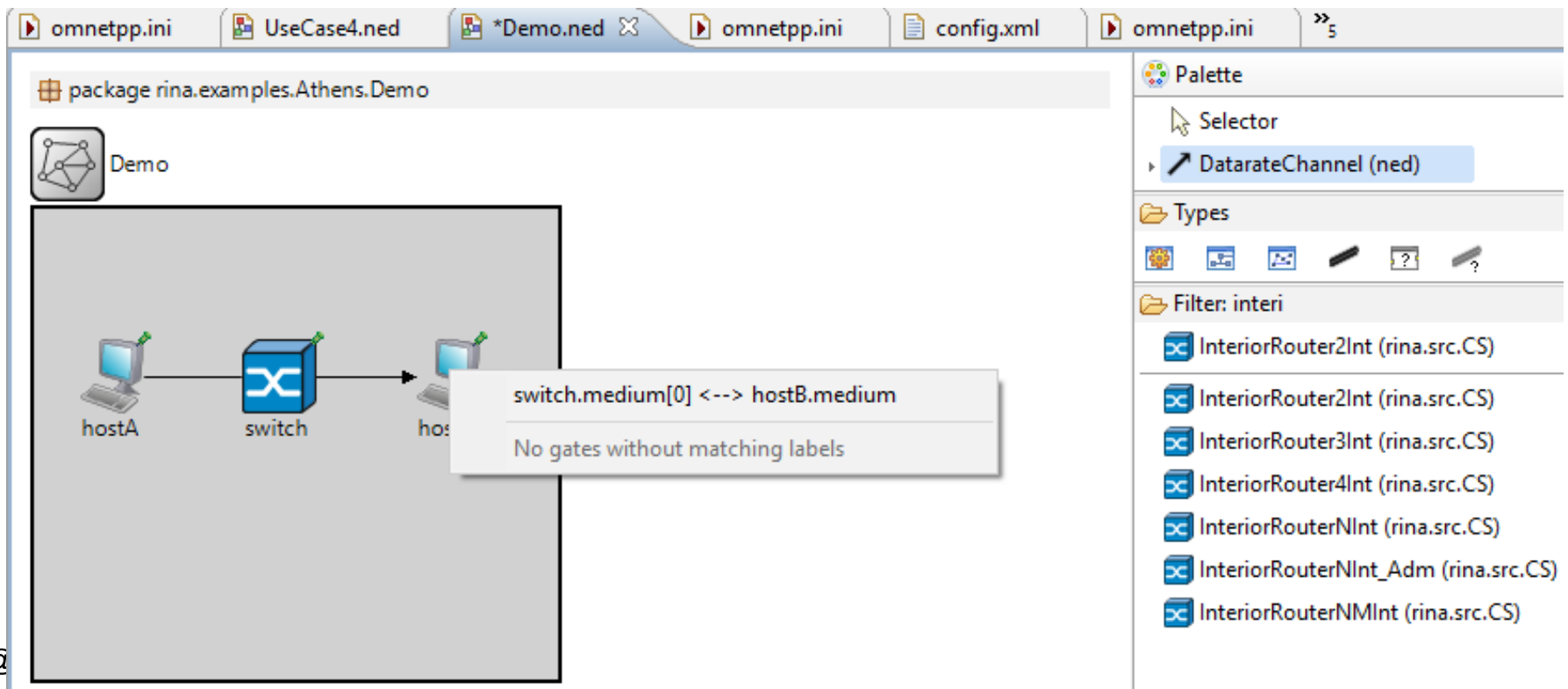
- ◆ Create new simulation in folder examples/Demos/Demo

The screenshot displays the NetSim IDE interface. On the left, the 'Project Explorer' shows a tree structure with folders like 'Binaries', 'Archives', 'Includes', 'policies', 'src', 'doc', and 'examples'. The 'examples' folder is expanded, showing subfolders like 'Athen', 'Basics', 'Conge', 'Demo', 'Us', 'out', 'playground', 'policies', 'scripts', and 'src'. A right-click context menu is open over the 'Demo' folder, with the 'New' option selected. The 'New' submenu is visible, showing options like 'Project...', 'File', 'File from Template', 'Folder', 'Analysis File (anf)', 'Class', 'Class (OMNeT++)', 'Compound Module', 'Header File', 'Initialization File (ini)', 'Message Definition (msg)', 'Network', 'Network Description File (NED)', 'Simple Module', 'Simulation', and 'Source File'. The 'Simulation' option is highlighted. A red arrow points from the 'Simulation' option in the context menu to the 'New Simulation' dialog box on the right.

The 'New Simulation' dialog box is titled 'New Simulation' and has a subtitle 'Choose folder for simulation files'. It contains a text field labeled 'Enter or select the parent folder:' with the value 'rina/examples/Demos'. Below this is a tree view showing the folder structure, with 'examples' expanded and 'Demos' selected. At the bottom, there is a text field labeled 'Name for new folder:' with the value 'Demo'. There are buttons for '?', '< Back', 'Next >', and 'Finish'.

# 1) Setup network

- ◆ Open Demo.ned and add two Host1AP onto canvas and one InteriorRouter2Int
- ◆ Rename them with F6
- ◆ Connect them with DatarateChannel

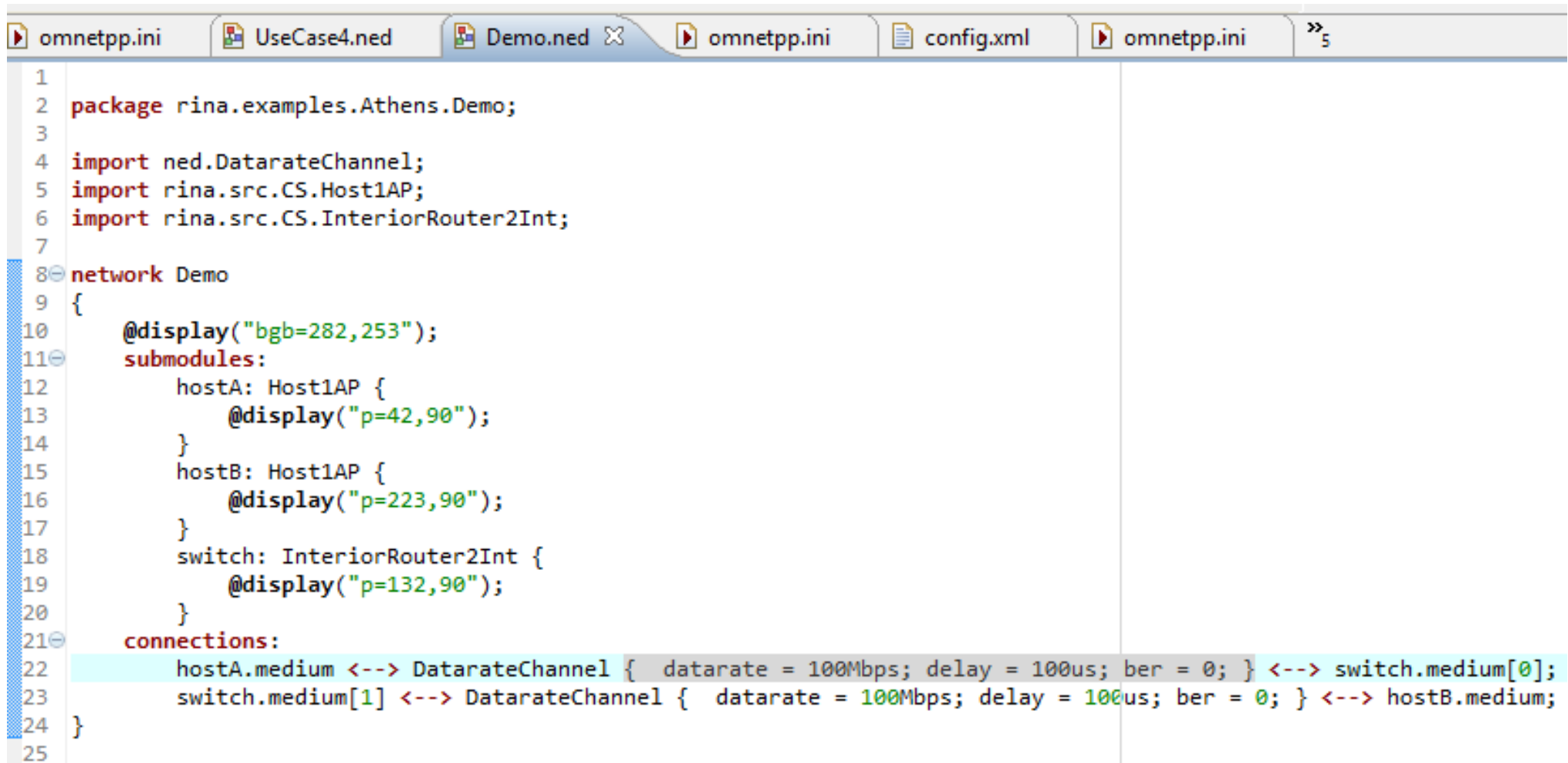




# 1) Setup network

## ◆ Change DatarateChannel properties

### ◆ Setup delay, ber, datarate



```
1 package rina.examples.Athens.Demo;
2
3
4 import ned.DatarateChannel;
5 import rina.src.CS.Host1AP;
6 import rina.src.CS.InteriorRouter2Int;
7
8 network Demo
9 {
10     @display("bgb=282,253");
11     submodules:
12         hostA: Host1AP {
13             @display("p=42,90");
14         }
15         hostB: Host1AP {
16             @display("p=223,90");
17         }
18         switch: InteriorRouter2Int {
19             @display("p=132,90");
20         }
21     connections:
22         hostA.medium <--> DatarateChannel { datarate = 100Mbps; delay = 100us; ber = 0; } <--> switch.medium[0];
23         switch.medium[1] <--> DatarateChannel { datarate = 100Mbps; delay = 100us; ber = 0; } <--> hostB.medium;
24 }
25
```

## 2) Schedule simulation

- ◆ `config.xml`

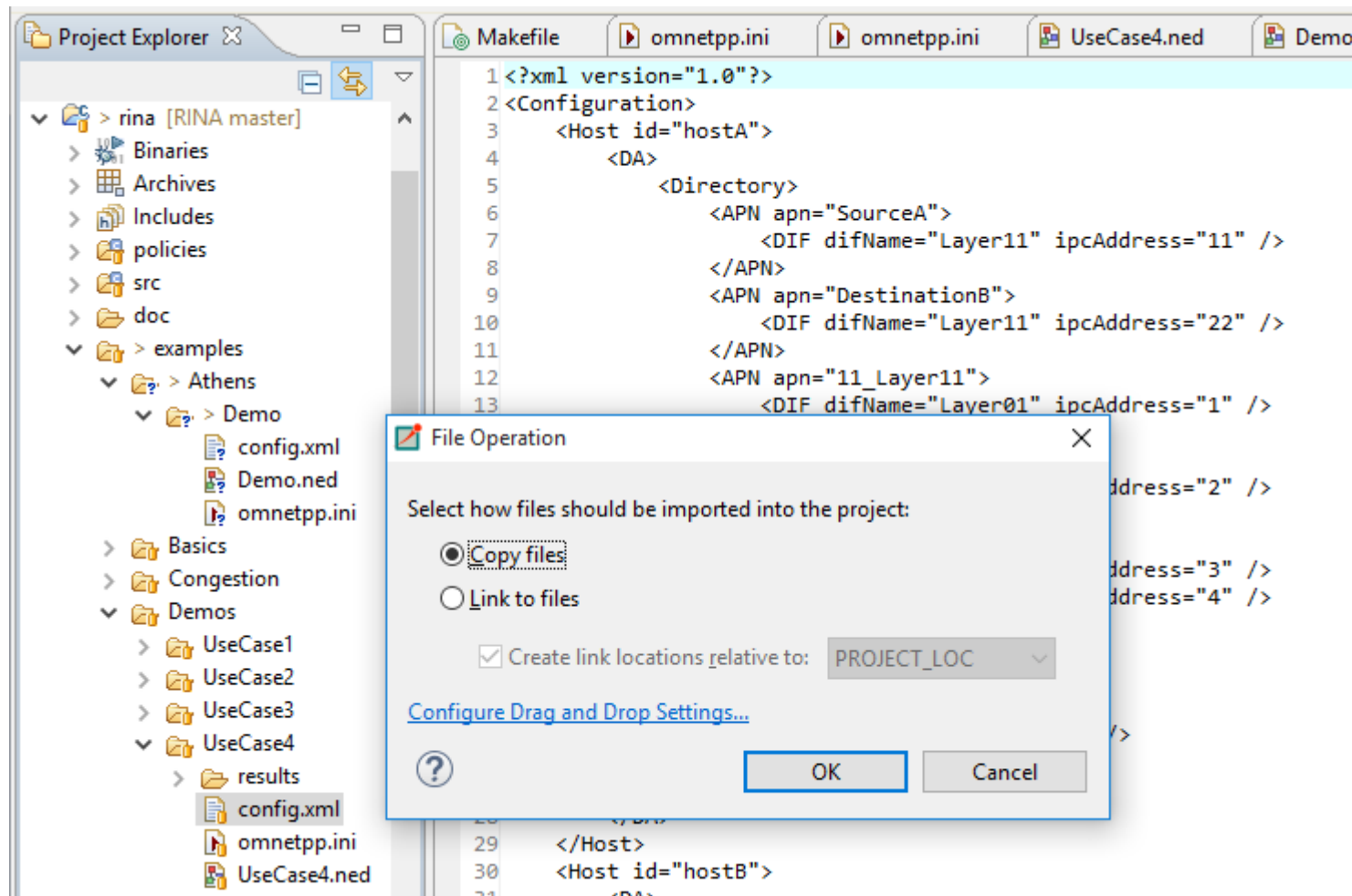
- ◆ DIF Allocator settings
- ◆ Schedule enrollment events
- ◆ Available QoS Cubes

- ◆ `omnetpp.ini`

- ◆ Assign addresses
- ◆ Bind `config.xml`
- ◆ Schedule AEMyPing

## 2) Schedule simulation: config.xml

- Copy config.xml from examples/Demos/UseCase4/config.xml



## 2) Schedule simulation: config.xml

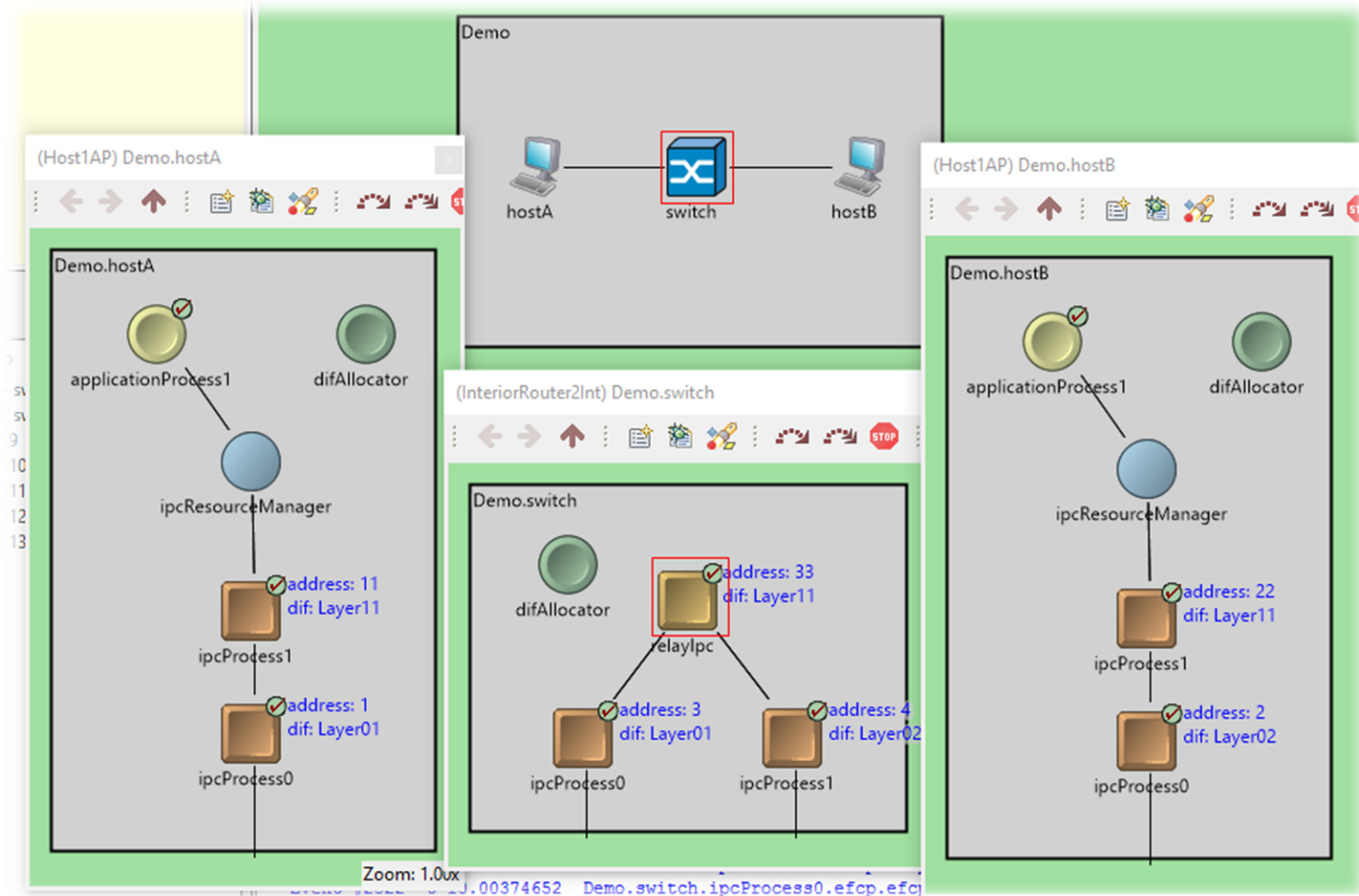
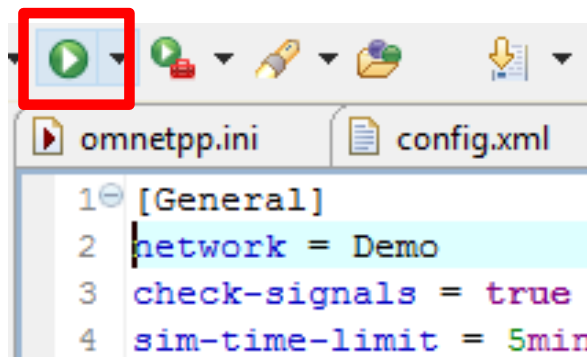
```
1 <?xml version="1.0"?>
2 <Configuration>
3   <Host id="hostA">
4     <DA>
5       <Directory>
6         <APN apn="SourceA">
7           <DIF difName="Layer11" ipcAddress="11" />
8         </APN>
9         <APN apn="DestinationB">
10          <DIF difName="Layer11" ipcAddress="22" />
11        </APN>
12        <APN apn="11_Layer11">
13          <DIF difName="Layer01" ipcAddress="1" />
14        </APN>
15        <APN apn="22_Layer11">
16          <DIF difName="Layer02" ipcAddress="2" />
17        </APN>
18        <APN apn="33_Layer11">
19          <DIF difName="Layer01" ipcAddress="3" />
20          <DIF difName="Layer02" ipcAddress="4" />
21        </APN>
22      </Directory>
23      <NeighborTable>
24        <APN apn="22_Layer11">
25          <Neighbor apn="33_Layer11" />
26        </APN>
27      </NeighborTable>
28    </DA>
29  </Host>
30  <Host id="hostB">
31    <DA>
32      <Directory>
33        <APN apn="SourceA">
34          <DIF difName="Layer11" ipcAddress="11" />
35        </APN>
36        <APN apn="DestinationB">
37          <DIF difName="Layer11" ipcAddress="22" />
38        </APN>
39        <APN apn="11_Layer11">
40          <DIF difName="Layer01" ipcAddress="1" />
41        </APN>
42        <APN apn="22_Layer11">
43          <DIF difName="Layer02" ipcAddress="2" />
44        </APN>
45        <APN apn="33_Layer11">
46          <DIF difName="Layer01" ipcAddress="3" />
47          <DIF difName="Layer02" ipcAddress="4" />
48        </APN>
49      </Directory>
50      <NeighborTable>
51        <APN apn="11_Layer11">
52          <Neighbor apn="33_Layer11" />
53        </APN>
54      </NeighborTable>
55    </DA>
56    <Enrollment>
57      <Preenrollment>
58        <SimTime t="5">
59          <Connect src="22_Layer11" dst="33_Layer11" />
60        </SimTime>
61      </Preenrollment>
62    </Enrollment>
63  </Host>
```

```
64 <Router id="switch">
65   <DA>
66     <Directory>
67       <APN apn="SourceA">
68         <DIF difName="Layer11" ipcAddress="11" />
69       </APN>
70       <APN apn="DestinationB">
71         <DIF difName="Layer11" ipcAddress="22" />
72       </APN>
73       <APN apn="11_Layer11">
74         <DIF difName="Layer01" ipcAddress="1" />
75       </APN>
76       <APN apn="22_Layer11">
77         <DIF difName="Layer02" ipcAddress="2" />
78       </APN>
79       <APN apn="33_Layer11">
80         <DIF difName="Layer01" ipcAddress="3" />
81         <DIF difName="Layer02" ipcAddress="4" />
82       </APN>
83     </Directory>
84   </DA>
85 </Router>
86 <QoSConfigSet>
87   <QoSConfig id="QoSConfig_WithoutDTCP">
88     <AverageBandwidth>10000000</AverageBandwidth>
89     <AverageSDUBandwidth>1000</AverageSDUBandwidth>
90     <PeakBandwidthDuration>20000000</PeakBandwidthDuration>
91     <PeakSDUBandwidthDuration>2000</PeakSDUBandwidthDuration>
92     <BurstPeriod>10000000</BurstPeriod>
93     <BurstDuration>1000000</BurstDuration>
94     <UndetectedBitError>0.01</UndetectedBitError>
95     <PDUDroppingProbability>0</PDUDroppingProbability>
96     <MaxSDUSize>1500</MaxSDUSize>
97     <PartialDelivery>0</PartialDelivery>
98     <IncompleteDelivery>0</IncompleteDelivery>
99     <ForceOrder>0</ForceOrder>
100    <MaxAllowableGap>0</MaxAllowableGap>
101    <Delay>1000000</Delay>
102    <Jitter>500000</Jitter>
103    <CostTime>0</CostTime>
104    <CostBits>0</CostBits>
105  </QoSConfig>
106  <QoSConfig id="QoSConfig_WithDTCP">
107    <AverageBandwidth>10000000</AverageBandwidth>
108    <AverageSDUBandwidth>1000</AverageSDUBandwidth>
109    <PeakBandwidthDuration>20000000</PeakBandwidthDuration>
110    <PeakSDUBandwidthDuration>2000</PeakSDUBandwidthDuration>
111    <BurstPeriod>10000000</BurstPeriod>
112    <BurstDuration>1000000</BurstDuration>
113    <UndetectedBitError>0.01</UndetectedBitError>
114    <PDUDroppingProbability>0</PDUDroppingProbability>
115    <MaxSDUSize>1500</MaxSDUSize>
116    <PartialDelivery>0</PartialDelivery>
117    <IncompleteDelivery>0</IncompleteDelivery>
118    <ForceOrder>1</ForceOrder>
119    <MaxAllowableGap>0</MaxAllowableGap>
120    <Delay>1000000</Delay>
121    <Jitter>500000</Jitter>
122    <CostTime>0</CostTime>
123    <CostBits>0</CostBits>
124  </QoSConfig>
125 </QoSConfigSet>
126 </Configuration>
```

## 2) Schedule simulation: omnetpp.ini

```
1 [General]
2 network = Demo
3 check-signals = true
4 sim-time-limit = 5min
5 debug-on-errors = true
6 #Application setup
7 **.hostA.applicationProcess1.apName = "SourceA"
8 **.hostB.applicationProcess1.apName = "DestinationB"
9 **.apType = "APPing"
10 #Static addressing
11 # Bottom DIF HostA<->Switch
12 **.hostA.ipcProcess0.ipcAddress = "1"
13 **.hostA.ipcProcess0.difName = "Layer01"
14 **.switch.ipcProcess0.ipcAddress = "3"
15 **.switch.ipcProcess0.difName = "Layer01"
16 # Bottom DIF HostB<->Switch
17 **.hostB.ipcProcess0.ipcAddress = "2"
18 **.hostB.ipcProcess0.difName = "Layer02"
19 **.switch.ipcProcess1.ipcAddress = "4"
20 **.switch.ipcProcess1.difName = "Layer02"
21 # Top DIF HostA<->Switch<->HostB
22 **.hostA.ipcProcess1.ipcAddress = "11"
23 **.hostB.ipcProcess1.ipcAddress = "22"
24 **.host*.ipcProcess1.difName = "Layer11"
25 **.switch.relayIpc.ipcAddress = "33"
26 **.switch.relayIpc.difName = "Layer11"
27 #DIF Allocator settings
28 **.hostA.difAllocator.configData = xmldoc("config.xml", "Configuration/Host[@id='hostA']/DA")
29 **.hostB.difAllocator.configData = xmldoc("config.xml", "Configuration/Host[@id='hostB']/DA")
30 **.switch.difAllocator.configData = xmldoc("config.xml", "Configuration/Router[@id='switch']/DA")
31 #Enrollment settings
32 **.switch.**.enrollment.isSelfEnrolled = true
33 **.hostB.ipcProcess1.enrollment.configData = xmldoc("config.xml", "Configuration/Host[@id='hostB']/Enrollment")
34 #QoS Cube sets
35 **.ra.qoscubesData = xmldoc("config.xml", "Configuration/QoS Cubes Set")
36
37 [Config Ping]
38 fingerprint = "dff8-6343"
39 #PingApp setup
40 **.hostA.applicationProcess1.apInst.dstApName = "DestinationB"
41 **.hostA.applicationProcess1.apInst.startAt = 10s
42 **.hostA.applicationProcess1.apInst.stopAt = 20s
43 **.hostA.applicationProcess1.AEMonitor.**.iae.size = 1024B
```

### 3) Run



# Notable Events

## ◆ t=5

- ◆ hostB enrolls to Layer02 and Layer11

## ◆ t=10

- ◆ hostA creates flows for AP communication

## ◆ t=10.3

- ◆ SourceA and DestinationB apps exchange ping messages

## ◆ t=20

- ◆ hostA deallocates Layer11 flow



## 5) Conclusion

*Final remarks*

*Usual problems*



# Summary

## ◆ Educational tool

- ◆ A way how to visualize what is happening in the native RINA network
- ◆ Helps to improve learning curve

## ◆ Research tool

- ◆ Check the growing list of citations  
<https://rinasim.omnetpp.org/#references>

# Something Is Not Working

- 1) Simulation crashes before it starts
  - ◆ Serious bug in your NED/C++ code
- 2) Simulation crashes after initialization
  - ◆ Error in static preconfiguration
    - check omnetpp.ini and config.xml
- 3) Simulation does not do what is supposed to
  - ◆ Error in your code logic
    - check the output of the Console
    - try to debug using GDB

# Other Topics

## ◆ We did not cover

- ◆ *How to play with policies? How to implement own policy?*
- ◆ *How to collect and interpret results?*

## ◆ Interested?

# You Want to Use RINASim

## ◆ Messages like:

my research project is to study architecture with RINA (Recursive Internetworking Architecture). I am willing to work with RINASim simulator, but unfortunately does not work with programming. I intend to work with simulator to collect data and analyze with another framework (INET). The goal of my research is to make a performance study between RINA architectures and TCP/IP using the RINA and INET frameworks.

Initially I was thinking of running some tests using wireless models with RINASim and then comparing them to the INET Framework models. Since I am not fluent in OMNeT, using the 4.6 version is an extreme pain because I haven't seen any working examples other than with LTE, which is currently out of my scope. OMNeT 5.0 finally has a semi-understandable wireless tutorial so I was trying to make my life easier.

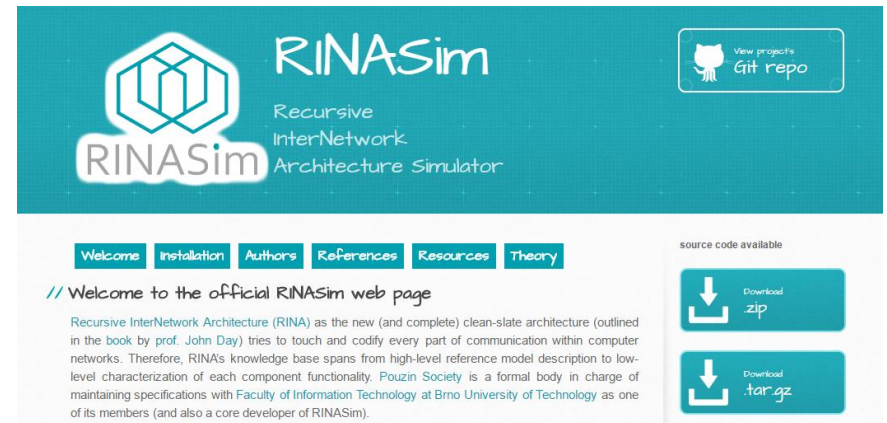
## ◆ Currently, RINASim does not offer any real data-link layer technology

## ◆ In order to use RINASim, you need to be programmer whether you like it or not 😊

# Need Help?

- Check the official webpage

- Visit <https://rinasim.omnetpp.org>



- Skype group chat

- [skype:?chat&blob=-bdq6qH\\_uDXIlbRk\\_4\\_XwqZyplfXPI4IzCq4P-S0BrsttjgPR8CNJKV9-Yyn1TYopaYZD2g3bIC\\_Yv0C](skype:?chat&blob=-bdq6qH_uDXIlbRk_4_XwqZyplfXPI4IzCq4P-S0BrsttjgPR8CNJKV9-Yyn1TYopaYZD2g3bIC_Yv0C)

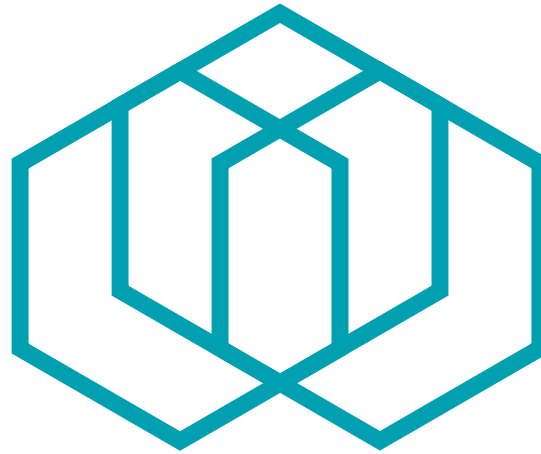
- <https://join.skype.com/B9Tt5aTPd0nC>



- Sign to mailing-list [rinasim@fit.vutbr.cz](mailto:rinasim@fit.vutbr.cz)

- Use <http://www.fit.vutbr.cz/mailman/listinfo/rinasim>





RINASim

*Thank you for your attention!*