



RESERVE

D4.4 v1.0

First Interconnection test of the nodes in pan-European simulation platform

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 727481.

Project Name	RESERVE
Contractual Delivery Date:	30.09.2017
Actual Delivery Date:	30.09.2017
Contributors:	Markus Mirz (RWTH), Steffen Vogel (RWTH), Antonello Monti (RWTH)
Workpackage:	WP4 - Pan-European real time simulation infrastructure and live 5G testing platform
Security:	PU
Nature:	R
Version:	1.0
Total number of pages:	24

Abstract

This report describes the experimental results performed by the three partners of the project providing simulation infrastructure: RWTH, POLITO, and UCD. The reports details the measurement of the performance of the laboratory interface as well as the first test of the new simulation platform developed within the project

Keyword list

Laboratory, Infrastructure, pan-European, Deployment, Simulation

Disclaimer

All information provided reflects the status of the RESERVE project at the time of writing and may be subject to change

Executive Summary

One of the objectives of RESERVE is the setup of a pan-European co-simulation that mirrors the development of the real European grid and may serve as a test platform for future scenarios. This distributed co-simulation infrastructure builds upon the components described in deliverables D4.1 and D4.2 and relies on the computational power provided by several laboratories.

In RESERVE, these laboratories are UPB, WIT, RWTH and POLITO. The backbone of the communication between the laboratories is VILLASnode, the software which interconnects simulation sites and is described in detail in D4.1. VILLASnode is already available as open source and has been deployed to dedicated machines in each simulation site.

Network data of the Irish and Romanian grids have shown to be too complex for being simulated monolithically on commercially available simulators like RTDS and OPAL-RT. This deliverable demonstrates a possible approach to this problem by partitioning the network into multiple smaller subsystems. Each of these subsystems is simulated by a dedicated simulation node. The dynamic phasor solver DPsim, described in D4.2 and D4.3, can be used to simulate large networks. However, in comparison to EMT-based simulations, its fidelity is relatively low. This deliverable takes actions required for a co-simulation of existing EMT simulators like OPAL-RT – RTDS with the new DPsim solver in order to combine the strengths of both systems. Each partner offers a different set of simulation possibilities. While DPsim could be used by all simulation sites, POLITO and RWTH offer commercial real-time simulators, WIT offers a cloud infrastructure and UPB has measurements which could be streamed into the simulation.

Before simulating large-scale scenarios, it is advantageous to characterize the quality of the connection between the simulation sites. This facilitates the estimation of the impact of the communication between simulators on the co-simulation results. Therefore, this deliverable includes an analysis of the co-simulation network, which depicts the expected latencies and where they may come from. It is shown that the latencies are in the expected range of tens of milliseconds, 20 to 60 ms depending on the partners involved.

The second part covers the available simulators, how they could be interconnected and a first simulation example between two different sites. Furthermore, it is described how we plan to proceed from the first simulation example towards the simulation scenarios defined in D1.1. The first step is to simulate the IEEE 9-bus grid of D2.1 which is used to study the frequency scenarios Sf_A and Sf_B. The generated simulation results may serve as input for WP2 to improve their proposals for new network codes.

Authors

Partner	Name	Phone / e-mail
RWTH		
	Steffen Vogel	Phone: +49 241 80 49755 E-Mail: stvogel@eonerc.rwth-aachen.de
	Markus Mirz	Phone: +49 241 80 49739 E-Mail: mmirz@eonerc.rwth-aachen.de
	Antonello Monti	Phone: +49 241 80 49700 e-mail: amonti@eonerc.rwth-aachen.de

Project Internal Reviewers

Partner	Name
ESB	Ronan Murphy
TransE	Anca Antemir
RWTH	Marco Cupelli

Table of Contents

1. Introduction	5
1.1 Task 4.3	5
1.2 Objectives of the Work Report in this Deliverable	5
1.3 Outline of the Deliverable.....	5
1.4 How to Read this Document	5
2. Network connectivity	7
2.1 Analysis.....	7
2.2 Virtual Private Network	9
2.3 Summary.....	10
3. Sites	11
3.1 Central server: GCP.....	12
3.2 Laboratories: RWTH, UPB, WIT, POLITO.....	12
4. First Test Simulation Scenario	13
4.1 Co-Simulation without Network Latency	13
4.2 Co-Simulation including Network Emulation.....	14
4.3 Distributed Co-Simulation	14
4.4 Summary.....	15
5. Planned Simulation Scenario.....	16
6. Conclusion	17
7. References.....	18
8. List of Abbreviations	19
9. List of Figures	20
10. List of Tables.....	21
Annex	22
A.1 Laboratory inventory	22
A.1.1 GCP.....	22
A.1.2 RWTH.....	22
A.1.3 UPB	22
A.1.4 WIT.....	22
A.1.5 POLITO	23
A.2 OpenStack Cloud environment.....	23
A.3 VILLASnode installation.....	23

1. Introduction

Renewables in a Stable Electric Grid (RESERVE) is a three-year European Commission funded project within the Work Program H2020-LCE-2016-2017. The project officially started in October 2016.

1.1 Task 4.3

This deliverable is the first major output of task 4.3 in WP4. This task is about the first interconnection tests conducted between the laboratories of the project partners in RESERVE. The interconnection tests include measurements regarding the network connectivity between the laboratory sites as well as distributed co-simulation tests.

1.2 Objectives of the Work Report in this Deliverable

The main objectives of this report are the presentation of the laboratory sites interconnected in the frame of RESERVE, the network connectivity measurements conducted between the laboratories and the results obtained from the first distributed co-simulation tests. Furthermore, the results of the network connectivity tests and the co-simulation tests are linked to show the coherency between the network properties and the effects seen in the simulation. Besides, the next planned simulation tests conducted in task 4.3 are described since this is the only deliverable from this task.

1.3 Outline of the Deliverable

The pan-European co-simulation developed in RESERVE involves four laboratory sites:

- ACS: Institute for Automation of Complex Power Systems RWTH Aachen University, Germany
- WIT: Telecommunications Software & Systems Group Waterford Institute of Technology, Ireland
- UPB: University Politehnica of Bucharest, Romania
- POLITO: University Politecnico di Torino, Italy

After the development of the laboratory interface components described in D4.1 has been largely finished, VILLASnode has been deployed to all four laboratory sites. A crucial step before running any simulations is the characterization of the connection between the laboratories which is presented in the first part of this deliverable. This allows predicting the impact of the communication on the simulation results, which helps to verify the validity of the results.

Furthermore, this deliverable introduces the available simulators and possible combinations for co-simulation scenarios. The current network of laboratories features commercial real-time simulators, like RTDS and OPAL-RT, the open source simulator DPsim described in D4.2, a cloud infrastructure to support simulations as well as network data, which can be streamed live and used as simulation input. This very heterogeneous setup provides the ground for a multitude of different simulation scenarios.

1.4 How to Read this Document

This document can be read on its own, but should the reader want to learn about the components that interconnect real-time simulators such as the simulator described in this deliverable, we suggest reading deliverable D4.1. Overall, this deliverable (D4.2) is related to the following document from the RESERVE project:

- D4.1 – Demonstration of prototype of laboratory infrastructure
- D4.2 – Functionality of the releases of the real-time solver, V1
- D4.5 – 5G extended functionality development to support voltage and frequency control

D4.1 offers insight into the laboratory interconnection infrastructure, which is the framework that is connecting instances real-time simulation laboratories. Regarding DPsim, the real-time simulator used for the connection tests, the interested reader can learn more by reading D4.2 and later on D4.3. This deliverable (D4.4) is input to deliverable D4.5 since it describes already the laboratory setup in which the base station will be integrated. The chart below provides a graphical representation of the dependencies between the deliverables in WP4 of RESERVE.

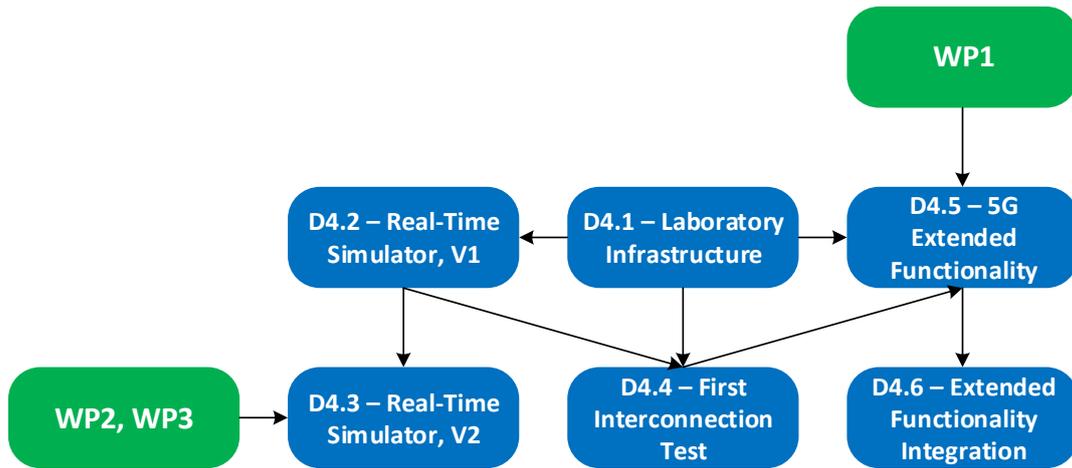


Figure 1: Relations between deliverables in WP4 and other work

2. Network connectivity

As motivated in D4.1, the Internet provides the transport medium to connect laboratories in RESERVE's pan-European laboratory infrastructure. The Internet is comprised of commercial and National Research and Education Networks (NREN). Due to their academic origin, all sites have direct connectivity to their respective NRENs (Table 1). NRENs in Europe are interconnected by GÉANT, a pan-European data network for the research and education community founded by the European Commission.

Table 1: Nation Research and Education network providers.

Site	NREN	Country	Website
ACS	Deutsches Forschungsnetz (DFN)	Germany	https://www.dfn.de
WIT	HEAnet	Ireland	https://www.heanet.ie
POLITO	Consortium GARR	Italy	https://www.garr.it
UPB	RoEduNet	Romania	https://www.roedu.net
all	GEANT	Europe	https://www.geant.org

2.1 Analysis

Figure 2 shows an approximation of the geographical routing paths that have been determined by a traceroute tool. The actual path is given by the network operators and can change depending on the available connection paths between them. As such, it is impossible for the sender or receiver to control this path. The figure shows, that for most connections the routing is close to the optimal airline distance. Only the path between ACS and WIT (black line) is taking a detour over Berlin and Hamburg, which results in an increased latency. Older tests showed that this detour could be eliminated by artificially splitting the path selection into two parts. By adding a third hop between WIT and ACS, e.g. in Amsterdam, we end up with two separate routes which, depending on the location of the introduced hop, can reduce the overall latency. Selecting the optimal location for such intermediate hops is difficult and requires deeper knowledge about the network topology and interconnections. Based on our experience, the latency added by each additional hop is marginal in comparison to the latency caused by geographical distance.

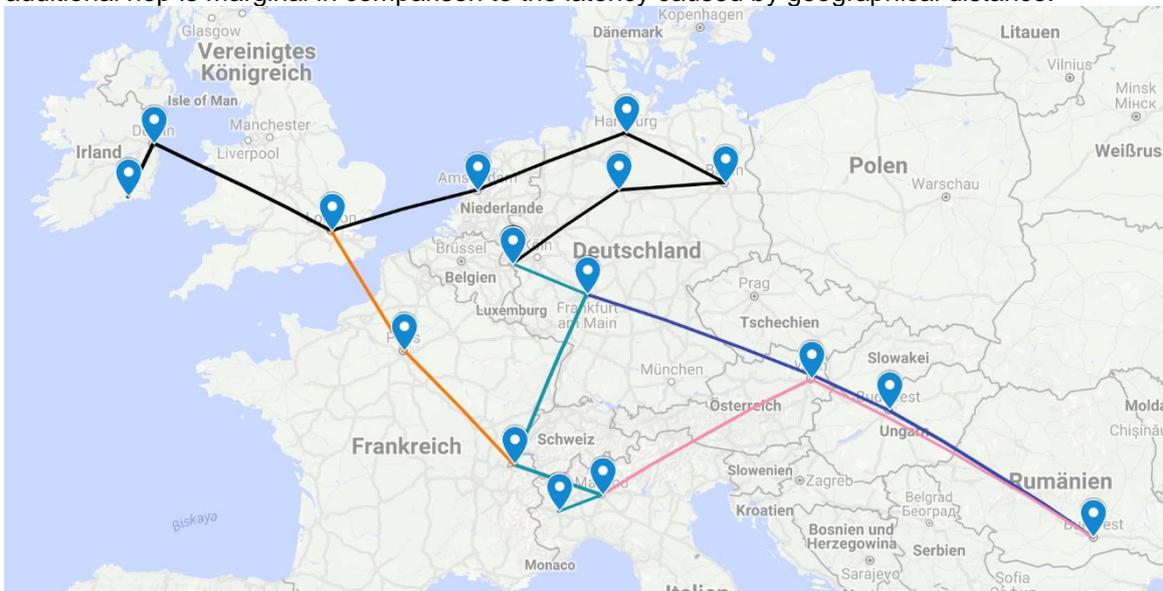


Figure 2: Routing Paths between ACS, POLITO, UPD and WIT.

Figure 3 shows the cumulative composition of the round-trip time (RTT) between ACS and WIT as seen from the intermediate routers. The data presented in Figure 2 has been collected using MTR an advanced traceroute tool ¹. This figure clearly shows that the latency has a mostly linear dependency with the geographical distance.

¹ <http://www.bitwizard.nl/mtr/>

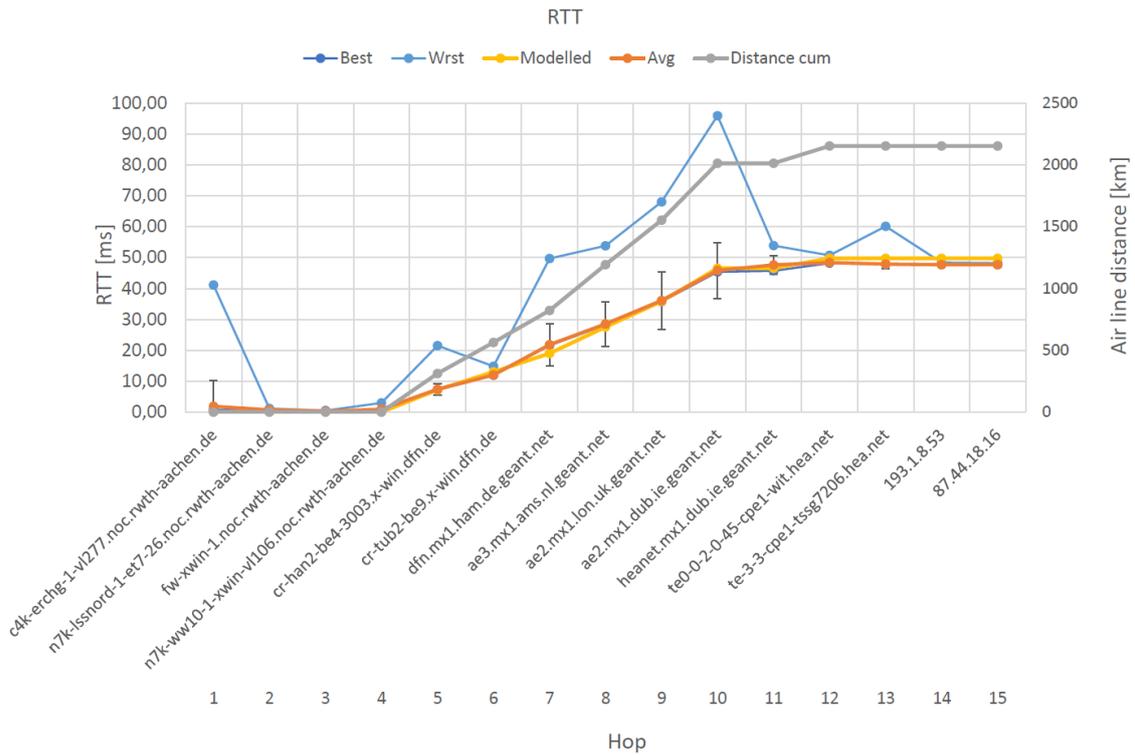


Figure 3: Network latency of ACS-WIT connection.

Figure 4 and Table 2 show the full matrix of RTTs between all nodes. By using this matrix, we can optimally map the simulation topology onto the network topology.

[ms]	UPB	POLITO	WIT
ACS	31,9	23,5	47,7
WIT	59,6	38,1	
POLITO	41,0		

Table 2: Round-trip time matrix.

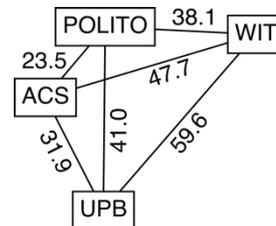


Figure 4: Round-trip time graph.

The RTT is only one, but important, metric for evaluating the quality of a connection for real-time simulation. Another metric of interest is the maximum packet rate, which we can use to exchange simulation samples while keeping the packet loss below a certain threshold. For this test, VILLASnode is used to generate sample streams at different packet rates and number of values.

Figure 5 shows the cumulative distribution of the RTT between ACS and WIT in dependency to the sending rate. Not shown in this figure is the loss and reordering of packets, which gets significant with higher rates. At higher rates, the connection is susceptible to third party network traffic which causes loss and reordering.

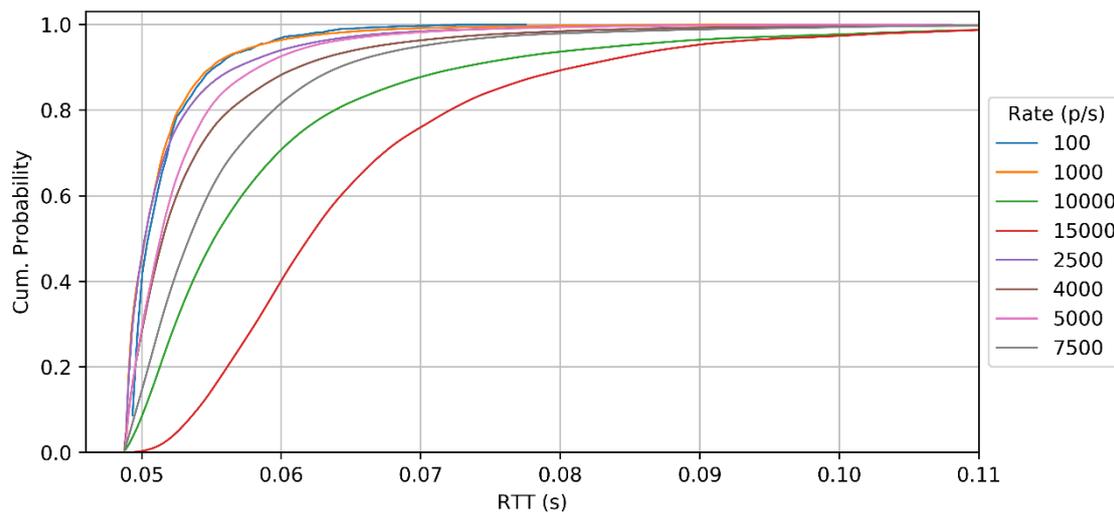


Figure 5: Cumulative probability of RTT.

2.2 Virtual Private Network

Due to its public accessibility, further measures to secure the network against eavesdropping and spoofing are required.

We have chosen to use the open source VPN software Tinc to establish a virtual private overlay network between all sites. Tinc is special because it does not follow the usual client / server architecture but implements a fully decentralized network in which each node has the same role. Tinc tries to establish as many point-to-point connections between the nodes in a network as possible. Traffic will be rerouted over a third node whenever this is not possible due to firewall restrictions. Our setup consists of five VPN nodes running the Tinc VPN software as well as other components of the VILLASframework. Except for the public hosted Google Cloud Platform (GCP) node, we must assume that firewalls restrict the access to laboratory nodes, which prevents direct connections between them. However, these direct connections are an essential requirement for conducting real-time simulations.

We considered two approaches to solve this issue:

- Firewall restrictions can be lifted by adding exceptions that allow traffic for certain source / destinations and protocols. As this approach requires involvement of the IT administrators of the respective sites, it usually requires some lead time.
- The public Tinc node (GCP) can act as a facilitator and use techniques commonly referred to as NAT hole-punching to enable a direct connection between the isolated nodes [1], [2]. However, the success of this approach depends on the firewall implementation. Some firewalls have shown to be immune against those hole punching techniques.

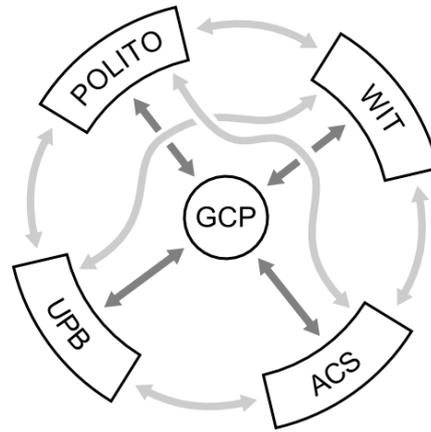


Figure 6: A fully meshed peer-to-peer VPN network.

2.3 Summary

This chapter presents the decentral VPN solution *Tinc* which is used to build a scalable overlay network. It is used to establish connectivity within RESERVE's pan-European laboratory infrastructure. First measurements have been collected to estimate the impact of the communication network onto the distributed co-simulation. The measurements provided a deeper insight into the causes of inherent communication latencies and possible measures to improve them. In the following experiments, the measurements will be used to map co-simulations in an optimal way to the simulation resources.

3. Sites

Figure 7 depicts the simulation components that could be possibly used in each location of the pan-European laboratory infrastructure. VILLASnode instances have been deployed to all sites in order to interconnect the simulators. They represent the edges of this network of simulation laboratories.

The simulators connected to the VILLASnode vary depending on the simulators that are available in the laboratories. In general, the open source dynamic phasor solver (DPsim) [3], which is further described in D4.2, can be used by every laboratory that provides reasonable computation power since it is available to all project partners. OPAL-RT simulators are available in RWTH and POLITO whereas RTDS is only available in the RWTH laboratory. Furthermore, UPB has operational grid data that could be used to represent the external connections of the simulated grid.

In general, the idea is that the EMT real-time simulators, RTDS and OPAL-RT, and the dynamic phasor solver, DPsim, complement each other. EMT simulations can support the investigation of fast events in small sections of the grid since their simulation approach is very detailed. DPsim, on the other hand, simulates in the phasor domain and includes electrical dynamics which makes it more efficient and less accurate than EMT simulation but more accurate than classic phasor simulation. This approach should allow the dynamic simulation of larger grids.

In addition, Google's Cloud Platform (GCP) hosts a central VILLASnode instance on a public cloud server. Wherever provided, the system clocks of all nodes are synchronized to local NTP timeservers. This is a requirement for proper alignment of collected results during post-processing.

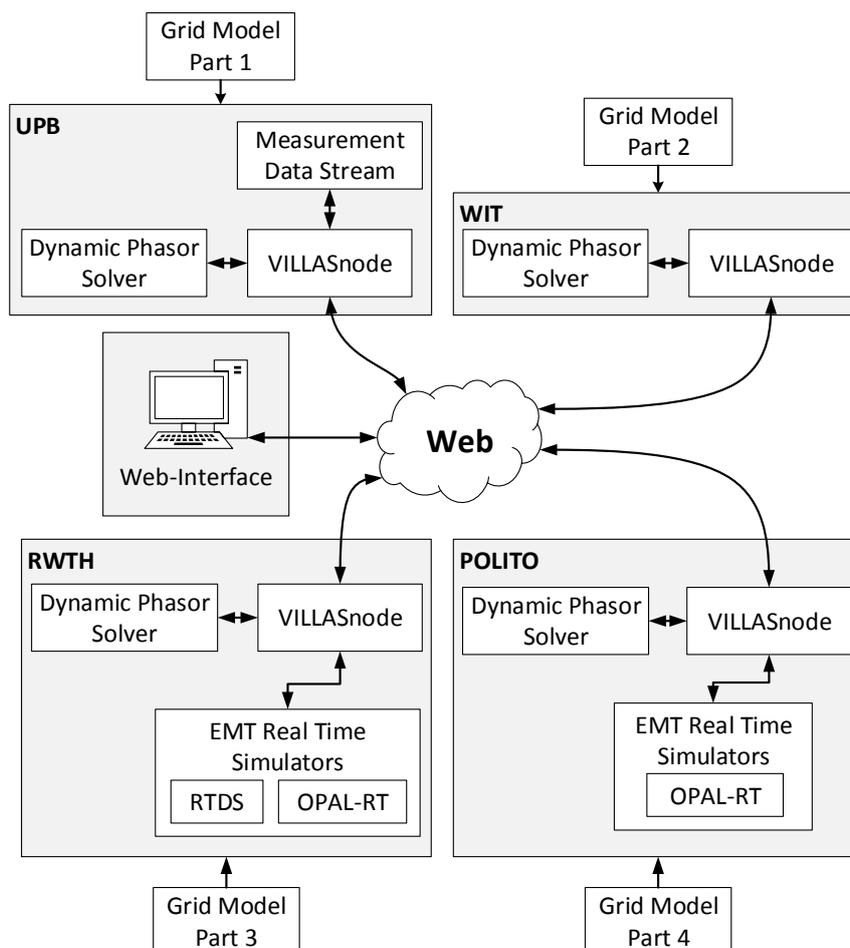


Figure 7: Simulation components available in the pan-European co-simulation

3.1 Central server: GCP

This central instance aggregates live data from the laboratories used for live monitoring over the web interface. For this purpose, a VILLASnode gateway is configured to merge data streams from each of the laboratory sites. However, central node does not handle any real-time critical data. The data streams for monitoring have a different characteristic. Usually they transfer sample data at a lower rate around 10 – 50 *smps / s* and contain more measurement points than the real-time critical interface quantities.

The public GCP node also hosts the RabbitMQ message broker. In future, it will be used to orchestrate the simulators and VILLASnode gateways. As such, this component is critical and sensitive in terms of security, as it would allow attackers to control and potentially harm the simulation equipment. To mitigate this risk, the message broker uses TLS/SSL for encryption and authentication. In addition, RabbitMQ provides access control lists to manage access to the simulation resources in a fine-grained manner.

In comparison to the remaining nodes, the GCP node requires a more complex setup. Docker containers are used to easily deploy & scale the stack of services onto new cloud VM instances. In particular, Docker's Compose & Swarm manage the stack of containers, which consists of the following services:

- An **Nginx** webserver acts as a proxy for WebSocket live data, static web content and HTTP REST APIs for the VILLASweb backend and VILLASnode instances.
- A **NodeJS** backend application implements a REST API for the web interface.
- A **MongoDB** database to provide data persistence for the backend.
- A central **VILLASnode** instance aggregates live data from all laboratories.
- A **RabbitMQ** message broker to exchange control messages between the simulation equipment.

Docker allows us to deploy the stack within minutes to new servers. This is helpful as the server resources only have to be available during the test phases and can be shutdown afterwards.

3.2 Laboratories: RWTH, UPB, WIT, POLITO

All laboratories are equipped with VILLASnode instances to interface local simulation or hardware resources with remote sites. Most of the labs also run a DRTS for the simulation of an electrical subsystem. Within the RESERVE project, OPAL-RT, RTDS and DPsim simulators are available. Generally, EMT based simulators are required for PHIL/CHIL experiments as latencies of the inter-lab connection are too high to allow for a stable operation of a direct HIL interface. Appendix A.1 presents the availability of hardware resources in detail.

4. First Test Simulation Scenario

After testing the quality of the connection between the laboratories, we have conducted several small test simulations. The aim is to validate the functions of all newly developed or extended components, the laboratory interface and the real-time solver. The RTT delay measurements described before, allow an estimation of the dynamics that might be able to propagate through the laboratory interface. However, real-time simulation test cases provide us with more details on the expected dynamic interaction between the distributed simulation laboratories since also the simulators themselves introduce latencies. For a first test simulation, the circuit depicted in Figure 8 is split into two parts, indicated by the red line, and each part is simulated in a different location.

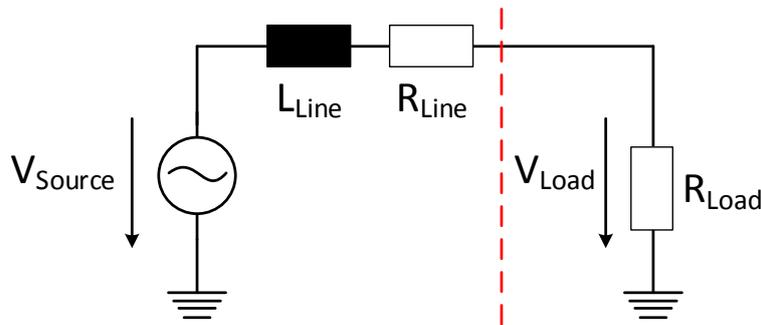


Figure 8: First test simulation circuit

The two locations that we consider first are RWTH and WIT since WIT is further away in terms of network latency than any other simulation partner. The source and line model are simulated in RWTH while the load resistance is simulated in the WIT OpenStack installation.

The depicted circuit is simulated for four different cases. First, both parts are simulated within one instance of DPsim. This case serves as reference for the three co-simulation cases. Secondly, the circuit is simulated using two separate instances of DPsim running on one computer that are interconnected via VILLASnode in the most optimal way. Then, the co-simulation is executed including a network latency emulation between two DPsim instances running on the same machine. Finally, one DPsim instance is running on a machine in RWTH while the other DPsim instance is executed in WIT.

The simulation time step is 1 ms in all cases. In comparison to EMT based simulation, this time step is relatively large and allows us to exchange interface quantities in each step. Whereas for EMT simulations a decimation must be applied.

The voltage source is set to 10 kV , the line inductance L_{Line} is 1 H , the line resistance R_{Line} is $1\ \Omega$ and the load resistance R_{Load} is $10\ \Omega$. At $t = 10\text{ s}$, the load resistance is changed from $10\ \Omega$ to $8\ \Omega$. This change is an ideal step, which would not appear like this in a real setup. Still, it is an interesting test case because the step introduces very high frequencies and, therefore, presents the worst-case scenario for the distributed co-simulation and dynamic phasor approach.

4.1 Co-Simulation without Network Latency

For the co-simulation case where both DPsim instances are coupled in an optimal way, the communication delay is rather small but already visible. The delay between the monolithic reference simulation and the co-simulation is about 3 ms as can be seen in Figure 9.

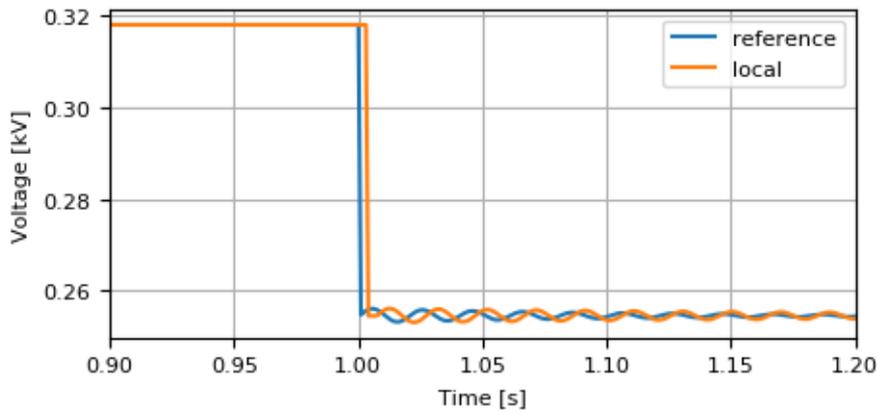


Figure 9: Comparison of monolithic reference simulation and local co-simulation

Since there is no communication network in between the two DPsim instances, this delay is only introduced by the software interface. DPsim is interfaced to VILLASnode through a shared memory segment and the two VILLASnodes instances, one for each DPsim instance, are exchanging data through a UDP connection.

4.2 Co-Simulation including Network Emulation

In this case, the communication delay is emulated using NetEm. As part of traffic control (TC) subsystem, NetEm is a special queuing discipline which is tightly integrated into Linux's networking stack [4]. The properties of the artificial delay for sending data are set to $30\text{ ms} \pm 1\text{ ms}$, normal distributed for each VILLASnode. This results in an emulated RTT of about 60 ms which is close to the measured RTT in Section 2.1. Therefore, the delay in simulation should be comparable to the distributed case. As can be observed in Figure 10, the resulting delay in the simulation is about 32 ms .

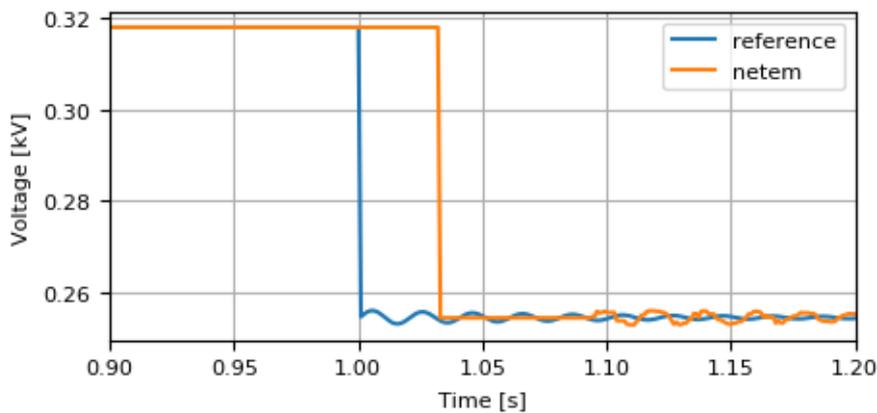


Figure 10: Comparison of monolithic reference simulation and NetEm co-simulation

4.3 Distributed Co-Simulation

As depicted in Figure 11, the distributed simulation between RWTH and WIT features a delay of about 33 ms . Hence, the measurement of the delay as in Section 2.1 and the simulation including emulated delay as presented in Section 4.2 allow a very good prediction of the behavior of the distributed simulation.

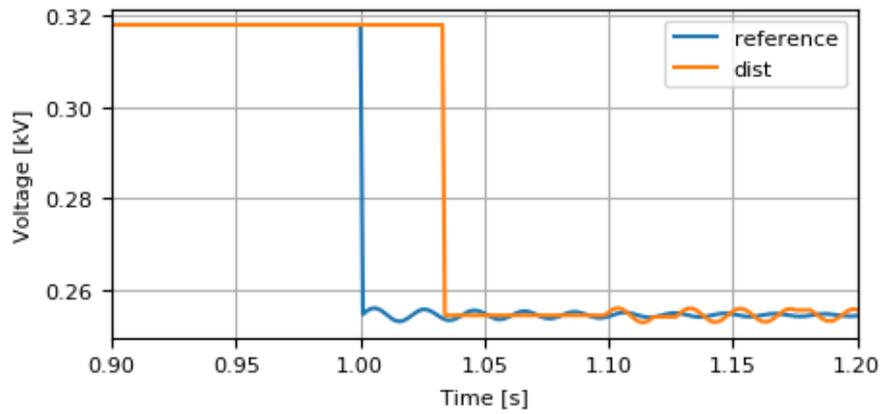


Figure 11 Comparison of monolithic reference simulation and distributed co-simulation

4.4 Summary

The simulation results show that the interface between the co-simulation interface VILLASnode and the open source real-time simulator DPsim has been successfully established. Furthermore, it can be seen that the overhead caused by VILLASnode is very small compared to the latency expected from the network connection with the partners in RESERVE. The comparison with the network connectivity measurement in Section 2 shows that predictions of the delay in the simulation can be very accurate based on the network measurements.

5. Planned Simulation Scenario

The next steps in task 4.3 involve the testing of the co-simulation infrastructure by simulating small grids. These tests go beyond the material which is covered in this deliverable but at this point we would like to give an outlook of what is to come in the associated task. An example for such grids, the IEEE 9-bus reference grid, is depicted in Figure 12.

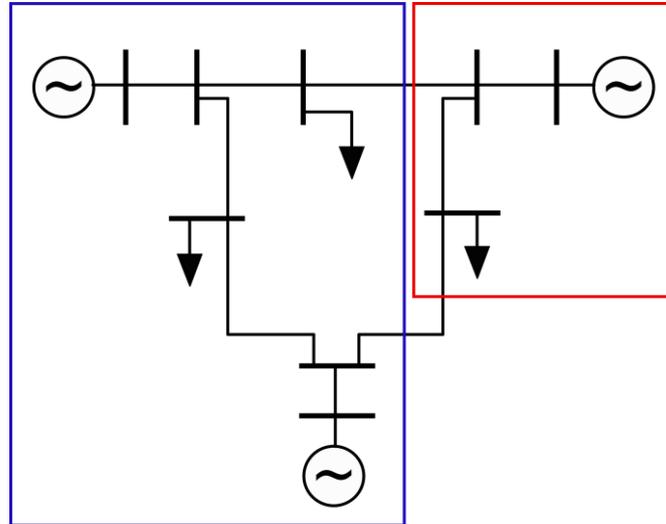


Figure 12: IEEE 9-bus system divided into two parts

The system will be split into two parts for co-simulation, one subsystem featuring two generators and the other subsystem with the remaining generator. The first simulation case is full mechanical inertia implementing all three generators as synchronous machines. Starting from this, the generators are one by one replaced with renewable (distributed) generation interfaced through converters. As long as there is at least one synchronous generator in the system, the simulation case refers to scenario Sf_A. At the point where all generators are consisting of converter based systems, the simulation refers to Sf_B. The control for these low- and no-inertia systems are developed in WP2. For this test case, the co-simulation infrastructure will look as shown in Figure 13. The blue section will be simulated by RWTH while the red section will be simulated by POLITO.

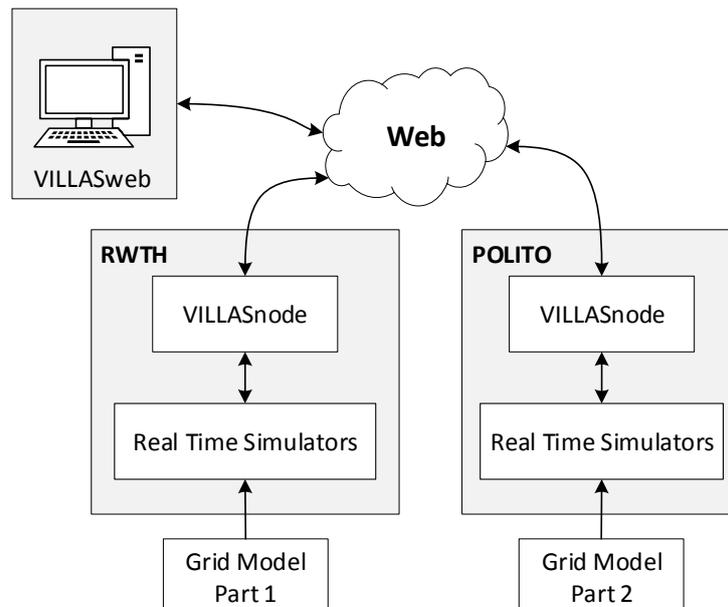


Figure 13: Co-simulation scenario for the IEEE 9-bus test case

6. Conclusion

Initial tests have shown that DPsim simulator and the VILLASnode gateway can be used to conduct simple geographically distributed simulations in real time over the Internet.

Small test cases demonstrate the successful integration of the open source simulator DPsim and VILLASnode. It is also shown that the communication network measurements and the simulation test cases are coherent. Furthermore, the simulation test points out an important condition for the validity of co-simulation results. The dynamics of variables which are propagated between the simulators should not be faster than the communication delay or otherwise the results will be different. This is exactly why dynamic phasors can be an important tool to support distributed real-time simulation. As described in D4.2, they allow the mapping of high frequency events to lower frequencies which decreases the impact of the communication delay.

Upcoming tests must validate these results with by using more complex networks consisting of multiple sources and real generator models. The simulation should be extended to include higher harmonics in order to demonstrate the feasibility of distributed frequency control studies over the co-simulation interface.

7. References

- [1] B. Ford, P. Srisuresh, und D. Kegel, „Peer-to-Peer Communication Across Network Address Translators.“, in *USENIX Annual Technical Conference, General Track*, 2005, S. 179–192.
- [2] G. Sliepen, „The difficulties of a peer-to-peer VPN on the hostile Internet“, in *Proceedings of the Free and Open Source Software Developers' European Meeting-FOSDEM*, 2010, Bd. 109.
- [3] M. Mirz, A. Estebasari, F. Arrigo, E. Bompard, und A. Monti, „Dynamic phasors to enable distributed real-time simulation“, in *2017 6th International Conference on Clean Electrical Power (ICCEP)*, 2017, S. 139–144.
- [4] S. Hemminger und others, „Network emulation with NetEm“, in *Linux conf au*, 2005, S. 18–23.
- [5] S. Vogel, „Development of a modular and fully-digital PCIe-based interface to Real-Time Digital Simulator“, Aachen, Aug-2016.

8. List of Abbreviations

ACS	Institute for Automation of Complex Power Systems
DFN	Deutsches Forschungsnetzwerk (German NREN)
GARR	Gestione Ampliamento Rete Ricerca (Italian NREN)
GCP	Google Cloud Platform
GEANT	European Research and Education Network
HEAnet	Higher Education Authority Network (Irish NREN)
ICMP	Internet Control Message Protocol
NAT	Network Address Translation
NREN	National Research and Education Network
POLITO	University Politecnico di Torino, Italy
RoEduNet	Romanian Education Network (Romanian NREN)
SSD	Solid State Disk
TC	Traffic Control (Linux Network / Queuing Subsystem)
TSSG	Telecommunications Software & Systems Group
UPB	University Politehnica of Bucharest
VM	Virtual Machine
VPN	Virtual Private Network
WIT	Waterford Institute of Technology

9. List of Figures

Figure 1: Relations between deliverables in WP4 and other work.....	6
Figure 2: Routing Paths between ACS, POLITO, UPD and WIT.....	7
Figure 3: Network latency of ACS-WIT connection.....	8
Figure 4: Round-trip time graph.	8
Figure 5: Cumulative probability of RTT.....	9
Figure 6: A fully meshed peer-to-peer VPN network.....	10
Figure 7: Simulation components available in the pan-European co-simulation	11
Figure 8: First test simulation circuit.....	13
Figure 9: Comparison of monolithic reference simulation and local co-simulation	14
Figure 10: Comparison of monolithic reference simulation and NetEm co-simulation	14
Figure 11 Comparison of monolithic reference simulation and distributed co-simulation.....	15
Figure 12: IEEE 9-bus system divided into two parts.....	16
Figure 13: Co-simulation scenario for the IEEE 9-bus test case	16

10. List of Tables

Table 1: Nation Research and Education network providers.....	7
Table 2: Round-trip time matrix.....	8

Annex

A.1 Laboratory inventory

This chapter gives a rough overview of which simulation resources are available at the RESERVE simulation sites.

A.1.1 GCP

The central server is hosted at Google Cloud Platform (GCP) on a datacenter in Belgium. Currently, the following configuration is used:

GCP Compute Instance

CPU:	1	vCPU	core	(Intel	Ivy	Bridge)
RAM:	2.75	GiB				
Region:	europe-west1-c (located at St. Ghislain, Belgium)					
Harddisk:	16	GiB	Solid State Disk (SSD)			

A.1.2 RWTH

At ACS, VILLASnode is running on a desktop workstation, running Fedora Rawhide, with RT_PREEMPT patched kernel 4.9.x

Custom Workstation

Motherboard:	Gigabyte X79 UD5
CPU:	Intel Sandybridge-E
RAM:	16 GiB DDR
Harddisk:	128 GiB SATA3 SSD
Network Adapter:	Intel Dual Port Gigabit Ethernet Adapter

In addition, the ACS node is equipped with two M-Series PCIe DAQ cards for interfacing the gateway with hardware.

ACS is also hosting eight GPC/PB5 equipped RTDS racks, an OPAL-RT OP5600 simulator with 12 cores and a compute cluster of several nodes for large scale simulation with DPsim. The compute cluster is interfaced via Infiniband to the VILLASnode gateway. The simulators are interfaced via direct Gigabit Ethernet links and custom FPGA fiber optic interfaces [5].

In addition, the ACS real-time lab is also equipped with HIL power amplifiers and inverters which will be used in WP5.

A.1.3 UPB

In UPB, the VILLASnode is running on a desktop workstation, running under Linux 4.8.10-200, having the following hardware characteristics

HP Workstation

Motherboard:	Intel C602
CPU:	Intel i3-2120 @ 3,30 GHz Clock Speed, Cache 3 MB, 1333
.MT/3 Memory	4 GiB SDRAM
Harddisk:	1 TiB SATA 7200 rpm 6Gb/s 3.5" HDD
Network Adapter:	Intel 82579LM Gigabit
Monitor:	Fujitsu SL23T-1 LED

The UPB node is special in a way that it is the only node, which sits behind a Network Address Translation (NAT) firewall, which assigns IP addresses dynamically. The UPB node is capable of simulating a subsystem using the DPsim solver.

A.1.4 WIT

The WIT VILLASNode instance is running on a virtual machine (VM) that is hosted on an OpenStack cloud environment, running the Ubuntu Linux 16.04 Xenial operating system for AMD64 and having the following characteristics

OpenStack Compute Instance

RAM:	2 GiB
CPU:	1 vCPU core
Harddisk:	8 GiB

Using WIT's OpenStack deployment, we can demonstrate the scalability of the DPsim solver on multiple VM instances and evaluate the possibility to use virtualization for scalable real-time simulations.

A.1.5 POLITO

At POLITO, VILLASnode is running on a commercial desktop system, running Fedora 25, with a RT_PREEMPT patched kernel 4.9.x

ASUS E810 Desktop Computer

Motherboard:	
CPU:	Intel® Core™ i5 4460T
RAM:	4 GiB SODIMM DDR3 Synchronous 1600 MHz
Harddisk:	500 GiB HGST HTS545050A7
Network Adapter:	1 GiB Ethernet Connection I217-LM

The POLITO node is interfaced to an OP5600 real-time simulation target from OPAL-RT.

A.2 OpenStack Cloud environment

The VM also has a special OpenStack security group profile assigned to it, with the following provisions

```
ALLOW IPv4 22/tcp from 0.0.0.0/0
ALLOW IPv4 655/udp from 0.0.0.0/0
ALLOW IPv6 to ::/0
ALLOW IPv4 to 0.0.0.0/0
ALLOW IPv4 12000-12001/udp to 0.0.0.0/0
ALLOW IPv4 1234/tcp from 0.0.0.0/0
ALLOW IPv4 655/tcp to 0.0.0.0/0
ALLOW IPv4 12000-12001/udp from 0.0.0.0/0
ALLOW IPv4 655/udp to 0.0.0.0/0
ALLOW IPv4 1234/tcp to 0.0.0.0/0
ALLOW IPv4 443/tcp from 0.0.0.0/0
ALLOW IPv4 655/tcp from 0.0.0.0/0
ALLOW IPv4 80/tcp from 0.0.0.0/0
```

This allows certain ports to be open towards the VM. The VM is also assigned a public facing Internet Protocol (IP) address, which means it is available on the public Internet.

A.3 VILLASnode installation

The VILLASnode was installed from its source code on this VM by making a git clone of the VILLASnode open-source project².

```
$ git clone --recursive https://git.rwth-aachen.de/VILLASframework/VILLASnode.git
```

² <https://git.rwth-aachen.de/VILLASframework/VILLASnode>

Then adding in the dependencies for the project were installed on the VM as per the documentation³.

And then running the following commands

```
$ sudo make install-thirdparty
$ make && sudo make install
```

Once this process was complete a check can be made to ensure the version of VILLASnode has been installed with

```
$ villas node --help
```

The WIT is in the process of putting together a source code package that would create and automatically install a VILLASnode instance on any OpenStack cloud environment.

³ <https://villas.fein-aachen.org/doc/node-installation.html#node-installation-source>