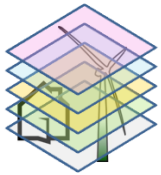
SmarterEMC2 project



Software Network Topology and Traffic Simulation for Overlay Telecommunication Network in Smart Grids

Velissarios Gezerlis, Ph.D.
Cosmote S.A. New Technologies Labs

25 - 10 -2017

COSMOTE
our world is you

SmarterEMC2 project

Objectives

1. Validation of ICT **(Information and Communications Technology) tools** that support the integration of consumers through Demand-Response services and the integration of DG/RES through Virtual Power Plants.
2. Exploration of new ICT tools that facilitate the management and operation of the Distribution Grid.
3. Demonstration of the innovative use of ICT tools in real-life **pilot sites.**
4. **Evaluation of the current communication infrastructure through hardware and software simulation.**
5. Contribution to **SG standardization** with emphasis on market-oriented and field-level standards.
6. Proposition of new business models that anticipate Active Customer Participation, increased RES penetration, Distribution Grid reliable operation, and open access to the Electricity Market.

SmarterEMC2 project

Partners

INTRACOM TELECOM

Durham University

Institute of Communication and Computer Systems (ICCS)

INESCTEC TECHNOLOGY & SCIENCE | ASSOCIATE LABORATORY | COORDINATED BY INESCPORTO PORTUGAL

Instituto de Engenharia de Sistemas e Computadores do Porto

AALBORG UNIVERSITY DENMARK

THALES

Thales Italia (THALIT) is the Italian subsidiary of the French Thales Group

Aydem

Aydem Electricty Distribution Co in Turkey

Elder

Electricity Distribution Services (ELDER) in Turkey

HEDNO

Hellenic Electricity Distribution Network Operator S.A

OTE

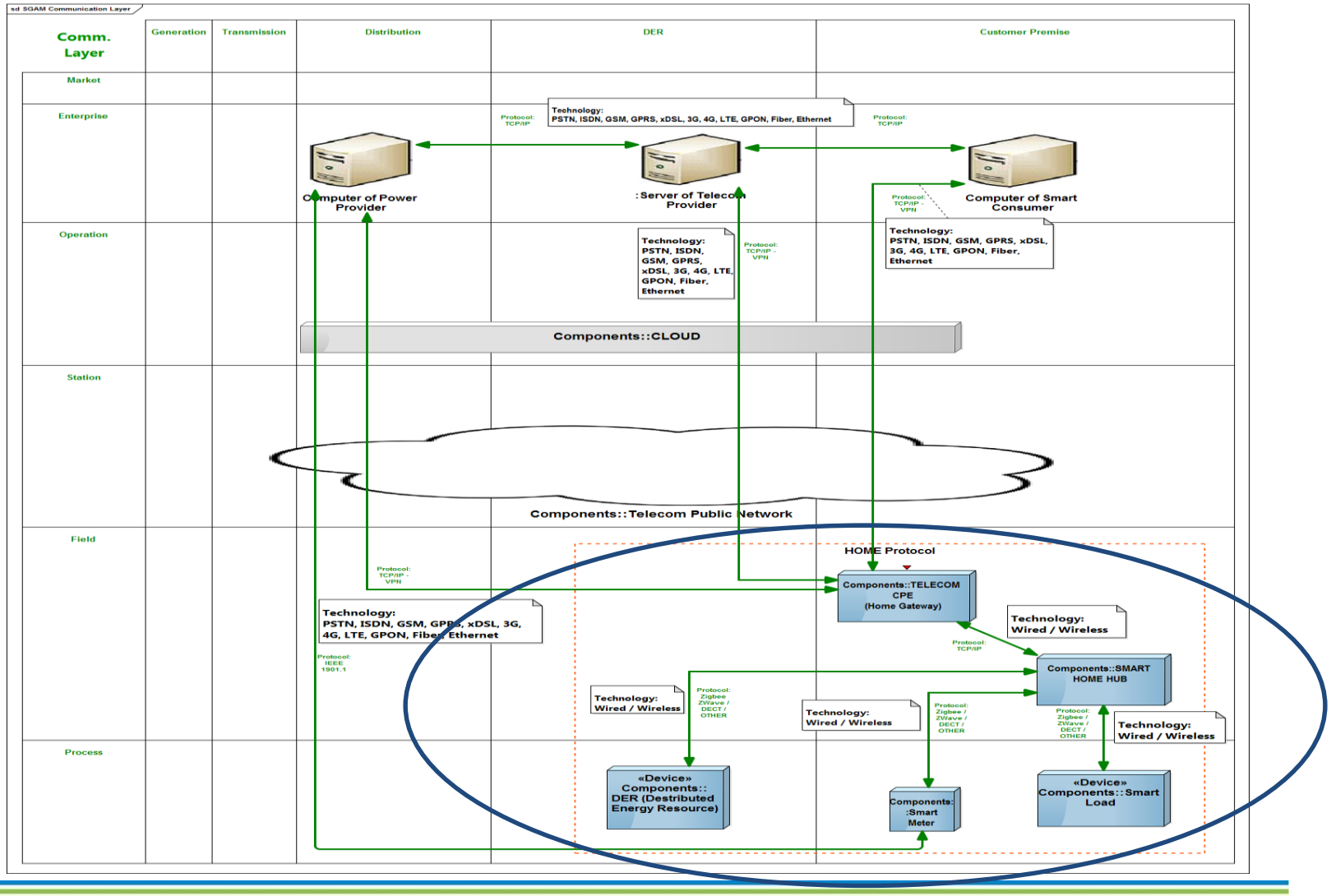
Hellenic Telecommunications Organization S.A. (OTE)

FUJITSU

Fujitsu Laboratories of Europe (FLE)

COSMOTE our world is you

Component Layer (SGAM model) OTE's Proposal in SmarterEMC2

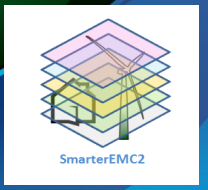


COSMOTE
our world is you

OTE Simulation Environment

- For SmarterEMC2, OTE is developing software tools/ procedures whose objective is to simulate access networks where:
  - Up to tens or hundreds of nodes (i.e. smart meters) are organized in specific topology,
  - Point-to-point, unicast and also broadcast traffic between nodes is being simulated,
  - traditional layer 3 Internet protocols (TCP/IP) with IPv4 and IPv6 address space is used.

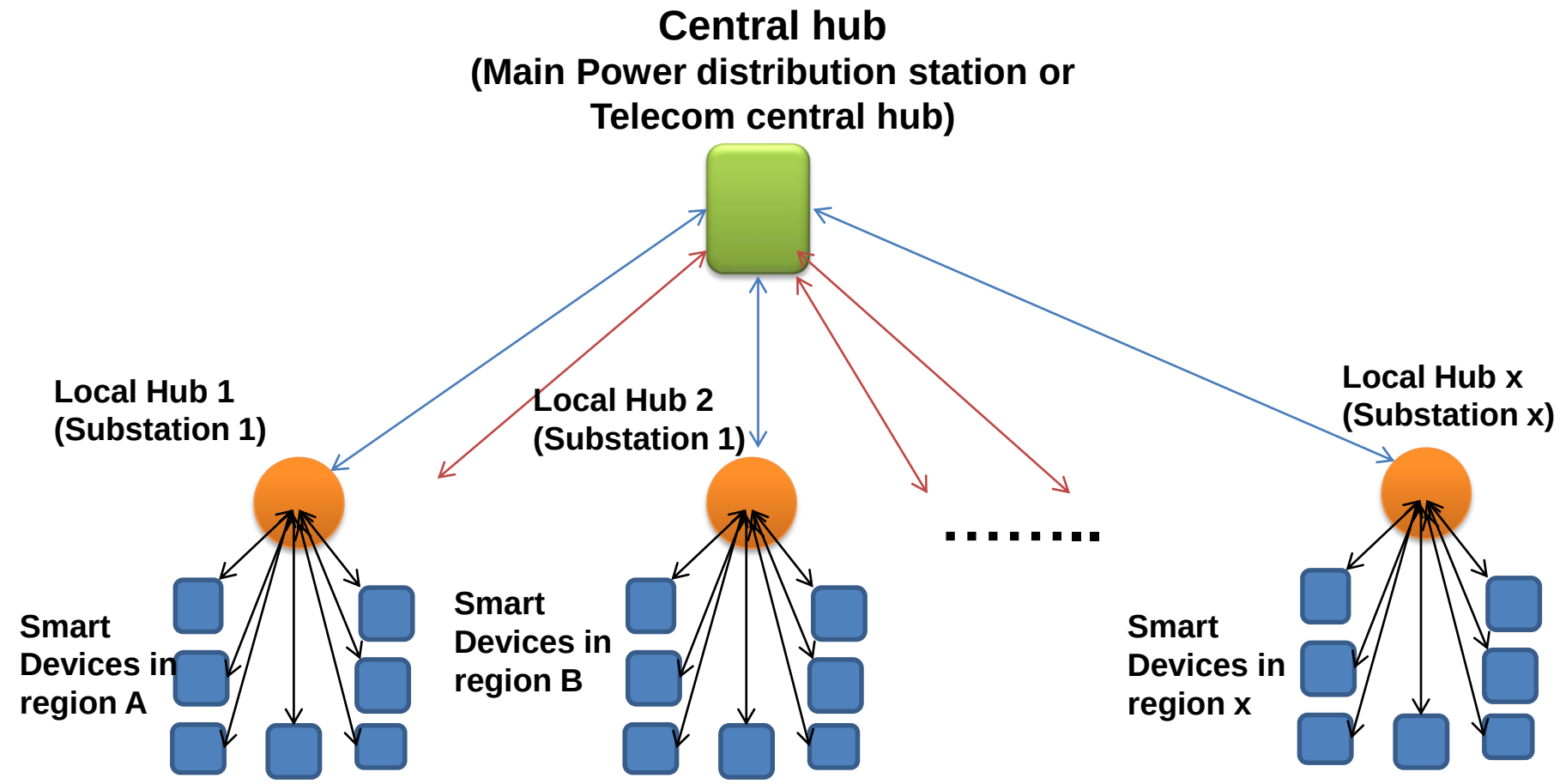
OTE Simulation Environment



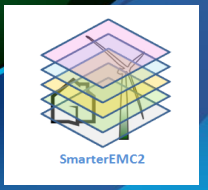
- More specifically OTE's software has the following capabilities:
  - Define Network Architecture and Dimensioning
    - Suggestion of specific overall topology (i.e.: tree, star)
  - Perform Network Traffic Simulation
    - congestion (queues), Latency (delay of packets), noise (errors in packet data)
  - Utilize specific Network Protocols:
    - Development of functions of TCP/IP Listener and Sender.
    - Extension to support UDP can be done upon request.

OTE Simulation Environment proposed topology

– **OTE proposed the following 3-level Tree-star topology:**



OTE Simulation Environment



- **3 level approach consists of :**
  - "**Smart Devices**" in the lower layer
  - "**Local hubs**" in the middle layer
  - "**Central hub**" in the upper layer
- **Tree topology:**
  - The **smart devices** of each region send/receive values to their regional Local Hub (star topology) periodically or on demand.
  - The **Local hubs** broadcast/receive values to/from the Central Hub (star topology) and broadcasts values to their regional smart-meters and gets the replies.
  - The **Central-hub** broadcast measurement commands to the Local Hubs and gets the replies.
  - The smart Devices cannot communicate with other Smart Devices or the central hub directly but only via the local hub they "belong" to.

OTE Simulation Environment

OTE's software simulation environment:

- OTE is developing software for sending/receiving TCP/IP packets, Using **Microsoft Visual Studio VB.NET** (used for advanced simulations). VB.net is the platform proposed for more "realistic" network simulation.

- The final version (not completed) of the simulation will described in the following presentation.

OTE Simulation Environment
Smart Device Module

Microsoft Visual Studio VB.NET (advanced simulation)

In order to simulate the 3-level tree-start topology we have developed the following modules:

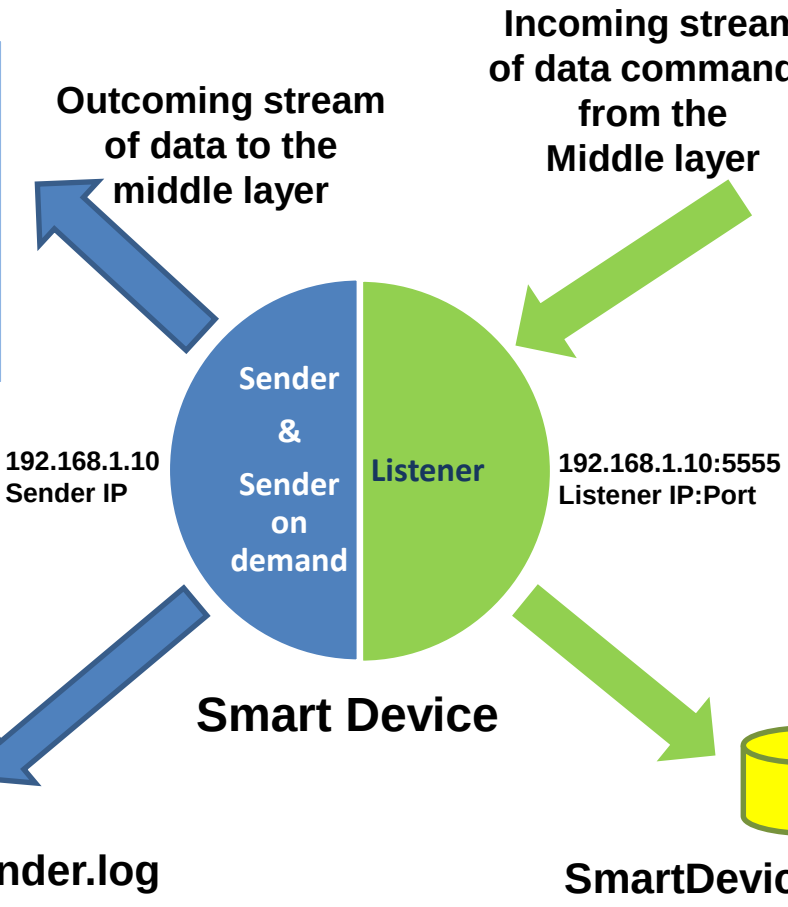
A. Smart Device module (lower layer node)

- This is a command line tool for windows environment.
- Based on TCP/IP communication protocol
- Consists of the following **sub-modules**:
  - **Listener** (listens TCP/IP Packets),
  - **Sender** (sends data to the local hub periodically)
  - **Sender on-demand** (sends response data when a measurement command has been received from the Central Hub)
- Has the following structure:

OTE Simulation Environment
Smart Device Module

Smart Device module structure

Values
- Fix (10.100.1000)
- Random (7,45,265)
- Last stream
- OK (alive)
- #id
- Shutdown



Data commands:
- Send now values
- Send Last stream
- Send fix values
- Send random values
- Send values with id xx
- Send OK (alive)
- Shutdown

OTE Simulation Environment
Smart Device Module

Smart Device module syntax

SmartDeviceModule.exe

TCP/IP SmartDevice module created by OTE Labs - 2016 -2017

Usage: SmartDeviceModule.exe -nm [sd name] -sn [LH name] -ip [LH IP] -p [LH. port] -s [stream] -d delay[0-5] -n noise[0-5] -t [millisecs] -lp [listener port] -rd [millisecs]

-nm	**[sd name]**	**the name of the smart device**
-sn	**[LH name]**	**the name of the Local Hub where the smart device sends data**
-ip	**[x.x.x.x]**	**Local Hub's IP where the smart device sends data**
-p	**[1-65535]**	**Local Hubs's port where the smart device sends data**
-s	**[value]**	**send data values: "fix" sends "10,100,1000" or "random"= send random stream**
-d	**[0-4]**	**delay distribution - under construction**
-n	**[0-4]**	**noise distribution - under construction**
-t	**[millisecs]**	**in every [millisecs] milliseconds period the Smart device sends data**
-lp	**[1-65535]**	**the port where the Smart device listens from a Local Hub**
-rd	**[millisecs]**	**wait randomly [0 to millisecs] before sending data**

OTE Simulation Environment
Smart Device Module

Smart Device module algorithm:

- On starting Smart Device module both sub-modules (Listener and Sender are started)
- Sender sub-module sends data to its Local Hub with two ways:
  - **a)** **Periodically** every **–t milliseconds** period of time
  - **b)** **On demand**, when the Smart Meter listener sub-module receives a **command** from its Local Hub
- SD listener sub-module listens to a specific IP:port and writes a record into the **SmartMeter_Listener.log** file, for every received data stream:

  {SM name, Local Hub ip:port,time,datastream}
- SM Sender module writes a record into the **SmartMeter_Sender.log** file, for each sent data stream:

  {SM name, Local Hub ip:port,time,datastream}

OTE Simulation Environment
Local Hub Module

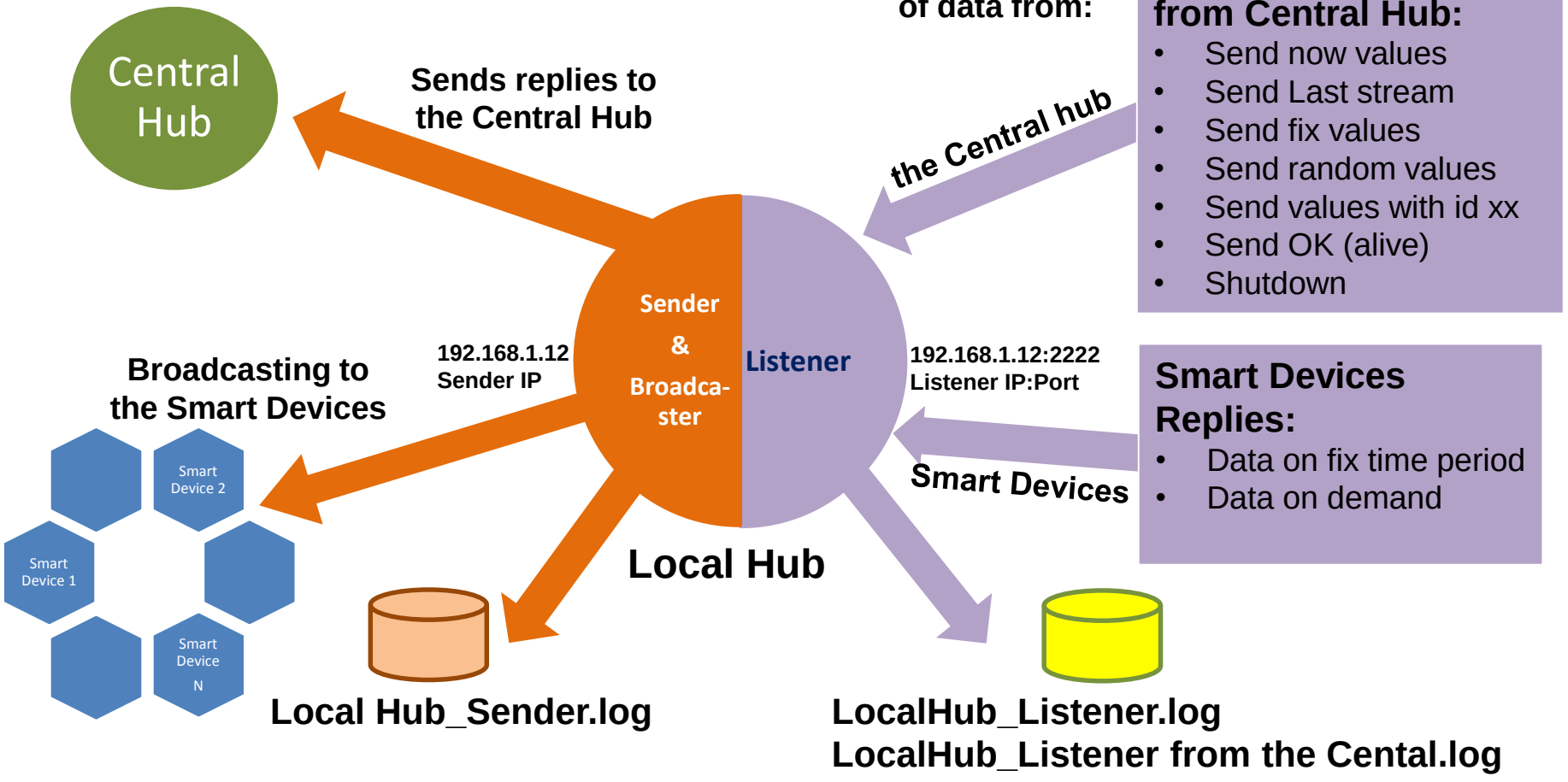
B. Local Hub module (middle layer node)

- It is a command line tool for windows environment.
- Based on TCP/IP communication protocol
- Consists of the submodules:
  - Local Hub **Listener** (for Smart Device and Central-Hub)
  - Local Hub **Broadcaster** to Smart Devices
  - Local Hub **Sender periodically** (collected data)
  - Local Hub **Sender on demand** (replies to Central-Hub)

- Has the following structure:

OTE Simulation Environment
Local Hub Module

Substation module structure



OTE Simulation Environment
Local Hub Module

B. Local Hub module syntax

SubstationModule.exe

TCP/IP LocalHub application created by OTE Labs 2016 -2017

Usage: LocalHubModule.exe -ln [sub id] -ip [centralIp] -p [port1-N] -sip [smart meter ip] -sp [sm port1-M] -d delay[0-5] -n noise[0-5] -lp [sub listener port] -t [milisecs]

-ln	[LH name]	the name of the Local Hub
-cip	[x.x.x.x]	the IP of the Central hub (upper layer)
-cp	[CH port]	Central hub port range where the Local-hub sends data
-sip	[x.x.x.x]	Smart device IP (lower layer)
-sp	[port1-M]	smart device's port range where the Local-Hub sends data e.g. 5000-5100 means 100 smart devices
-d	[0-3]	delay distribution - under construction
-n	[0-3]	noise distribution - under construction
-lp	[1-65535]	the port where the Local-hub listens from a Central-hub or a Smart Device
-t	[millisecs]	the period of time in milliseconds that the Local-hub sends data to Central hub
-rd	[millisecs]	wait randomly [0 to millisecs] before sending data

OTE Simulation Environment
Local Hub Module

Local Hub module algorithm:

- On starting Local hub module both sub-modules (**Listener** and **Sender** are started).
- **Listener** sub-module receives periodically data from **Smart Devices** and writes the data into Substation_Listener.log file

  {SM name, Smart Meter ip:port, time, datastream, time to receive}
- **Local Hub Sender** submodule **periodically** every **–t milliseconds** time, collects any received information from Smart Devices (already logged) and sends data to the Central-hub.
- **Listener** sub-module may also receive **command streams** from the Central-hub (e.g. "**sendnow**") and writes the data to the a log file. Each command stream from the C.H. has a specific **"Token".**
- Then the **Broadcaster module**, broadcasts the "command" with specific token to its own Smart Devices and the Local Hub waits for the replies.

OTE Simulation Environment
Local Hub Module

Local Hub module algorithm:

- Each Smart Device on getting a broadcasted command from the Local Hub with specific token, replies immediately the measurements to the Local Hub including the token.
- **Local Hub sender on-demand** module collects all the replies from the Smart Devices with the same token and sends the reply to the Central-Hub.

- Sender sub-module, writes any data to the **LocalHub_Sender.log** file

  {Local hub name, recipient ip:port, time to send, datastream}

  that have been sent to the Smart Devices or the Central hub.

OTE Simulation Environment Central Hub Module

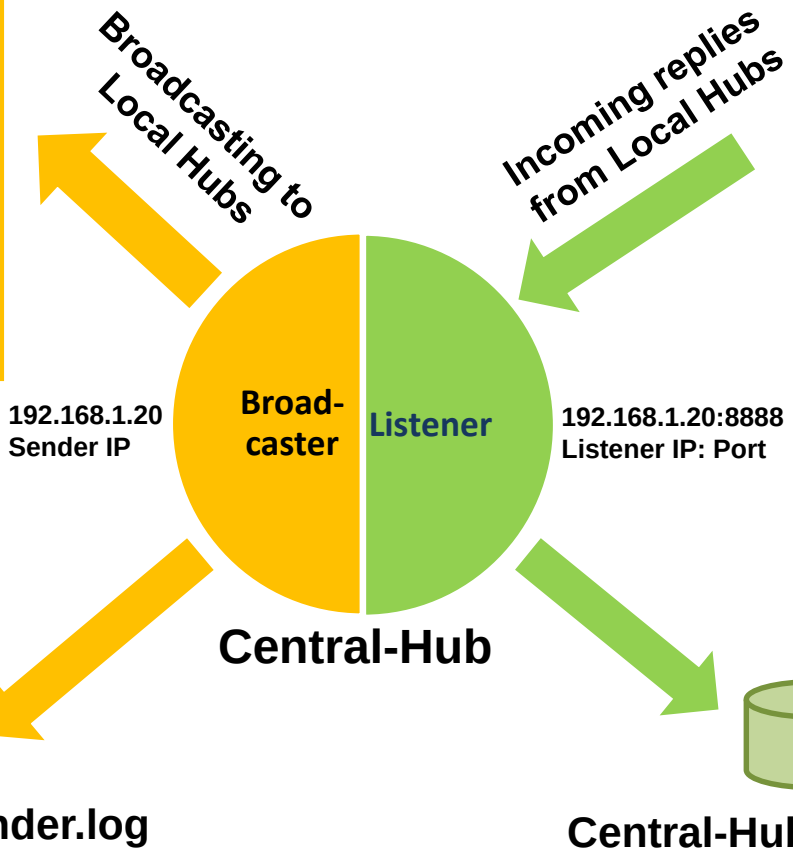
C. Central-hub module (Upper layer node)

- This is a command line tool for windows environment.
- Based on TCP/IP communication protocol
- Consists of the **sub-modules**:
    - Central-hub **Listener**
    - Central-hub **Broadcaster**
- Has the following structure:

OTE Simulation Environment
Central Hub Module

Central-hub module structure

Demand Commands
- Send now values
- Send Last stream
- Send fix values
- Send random values
- Send values with id xx
- Send OK (alive)
- Shutdown



Local Hubs

Replies:
- Array of collected data received from Smart Devices

Triggers the CentralHub to send a Demand

External Sender module

COSMOTE our world is you

OTE Simulation Environment Central Hub Module

C. Central-hub module sysndax

CentralHubModule.exe

TCP/IP Central Hub application created by OTE Labs - V.G. (c) 2016 -2017

Usage: CentralHubModule.exe -cn [CH name] -ip [LH ip] -p [port1-portN] -sip [smart meter ip] -sp [sm port1-M] -d delay[0 -5] -n noise[0-5] -lp [sub listener port] -t [milisecs]

-cn [CH name] the name of the Central Hub

-ip [x.x.x.x] Local Hubs IPs (middle layer)

-p [port1-M] Local Hub's port range where the Central-Hub broadcasts commands

-d [0-3] delay distribution - under construction

-n [0-3] noise distribution - under construction

-cp [1-65535] the port where the Central-hub listens from Local hubs

-t [millisecs] the period of time in milliseconds that the Central-hub broadcasts data to local hub

if -t = 0 then the Central hub will not send commands

if -t > 0 then the Central hub sends randomly or periodically commands every t time

-rd [millisecs] wait randomly [0 to millisecs] before transmitting data

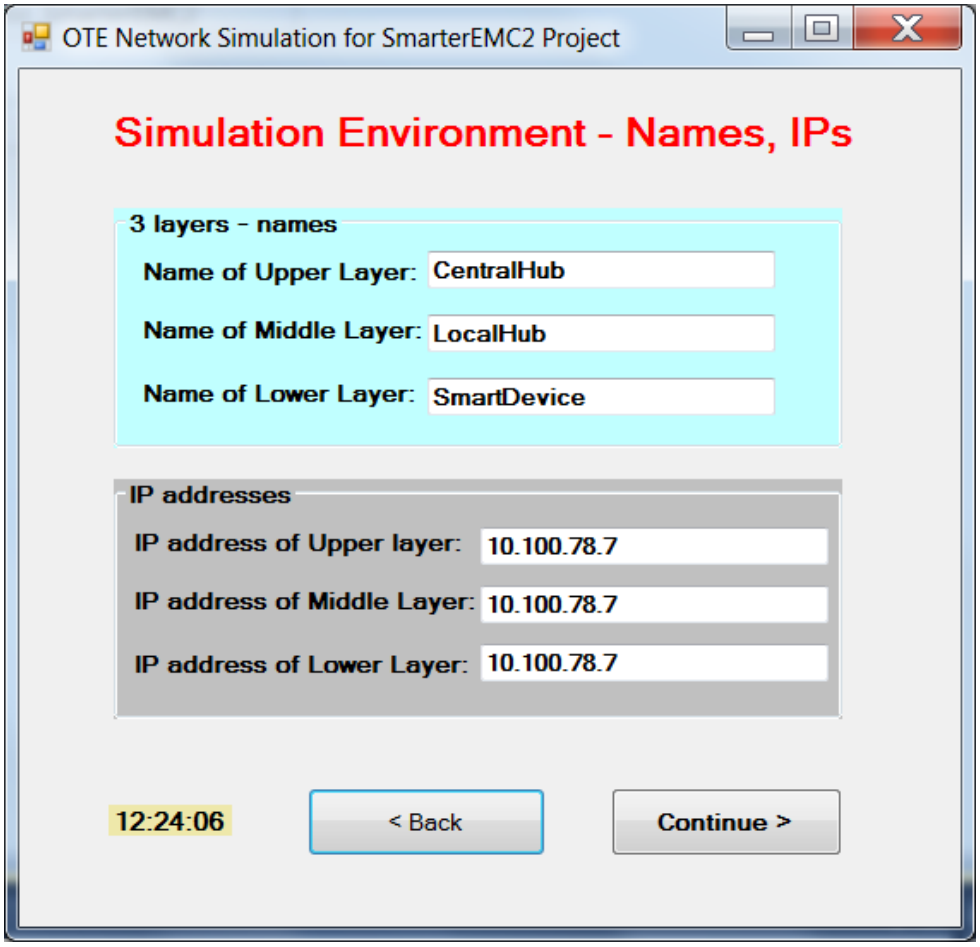
-s [datatypes] datatypes = ("sendnow_periodicly", "sendnow_at_random", "no_send")

OTE Simulation Environment Central Hub Module

Central-hub module algorithm:

- On starting Central-Hub module both sub-modules (Listener and broadcaster are started)
- Broadcaster sub-module broadcasts measurements commands to the Local Hubs with two ways:
  - **a)** **Periodically** every **–t milliseconds** period of time, if this feature has been selected.
  - **b)** **When a triggering command** has been received from the **External Sender module** (it is a simple module just to trigger the Central-Hub to broadcast).
- Every broadcasted command has a **unique token**.
- Central-Hub Listener gets the replies from the Local hubs **grouped by** the unique token number.
- Listener and broadcaster writes data to log files.

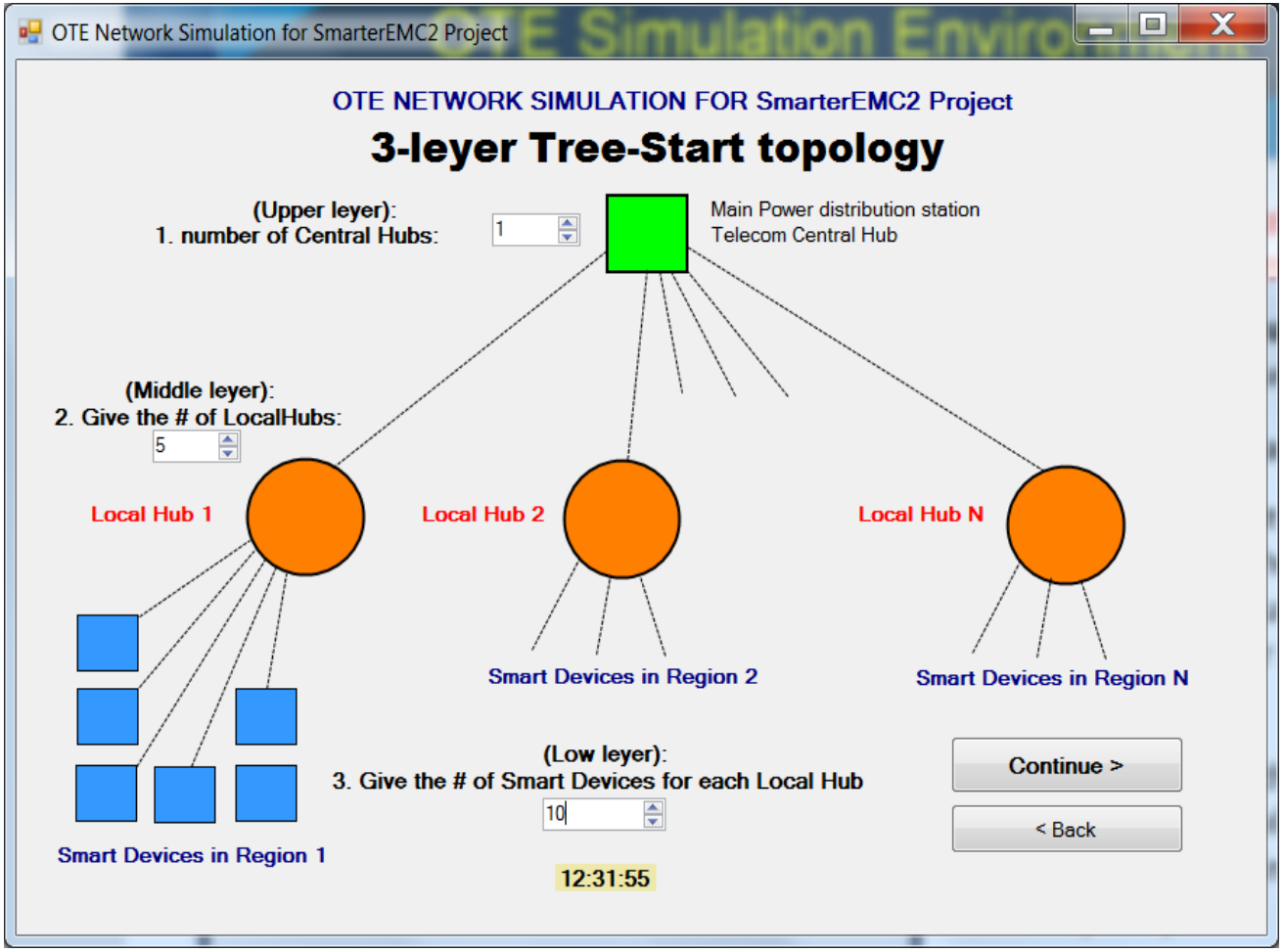
OTE Simulation Environment Console Application



Console Application design Names, Ips Form

- In this form the **names** of each layer node are given.
- Also, the **IPs** of each layer.
- If different IPs are give for different layers, then the application will run in **distributed mode** (each layer in deferent computers).
- If all layers have the same IP, then the application will run in **single mode** (all layers in the same computer)**.**

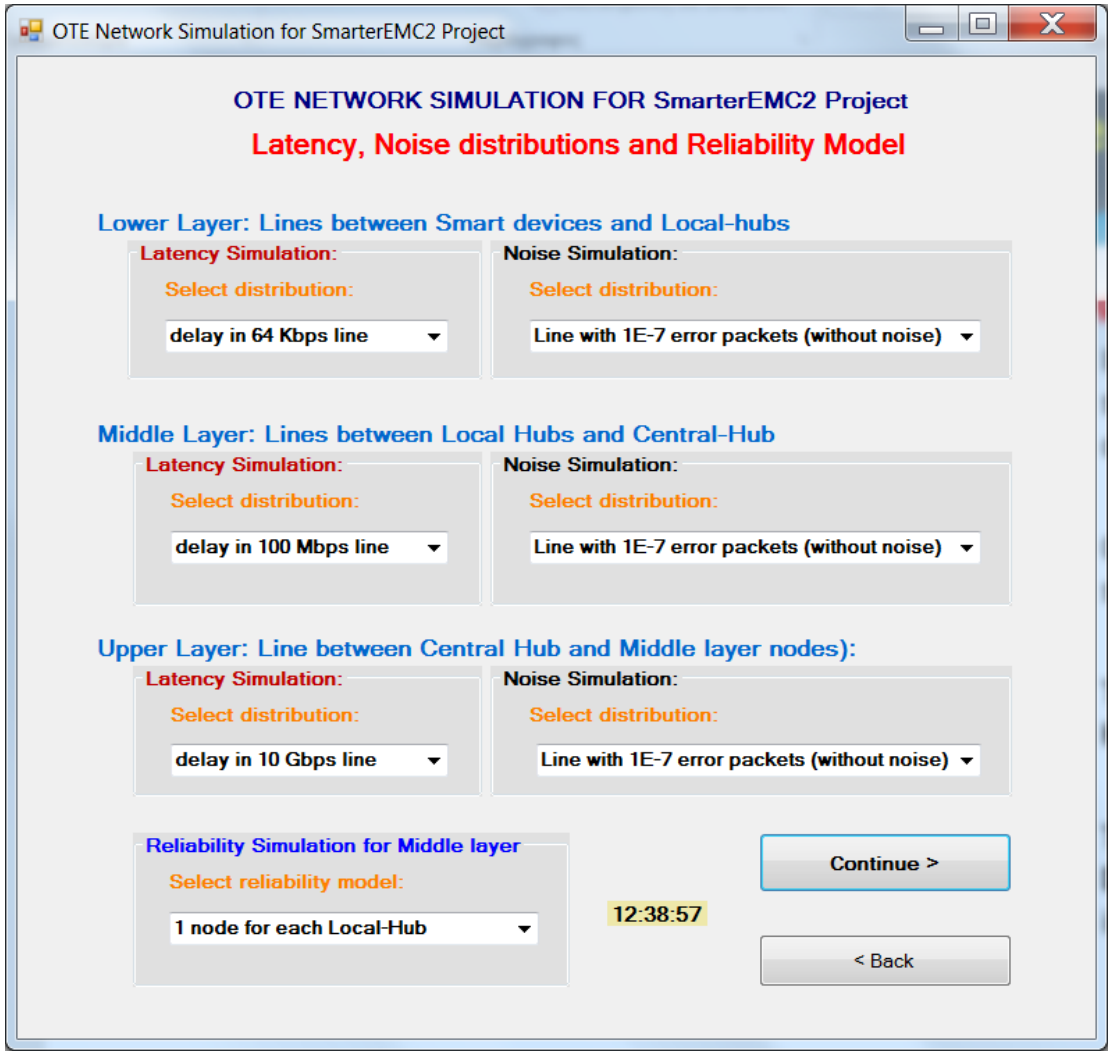
OTE Simulation Environment Console Application



Topology form

- In this form the tree – star topology is designed.
- One **Central-Hub** is selected.
- The number of **Local Hubs** is selected.
- The **number of Smart Devices per Local Hub** is selected.

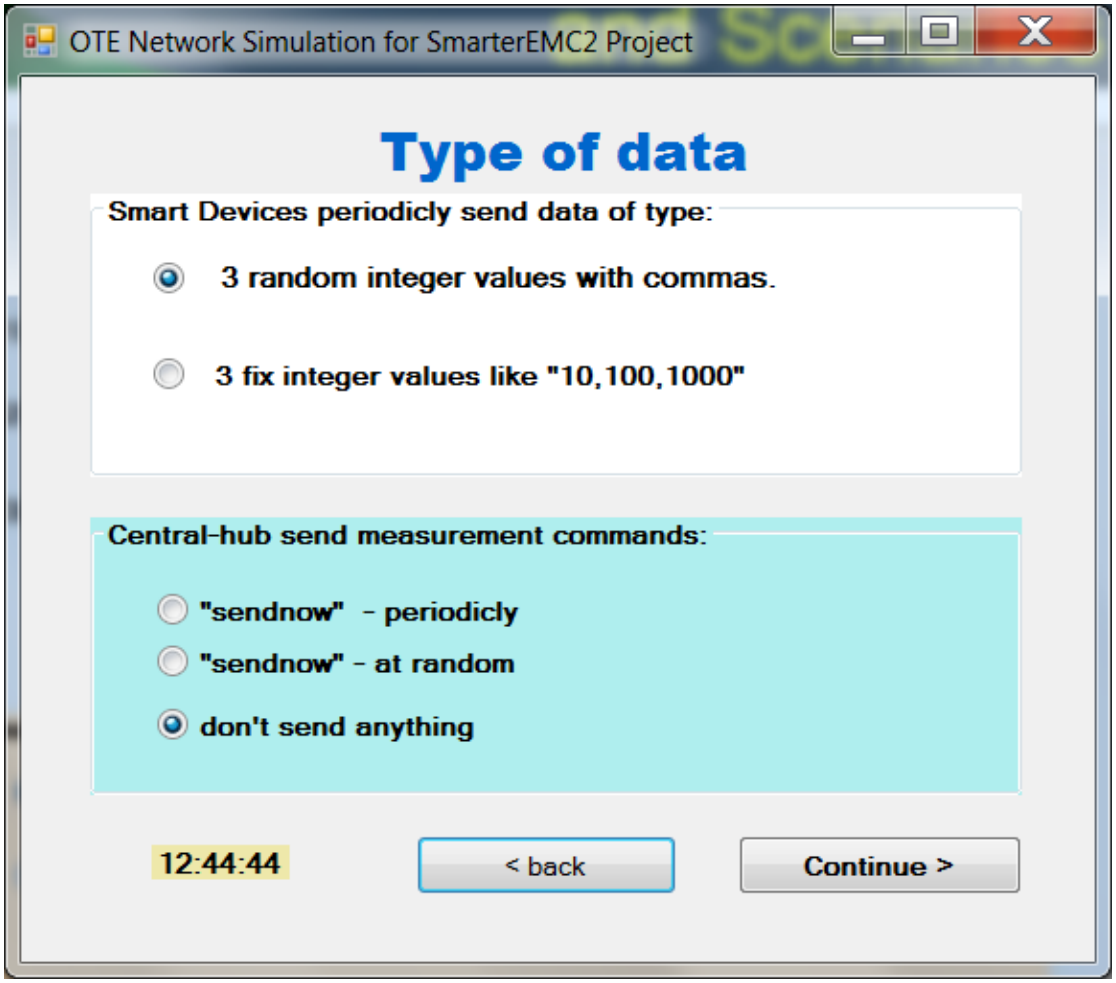
OTE Simulation Environment Console Application



Latency, Noise Distribution Form

- In this form the **Latency**, **noise distribution** is selected.
- Also the **reliability model** for the local hubs.
- For each line we select the level of noise (e.g. 1E-7 error)
- And the delay (e.g. delay in 100 Mbps line).
- Those features are still under developing.

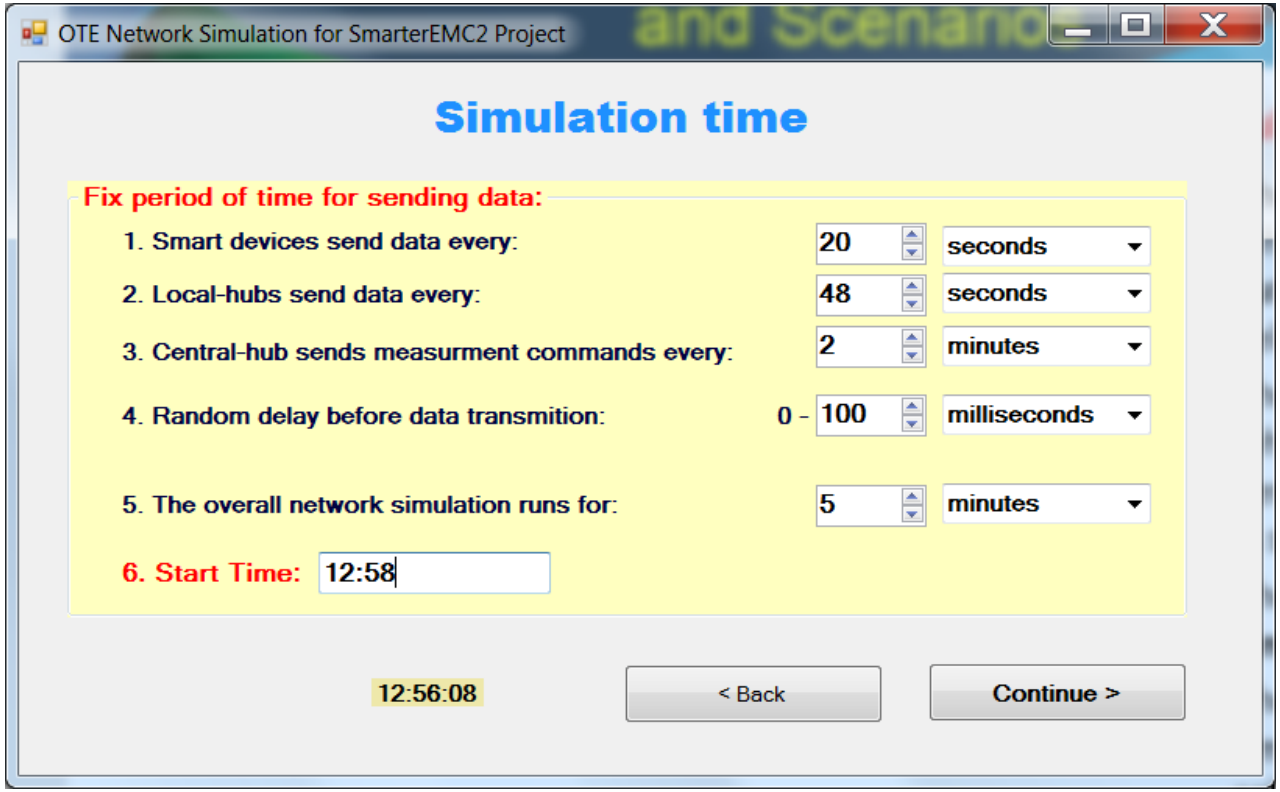
OTE Simulation Environment Console Application



Type of Data Form

- In this form the **type of Smart Devices data** is selected: a) 3 random integer values (e.g. 6,57,456) or; b) 3 fix values “10,100,1000”).
- Also, the **type of the Central-hub measurements commands:** a) sendnow-periodically: the CH sends periodically the command “**sendnow**”.
- b) sendnow-at random: the CH sends at random the command “**sendnow**”.
- c) the CH doesn't send anything periodically.

OTE Simulation Environment Console Application

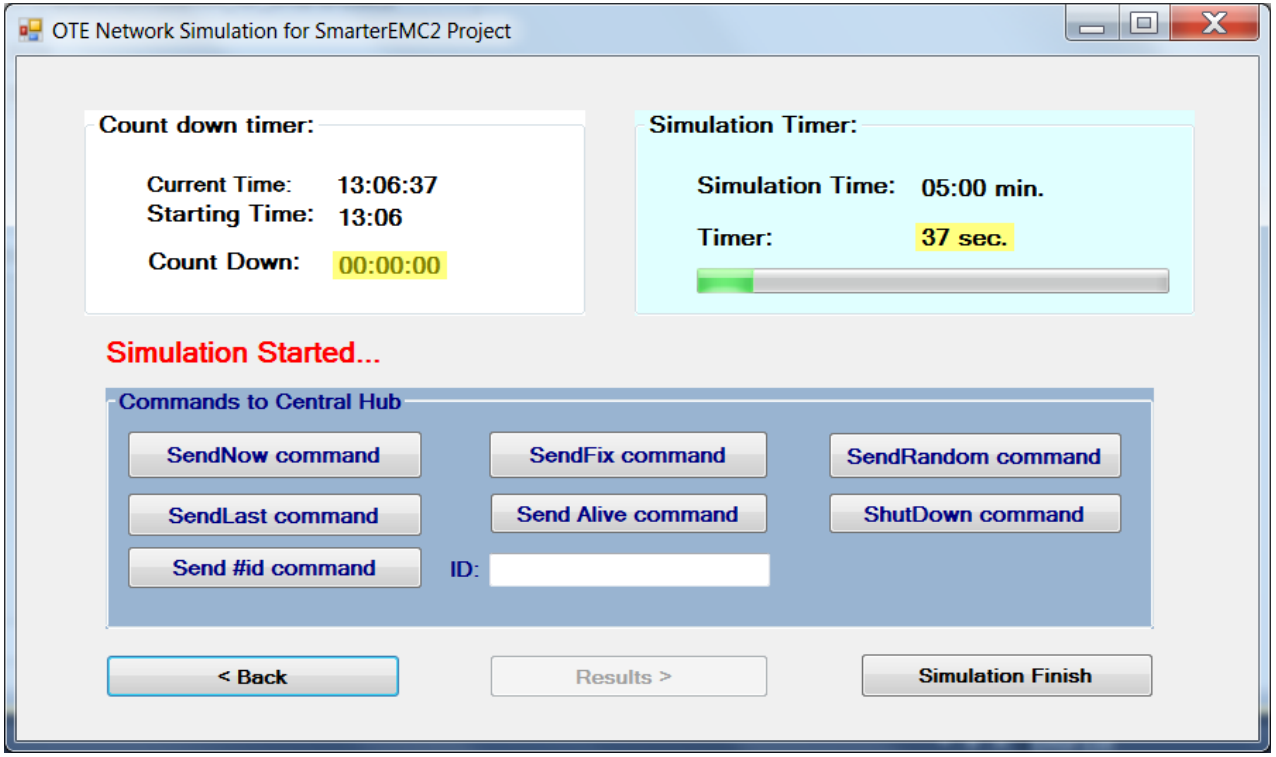


Simulation Time Form

- In this form the **time for the periodical send** of each type of node is selected.
- Smart Devices time
- Local Hubs time
- Central-Hub time (it is enabled only if the sendnow – periodically or sendnow –at-random is selected in the previous form)

- A small random delay time before transmission.
- The overall simulation time
- The starting time.

OTE Simulation Environment Console Application



Simulation Running Form

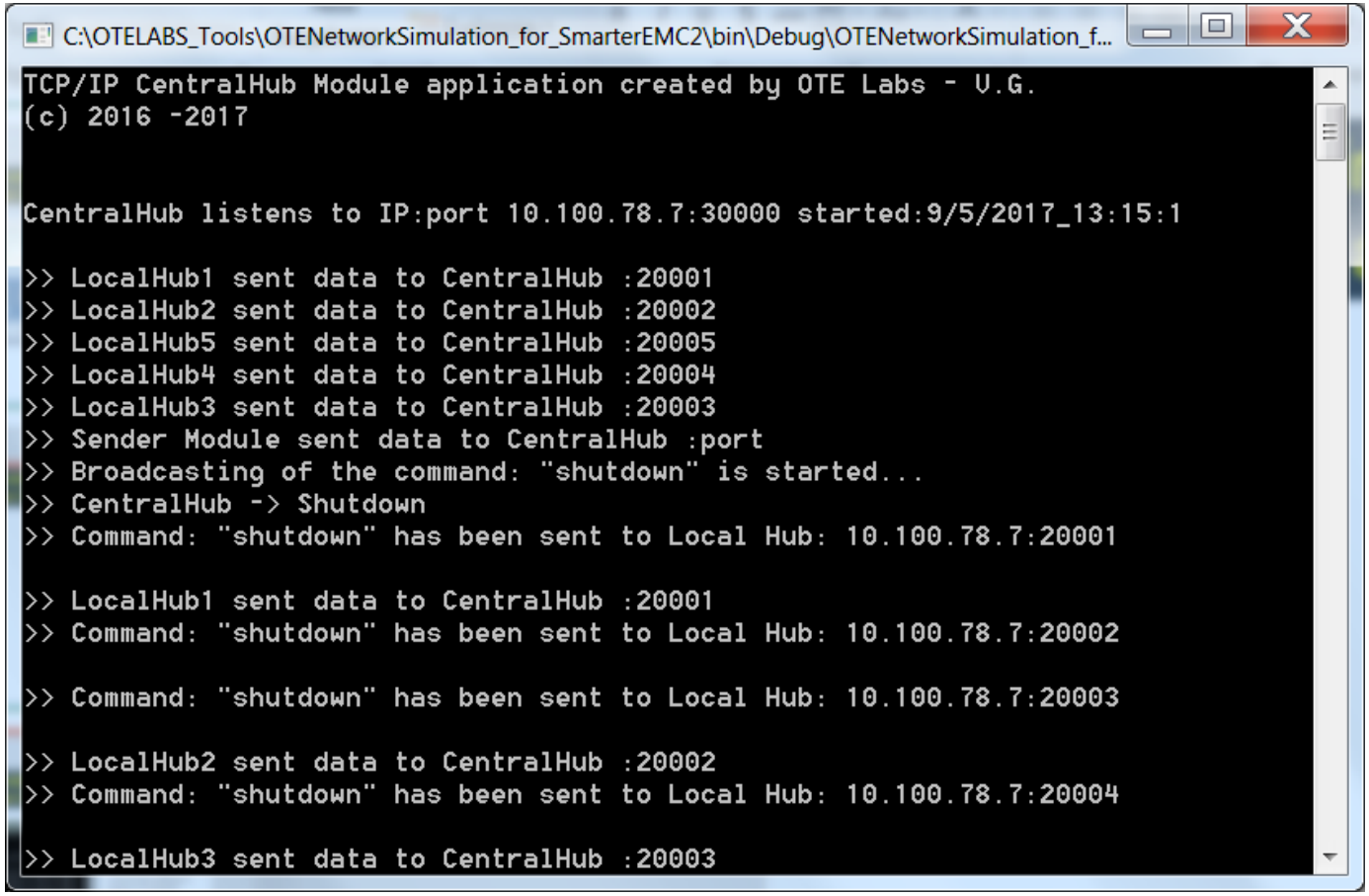
In this form the simulation timers are presented.

Also, we can trigger the Central Hub using the 7 buttons to send the corresponding command.

If the shutdown command is selected all the nodes of the simulation environment are closed.

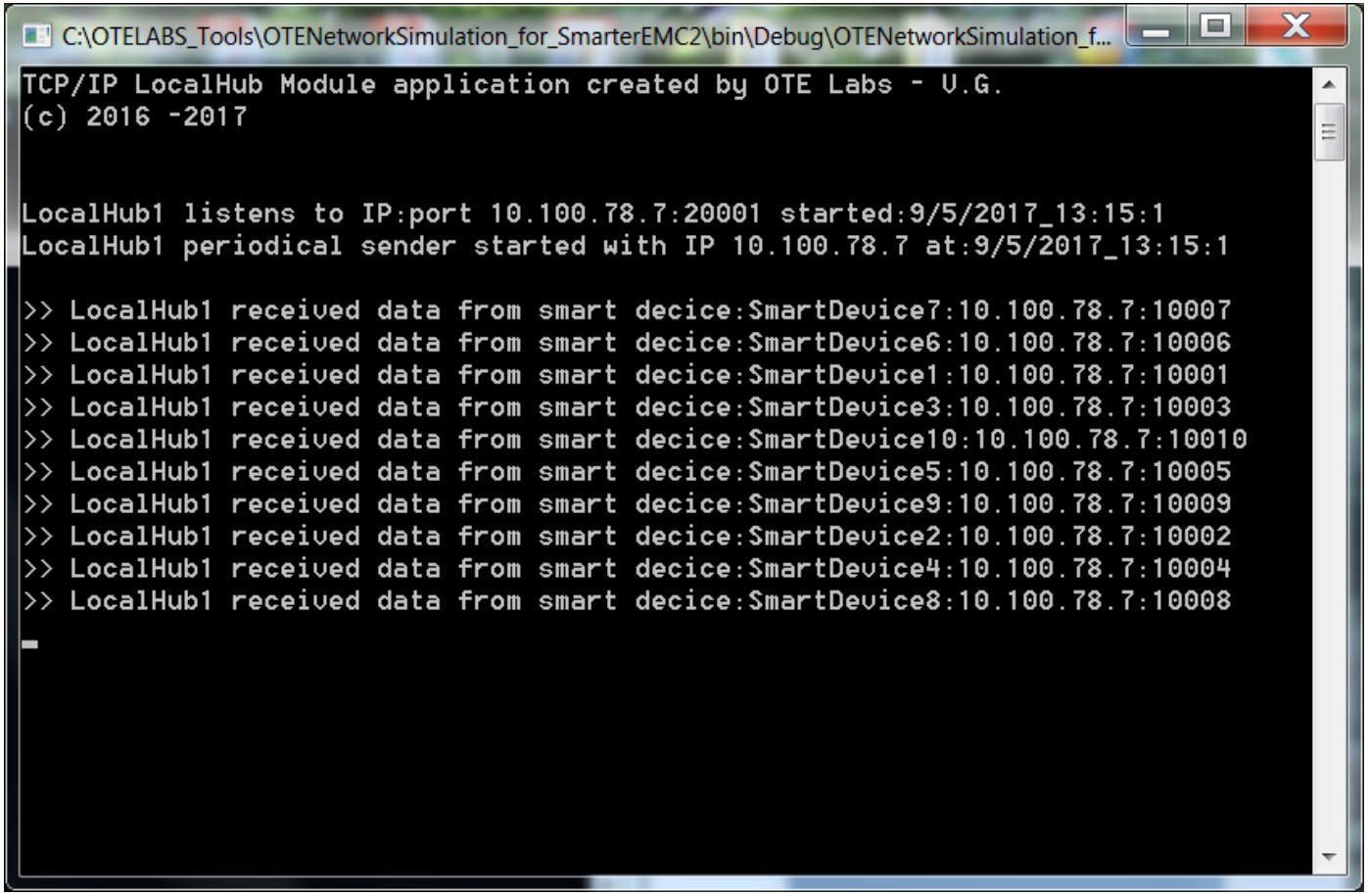
OTE Simulation Environment Simulation Running

- **Central-hub module running**



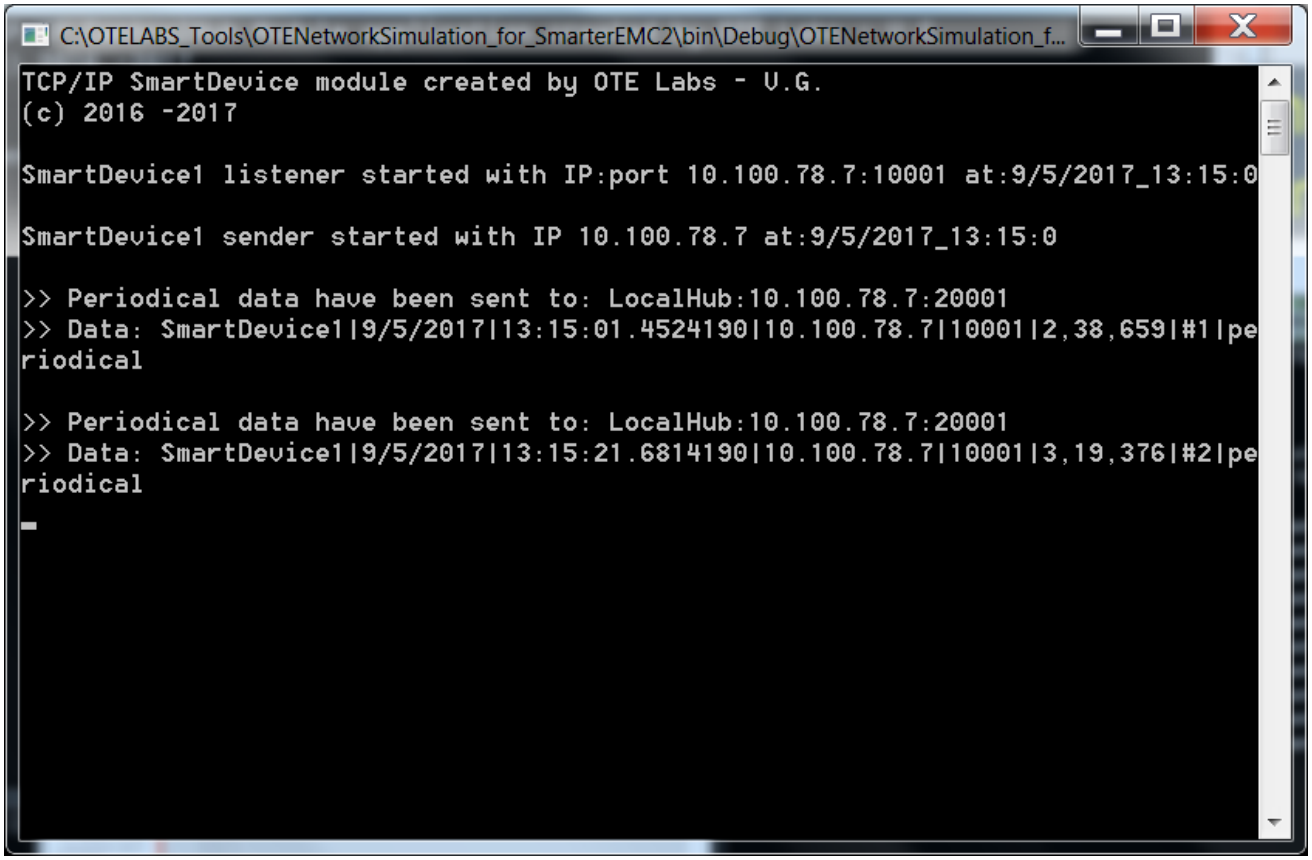
OTE Simulation Environment Simulation Running

- **Local hub module running**



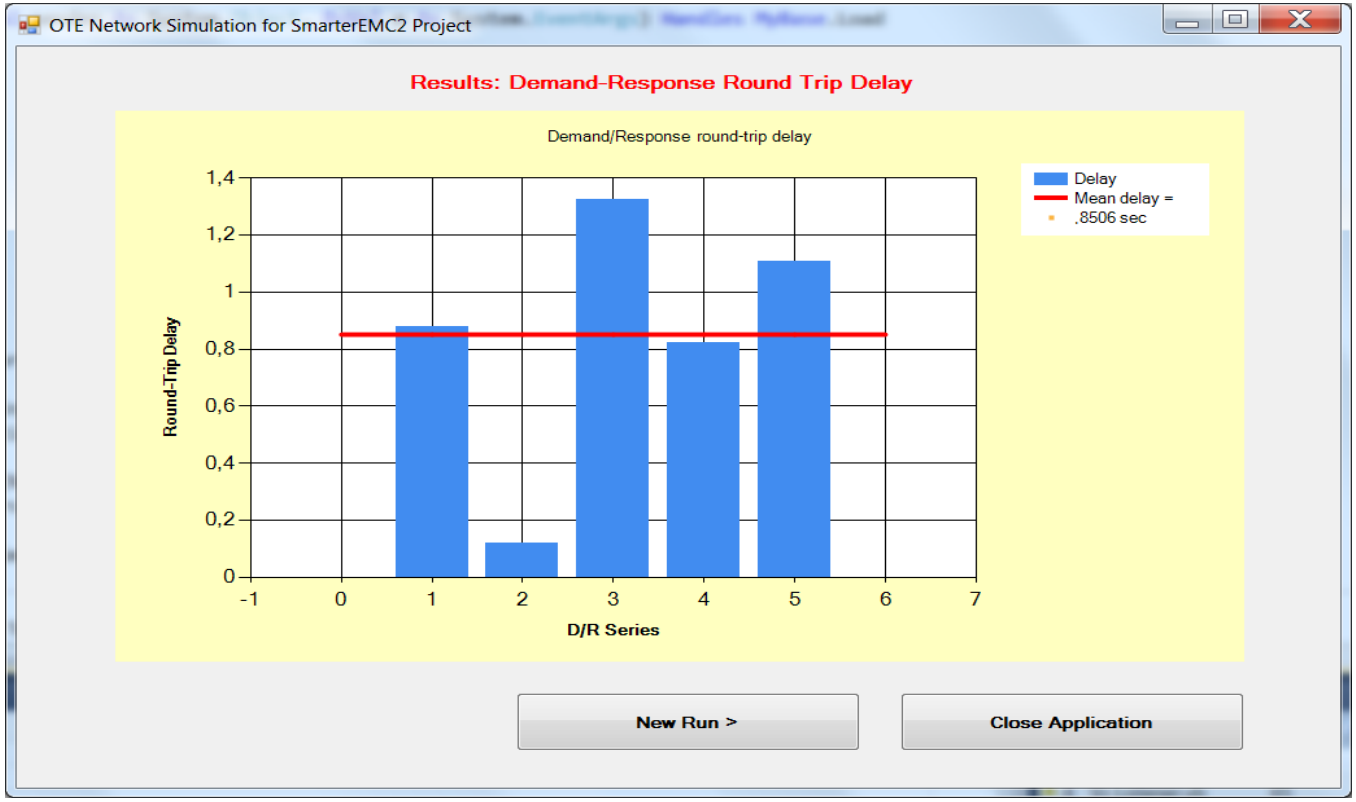
OTE Simulation Environment Simulation Running

- **Smart Device module running**



OTE Simulation Environment Simulation Results

- With programming in the logged data, useful results can be extracted in graphs, Like **Demand-Response Round-Trip Delay (the processing time is included)**



OTE Simulation Environment Simulation Results

The simulation can be run for:

- Different topologies (number of nodes) in the middle and lower layer;
- Different type of data (fix, random values);
- Simulation timers;
- Measurement commands from the Central-hub.

Example: Running for different topologies with the same number of nodes in the lower layer (100 nodes) the Round-Trip Delay can be measured as:

Central-Hub nodes	1	1	1
Local-hub nodes	10	5	2
Smart Devices nodes	10/LH	20/LH	50/LH
Round-Trip Delay	**1,02s**	**1,3 s**	**2,6s**

Thank you!