"Integrating NFV-MANO with Wireless Services: Experiences and Testbed Development"

Nikos Makris, Christos Zarafetas, Alexandros Valantasis,
Thanasis Korakis, Leandros Tassiulas

Centre for Research & Technology Hellas, CERTH, Volos, GR

University of Thessaly, Electrical & Computer Engineering Department, Volos, GR

Yale University, Department of Electrical Engineering, New Haven, USA

Presentation Outline

1) **Introduction - Motivation**
2) **The NITOS wireless testbed**
3) **Slicing wireless interfaces**
4) **Discussion**
5) **Conclusions - Future Work**

Presentation Outline

1) **Introduction - Motivation**

2) The NITOS wireless testbed

3) Slicing wireless interfaces

4) Discussion

5) Conclusions - Future Work

Introduction

- NFV is a major enabler for 5G:
  - Reduced CAPEX and OPEX costs for deployed (virtual) services
  - Optimized reconfiguration of the provided services
  - Easy integration of new services
  - Operational efficiency due to the concepts of network slicing and multi-tenancy
- Through NFV-MANO a generic architecture for the management and orchestration of virtualized resources was defined
  - Releases of compatible tools with the NFV MANO architecture (e.g. OpenSourceMANO)
  - Abstraction of hardware computational, network and storage resources
  - Focus only on the orchestration of services, formation of forwarding graphs, etc.

Our Contribution

- NFV-MANO is a generic architecture, currently targeting datacenter resources
- What happens if instead of traditional Ethernet connectivity we bring to the picture wireless networks?
- How can multi-tenancy and slicing be addressed in such an environment?
  - Solutions exist in literature, but not with production grade software (like OpenFlow)
- Application of the solution over a wireless testbed, with multiple wireless technologies that the deployed VNFs can use
- Compute nodes are distributed in the infrastructure, and services are orchestrated through OpenSourceMANO (OSM)

Presentation Outline

1) Introduction - Motivation

2) **The NITOS wireless testbed**

3) Slicing wireless interfaces

4) Discussion

5) Conclusions - Future Work

NITOS Testbed

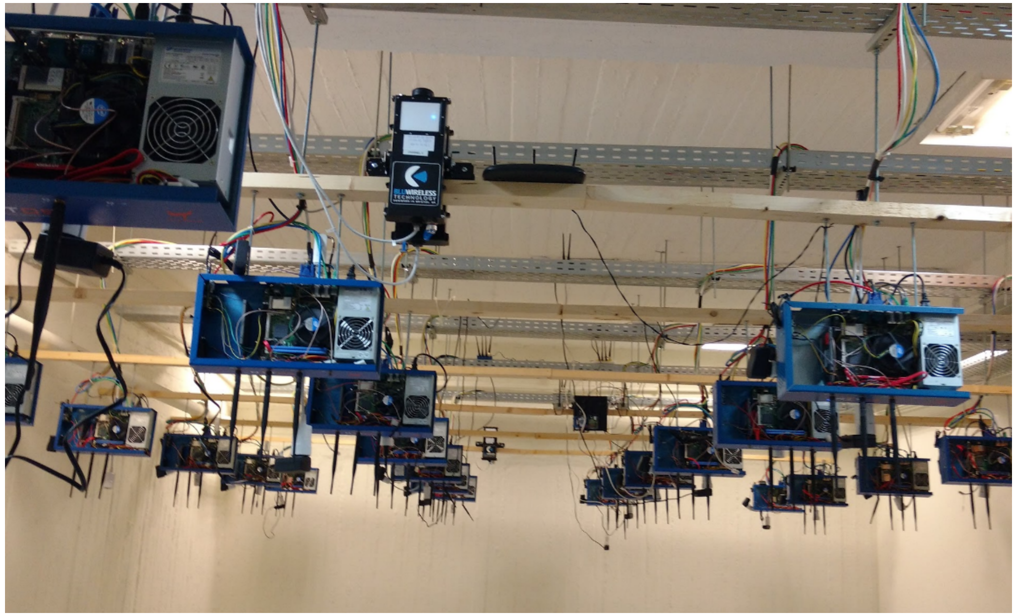
- Wide offering of resources for Wireless/Wired/IoT/Cloud remote experimentation

- Organized in three different setups, to cover different experimental settings
  - Indoor RF-isolated setup
  - Outdoor, prone to external uncontrolled interference
  - Office setup, with low external interference

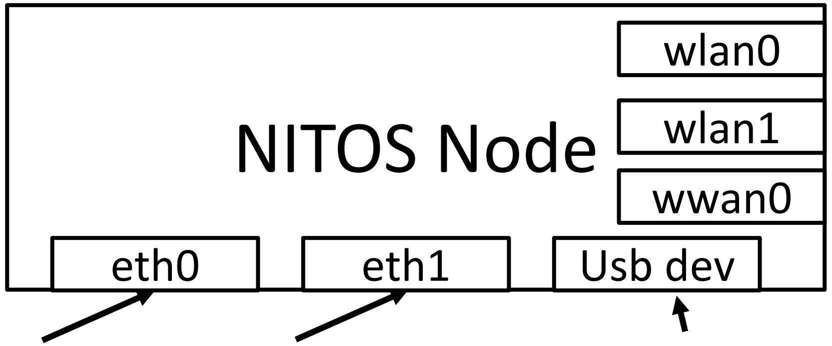
- Resources can be ”mixed” and “matched” from different locations in a single end-to-end slice

NITOS Testbed (indoor/outdoor)

- Over 100 physical machines as testbed nodes that can be reserved with the testbed's tools, offering different technologies (COTS LTE, WiFi, WiMAX, Open Source LTE, SDRs, mmWave Units, OpenFlow switches, ZigBee/LoRa/LoRaWAN sensors and cloud infrastructure)



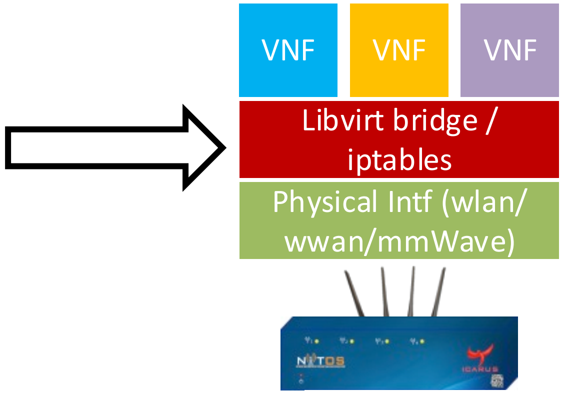
Overall Architecture



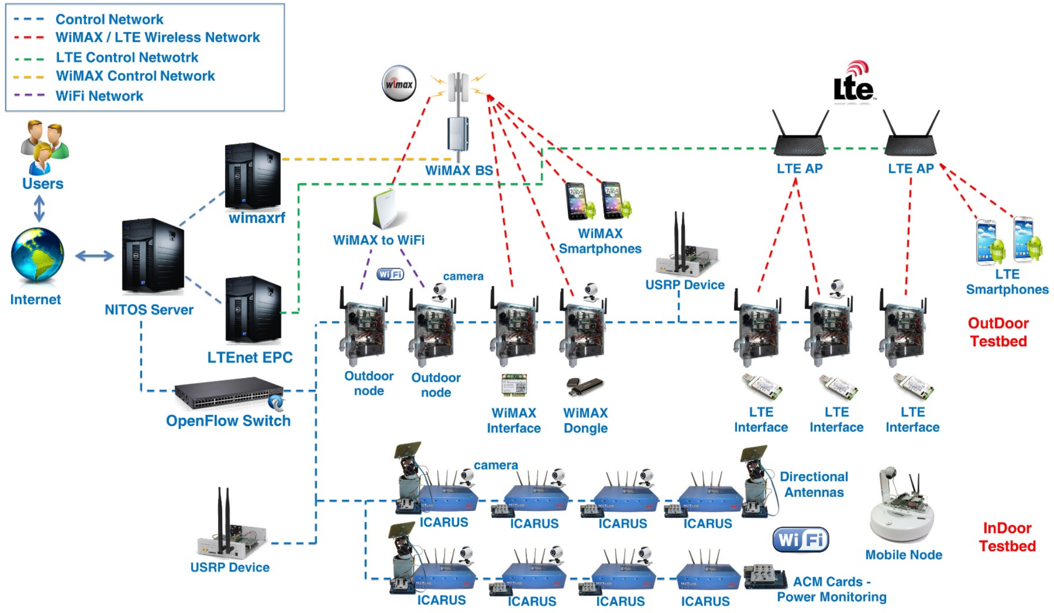
Control network (addresses assigned by the testbed)

Experimental network (free addressing, can be used for mmWave or Ethernet experiments)

USRP device, allowing the node to operate as a base station



Baseline Host Image



Presentation Outline

1) Introduction - Motivation

2) The NITOS wireless testbed

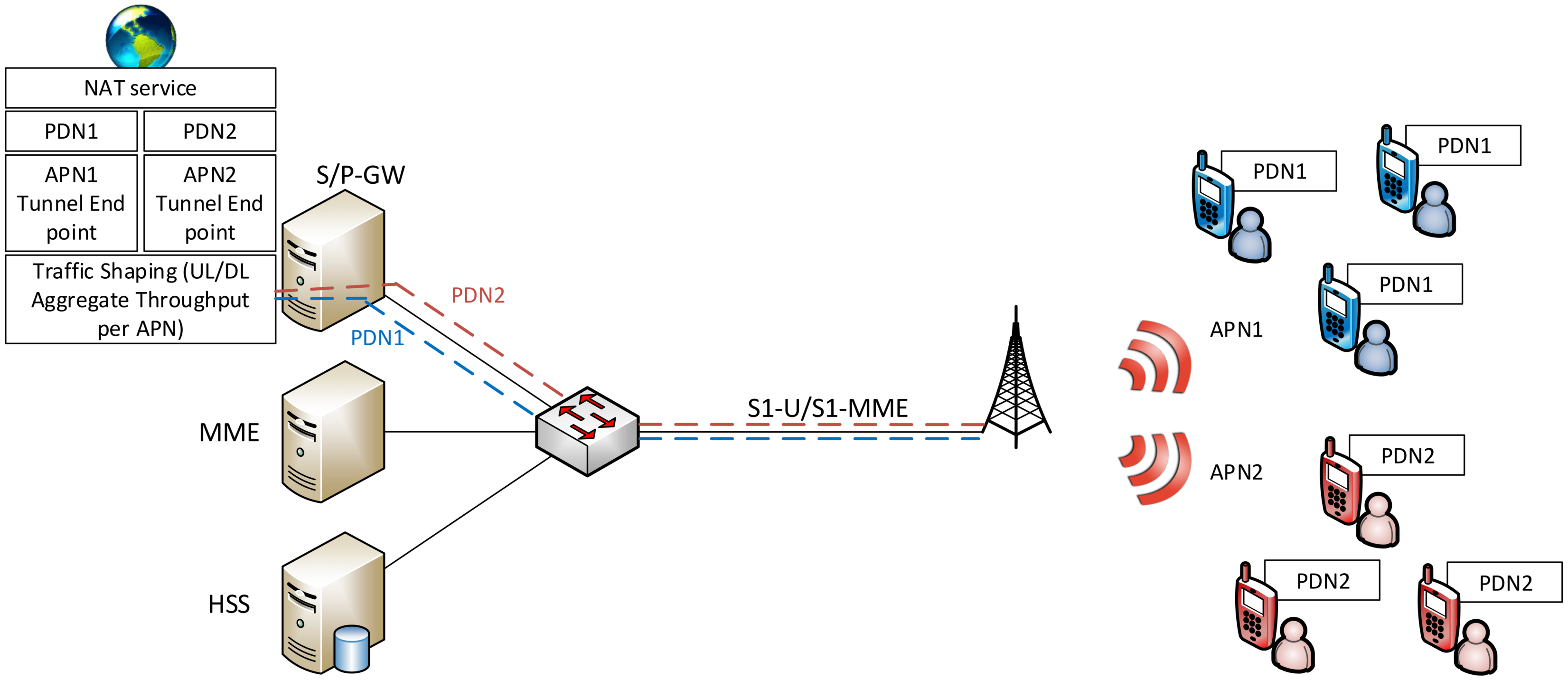
3) **Slicing wireless interfaces**

4) Discussion

5) Conclusions - Future Work

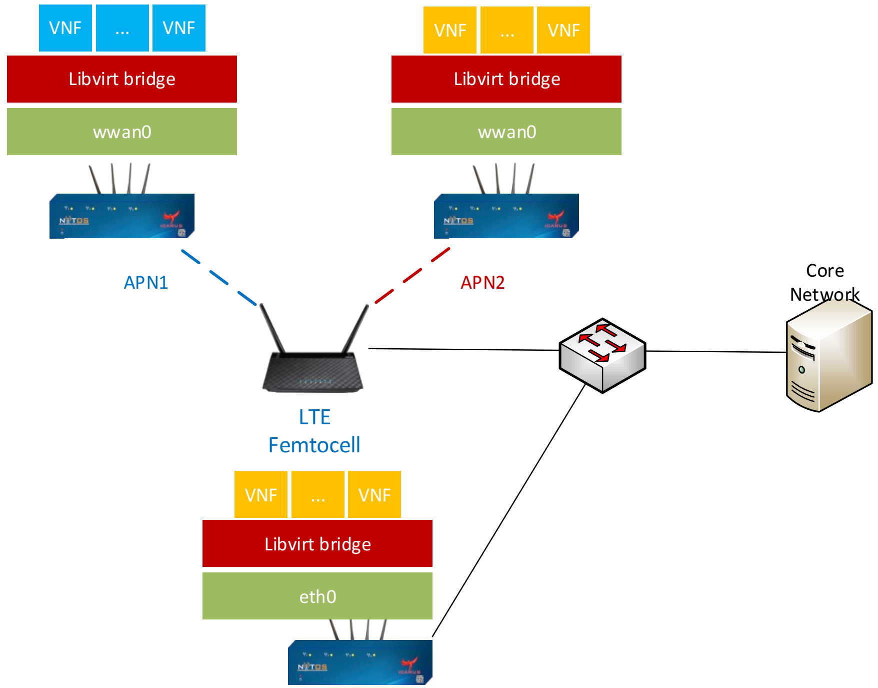
COTS LTE Network slicing (1/2)

- Using different PDNs on the Core Network side
- Different PDNs equal to different tunnels terminated at the PDN-GW



COTS LTE Network slicing (2/2)

- Each PDN is mapped to a unique APN in the RAN
- Traffic Shaping at the Core Network to guarantee transmissions at the RAN
- Prior to orchestration, calls are made to a specific testbed service to prepare the slice of the infrastructure

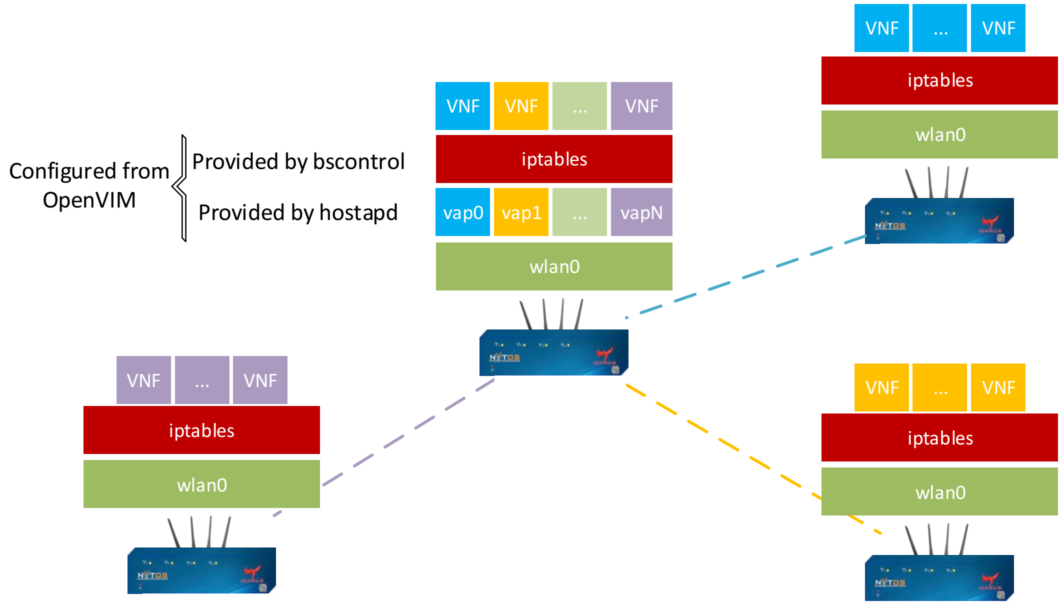


WiFi Network Slicing (1/2)

- Using Virtual AP configuration: multiple APs can be running on top of the same physical card
- Interface can be configured as an AP, or a client that is associated to it
- Multiple APs can be instantiated on top of the same physical card
  - Each with its own ESSID (network name), but physical parameters (e.g. wireless channel) are the same across all the VAPs

WiFi Network Slicing (2/2)

- Iptables are used to ”bridge” the VNF interfaces with the virtual APs
- Prior to instantiation of the VNF, calls are made to the *bscontrol* service that takes care of configuring the network interfaces on the compute node
- VNFs can be completely isolated without sharing any traffic between different slices
- Work in progress: integrate support for supporting guarantees on the air time that each VNF will get

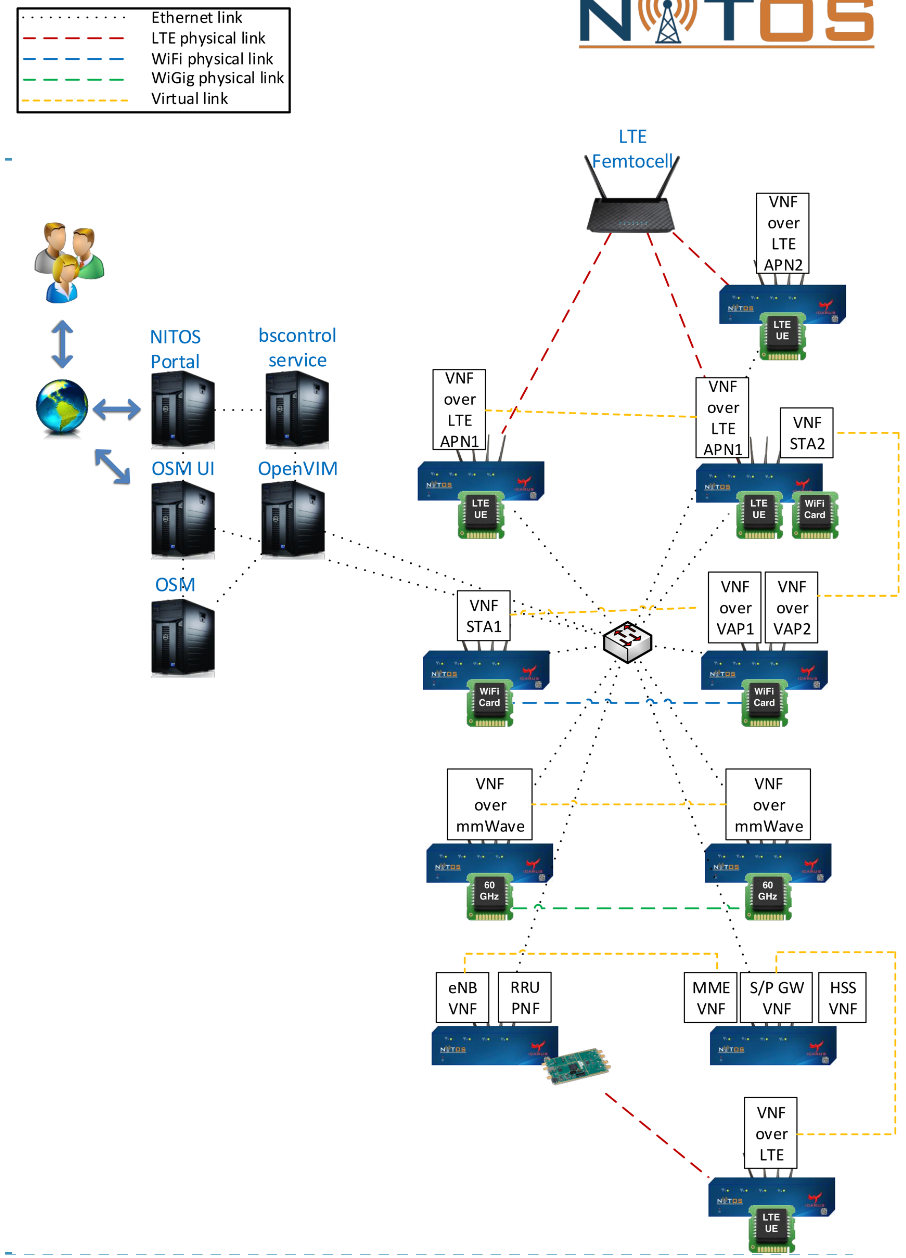


mmWave Slicing

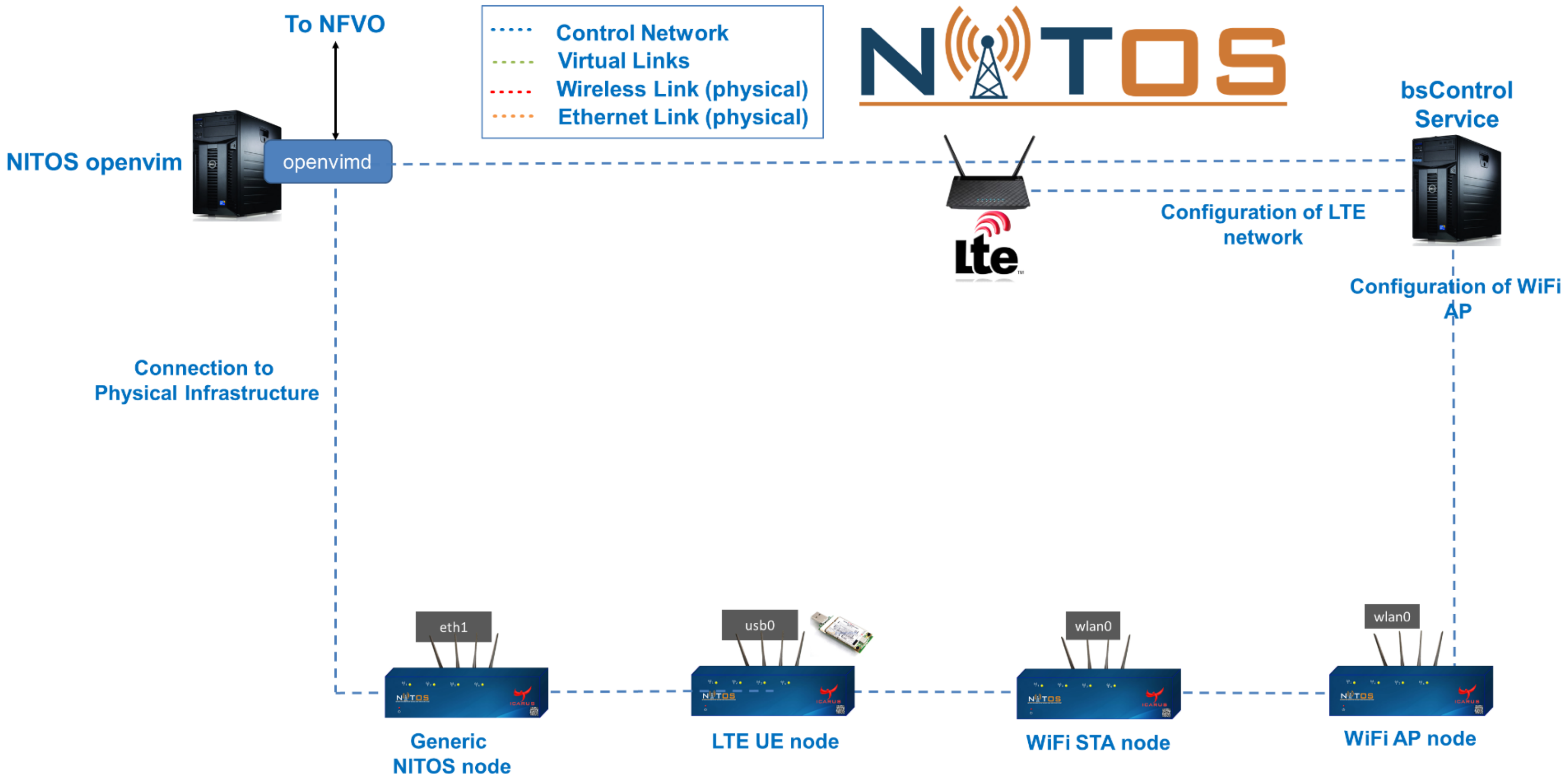
- Each mmWave node of the testbed can be reached by setting a VLAN ID
- Support for different slices:
  - Through QinQ VLANs
  - Outer VLAN tag addresses each node separately
  - Inner VLAN tag addresses each slice
- mmWave nodes support OvS
  - Work in progress: manage the transmissions of each slice with guarantees on the airtime

Putting it all together

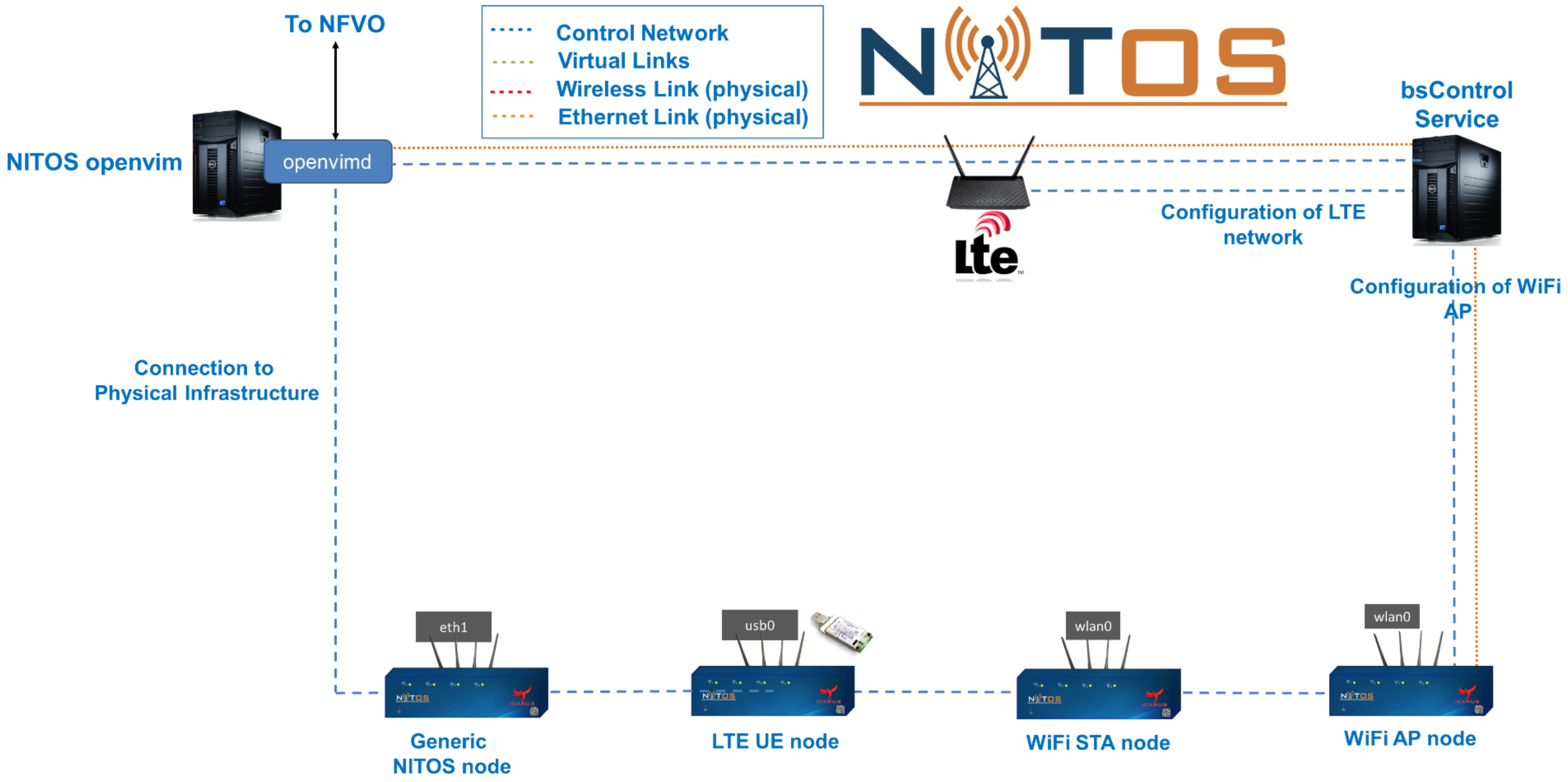
- Augmented description of the VDU specifying a new type of interface
  - LTE with APN parameters for the RAN for providing guarantees
  - WiFI AP or WiFi Station
  - mmWave connection
- Dedicated services with REST APIs take care of the handling of the networking interfaces
- Called directly from the VIM for preparing the compute node prior to orchestration
- Extensions to the OSM UI API to facilitate the orchestration of VNFs with wireless connectivity



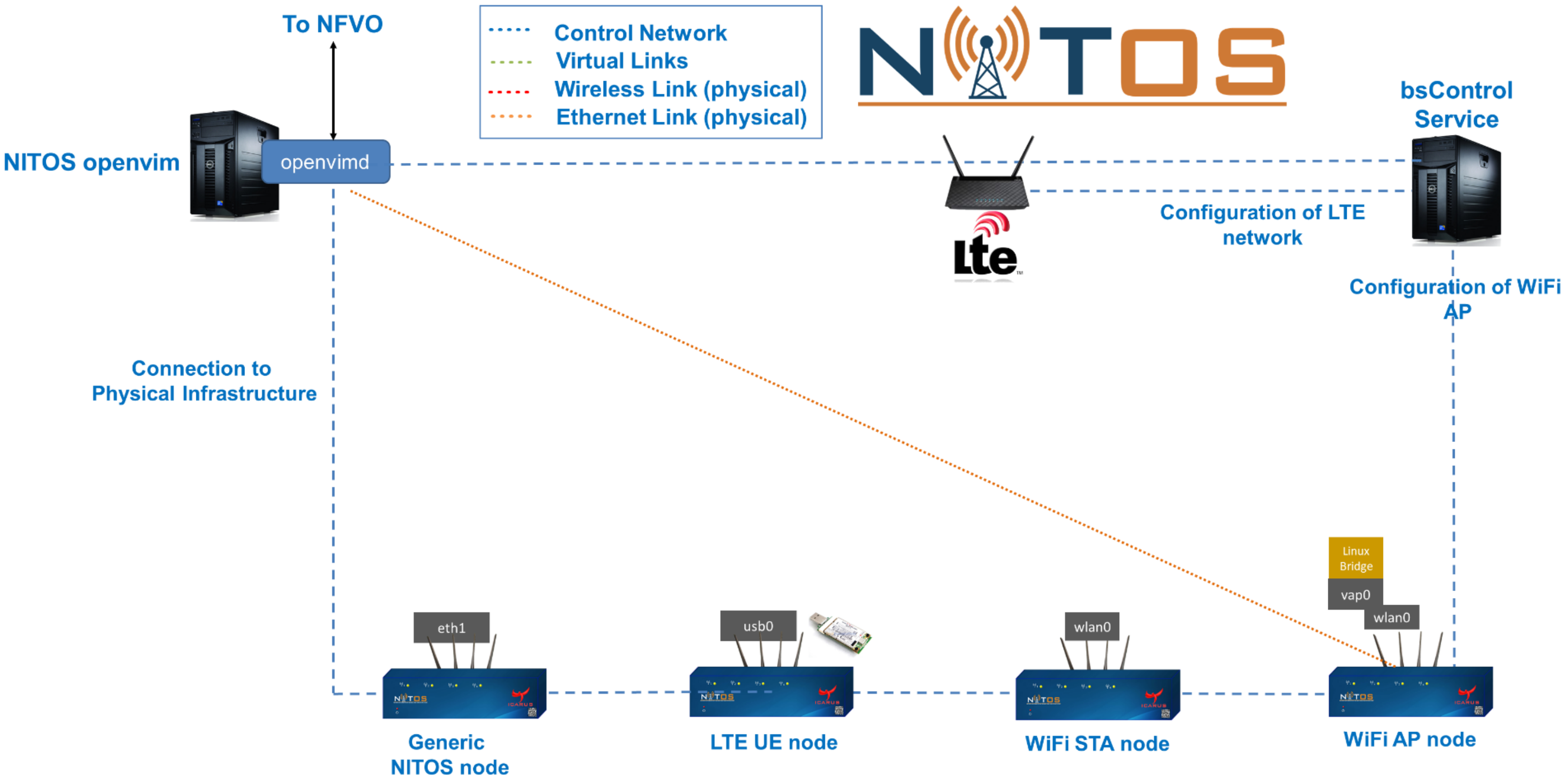
Example Execution – WiFi case



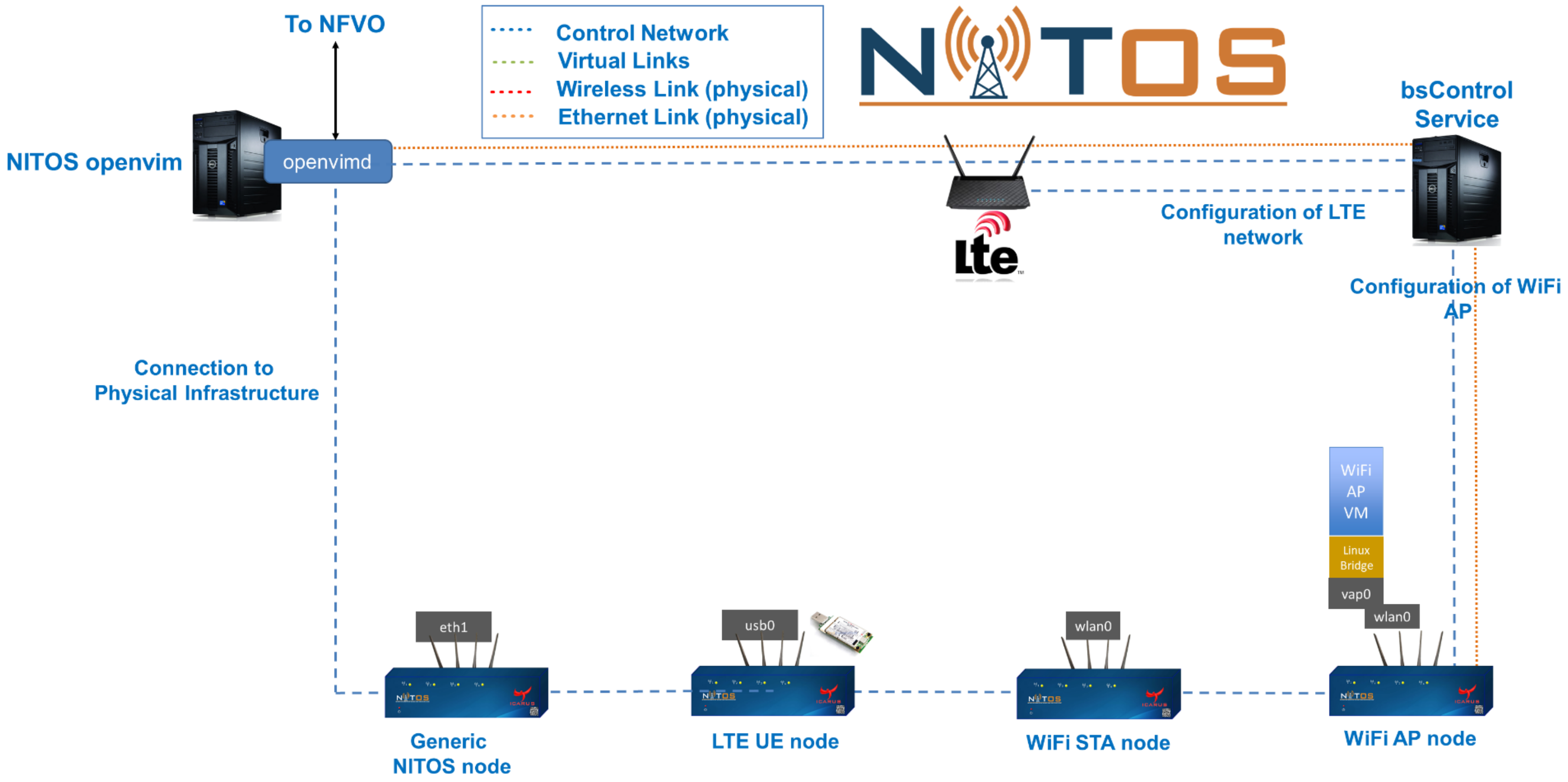
Request for a VM with a WiFi VAP1 connection



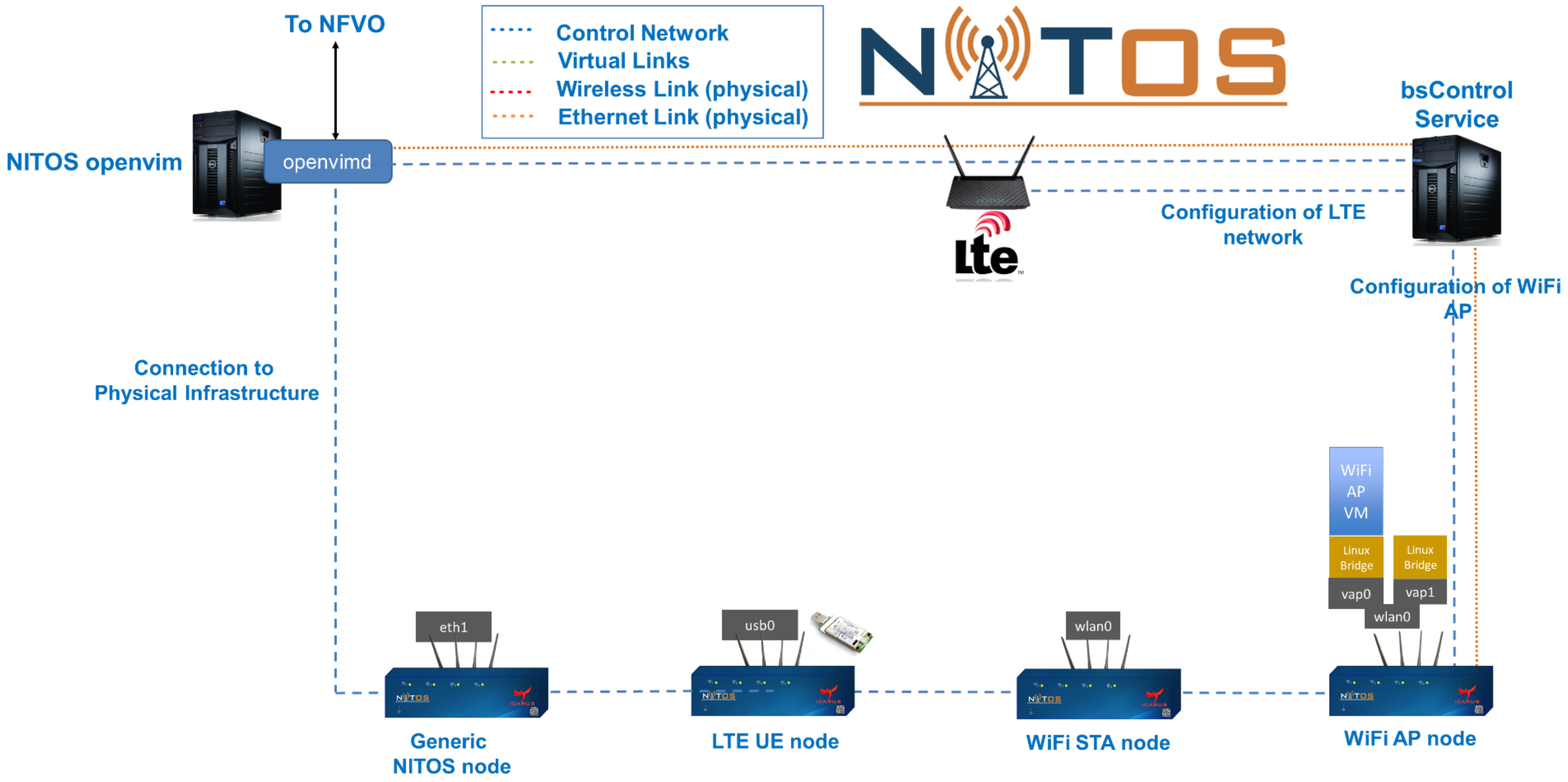
PHY settings are configured, instantiating the service



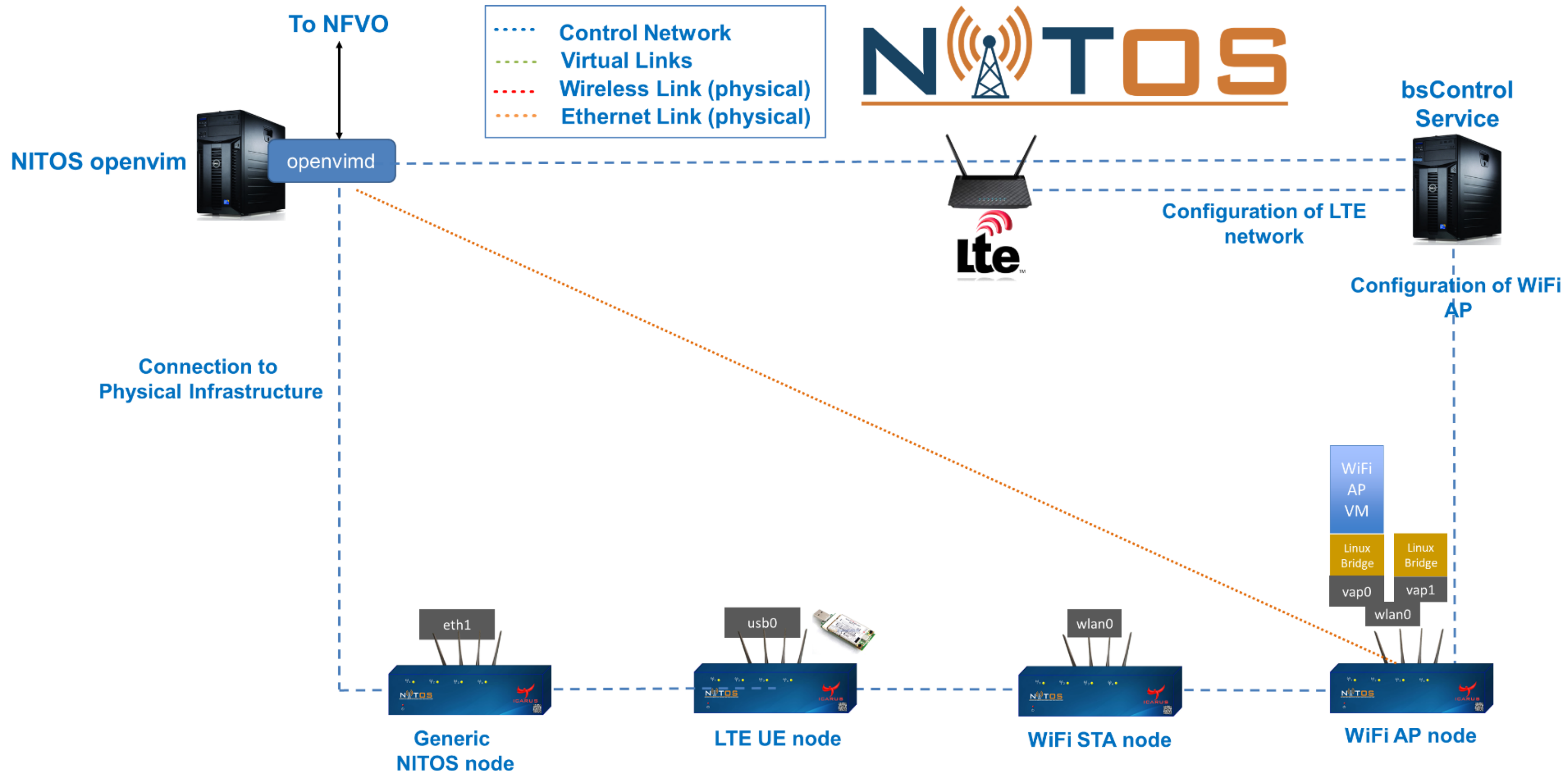
Setting up a second VM with a VAP2



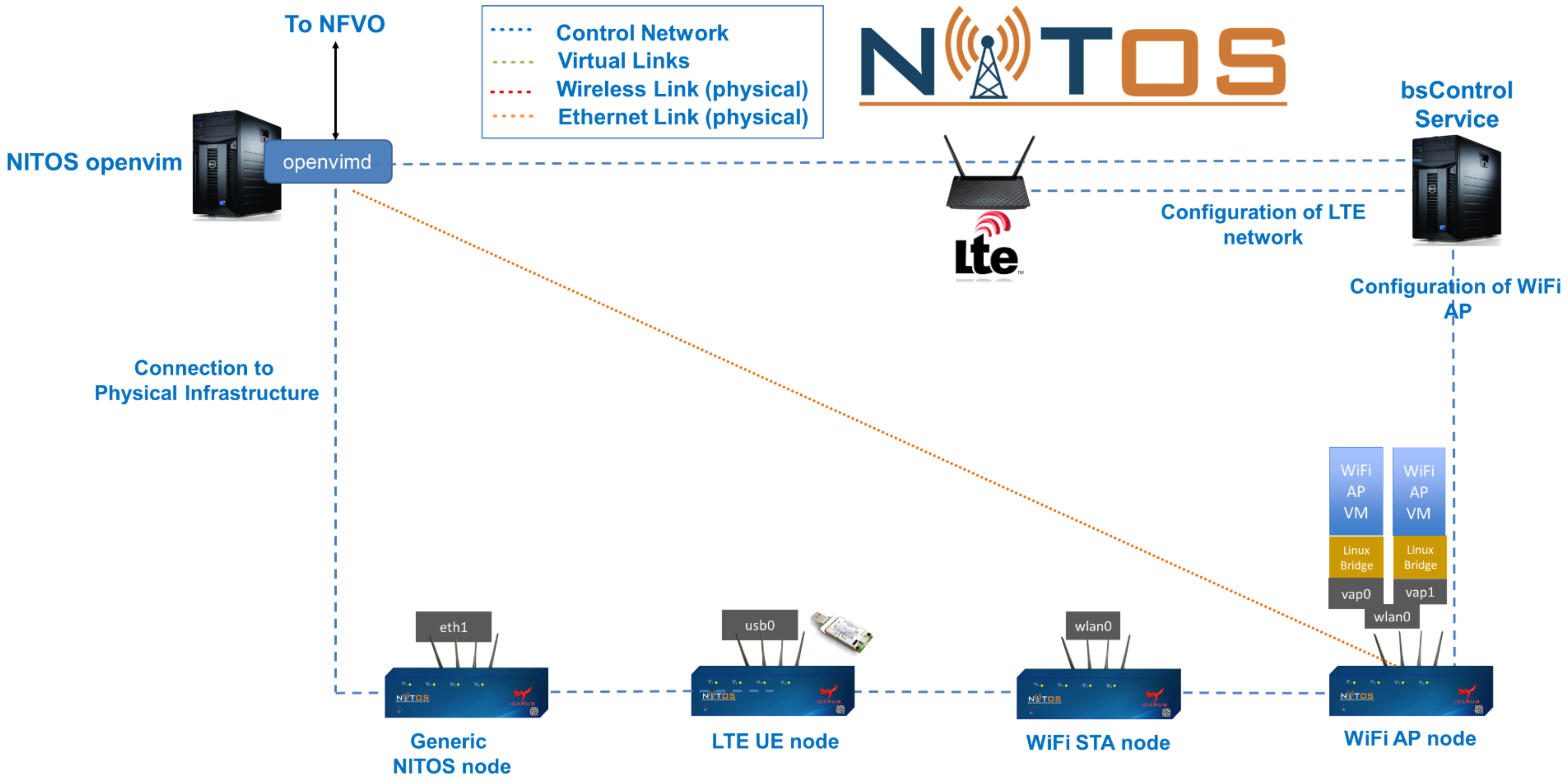
Setting up a second VM with a VAP2



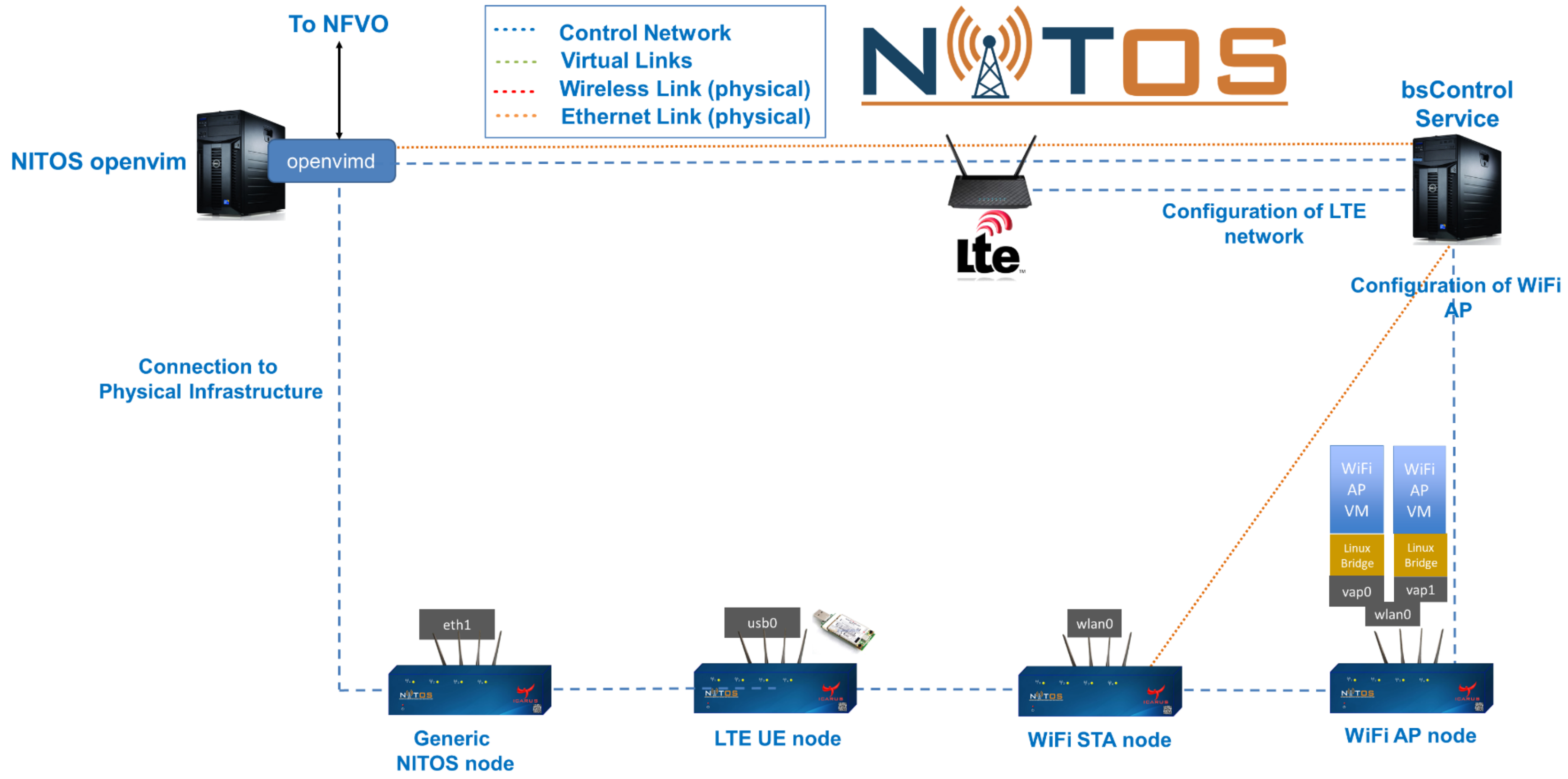
PHY settings are configured, instantiating the service



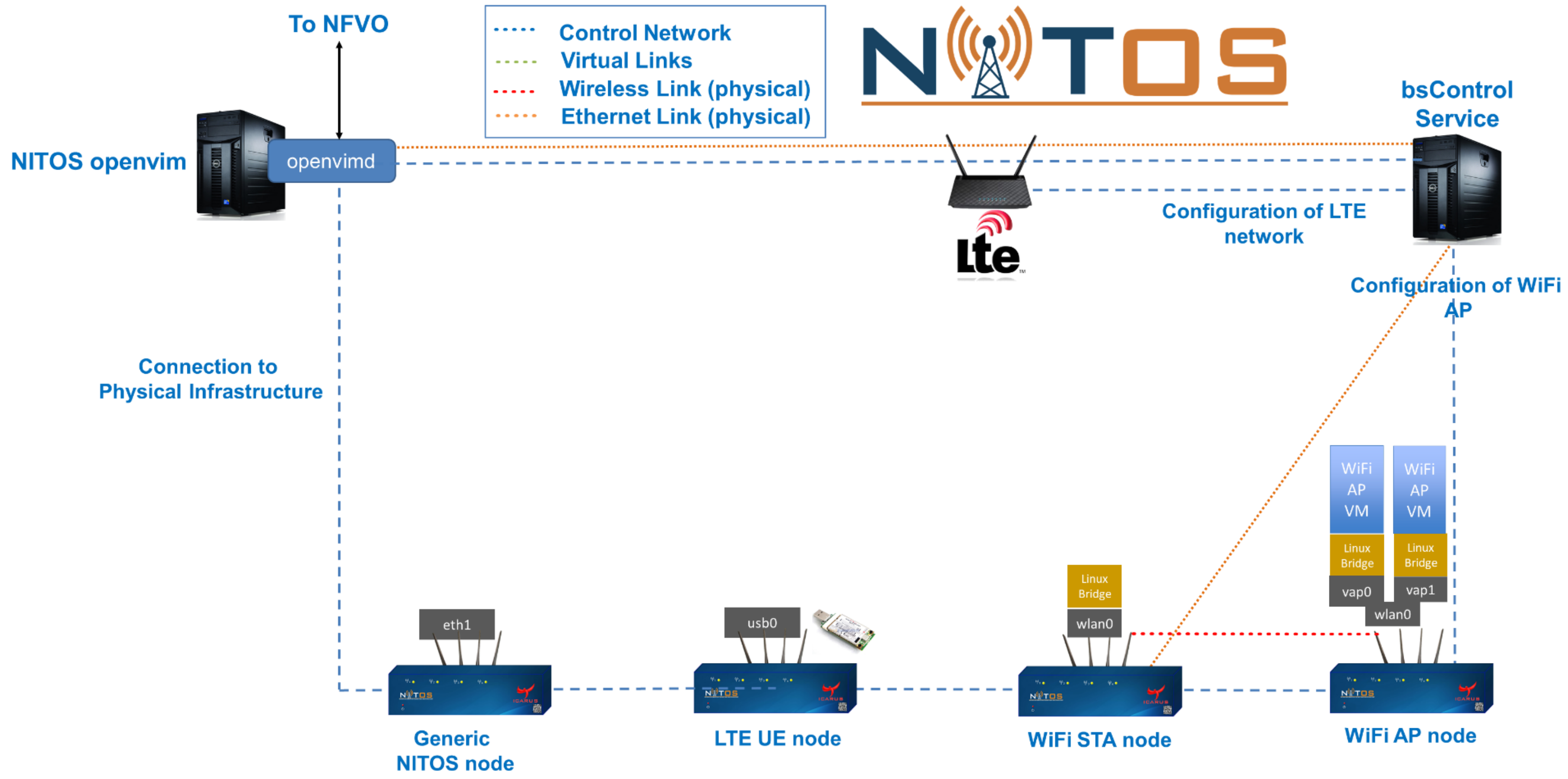
2 VMs over two VAPs are configured



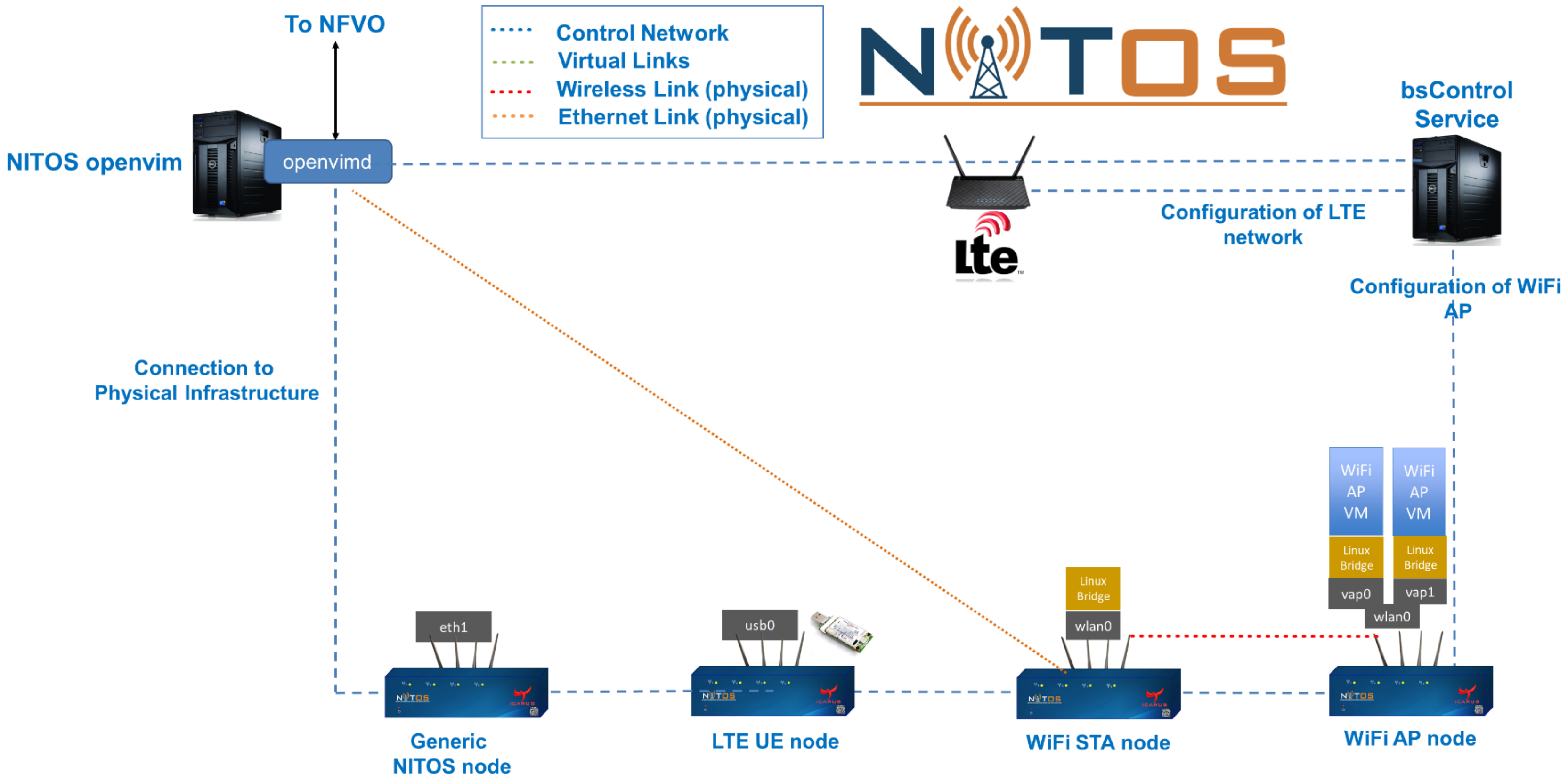
Setting up a VM to use a client connection



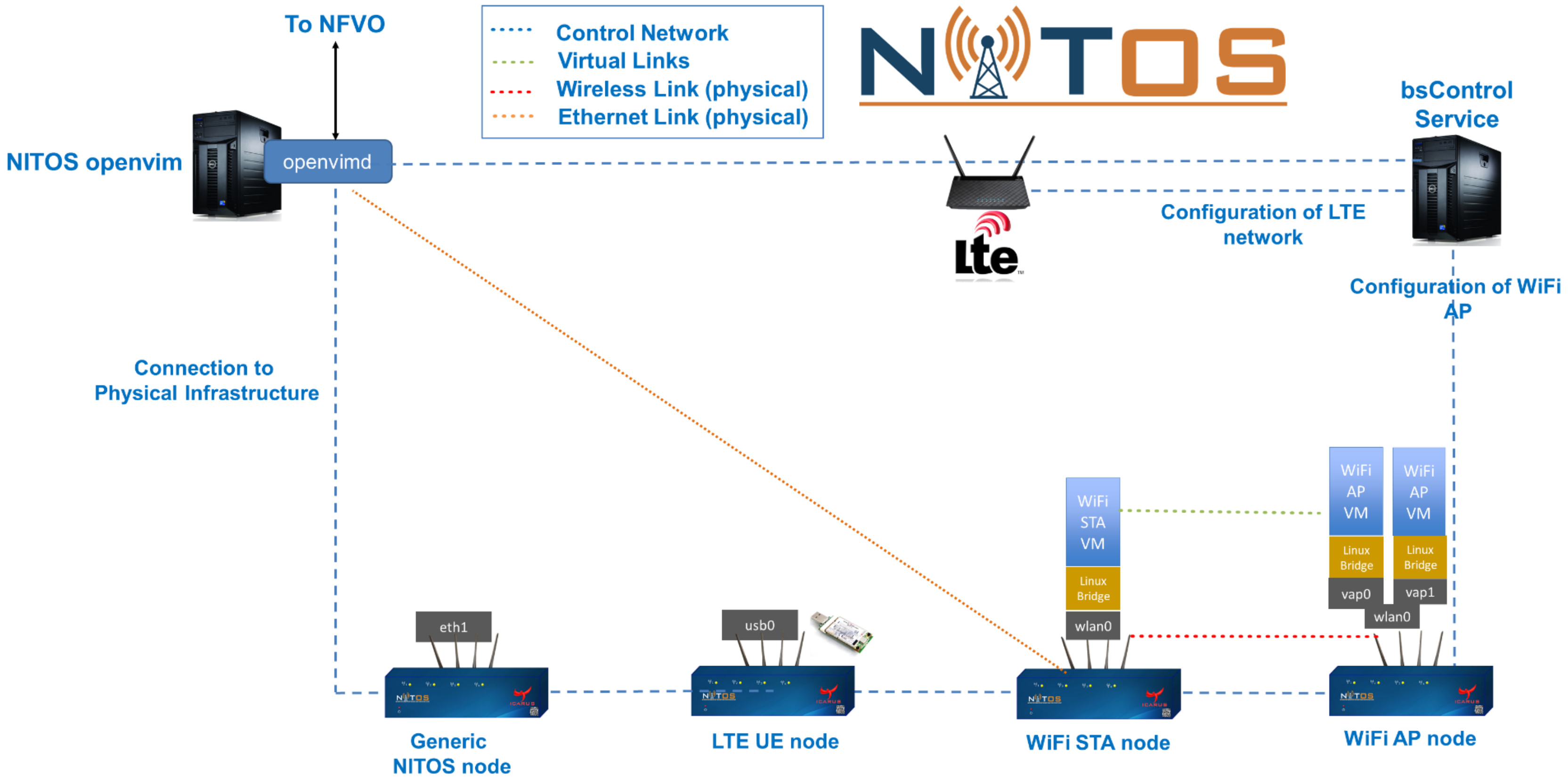
PHY connection from client to AP established



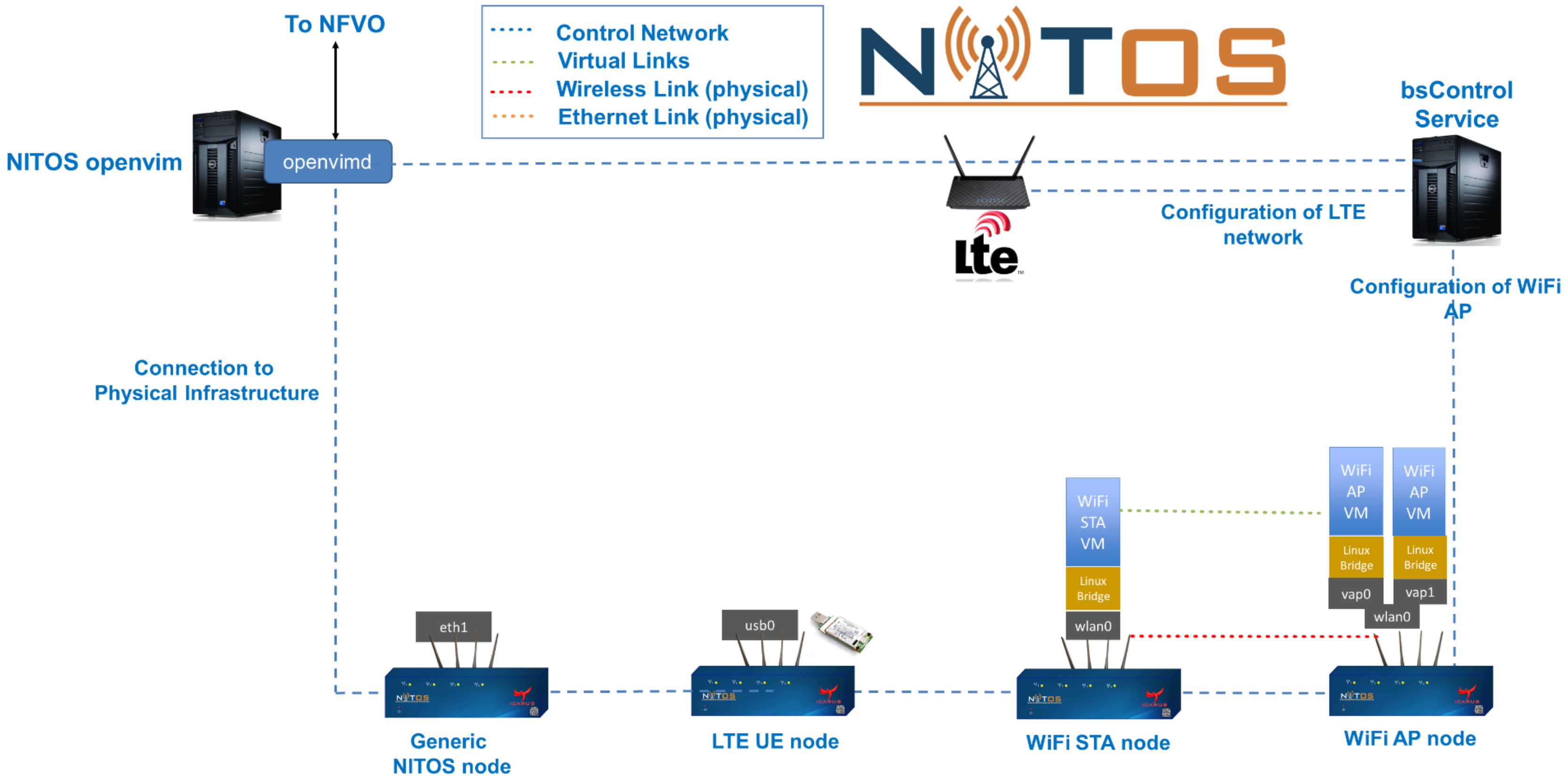
Instantiating the station side VM



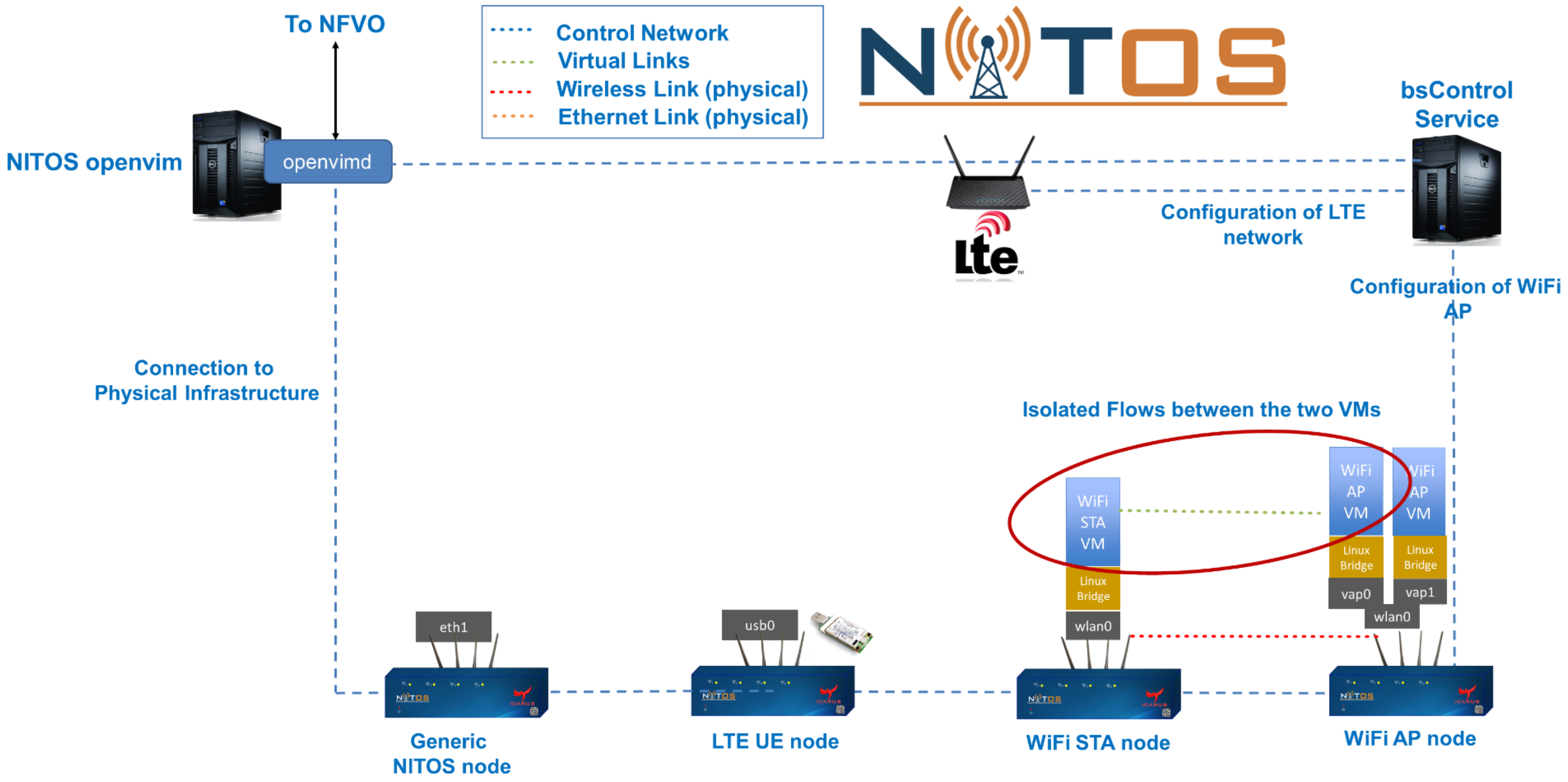
Request for a VM with a WiFi VAP1 connection



Virtual link between the VMs over wireless is provided



Virtual link between the VMs over wireless is provided



Presentation Outline

Discussion (1/2)

- Good solution for mildening the learning curve of new users of the testbed
  - Traditional methods would require the user to use the terminal and send commands to the testbed services for provisioning a single LTE link
  - Easy and repeatable examination of large wireless topologies
- Portability of VNFs across different testbeds
  - Generic setup of the nodes of the testbed, lots of “datacenter” deployable solutions can be orchestrated in the testbed
- Effortless setup of VNFs for 5G
  - E.g. VNF of base station that is orchestrated with a mmWave front-/backhaul link

Discussion (2/2)

- Generic solution to be applied at other testbeds as well
- Can serve as an overlay on top of Future Internet Research and Experimentation (FIRE) testbeds
- Need for efficiently mapping the existing resource descriptions (RSpecs) to NSDs/VNFDs
- Solution for this mapping:
  - Use of the SFA protocol used to expose the testbeds API to prepare the compute nodes prior to experimentation
  - Mapping of the VNFDs to the RSpecs for Virtual Resources

Presentation Outline

1) Introduction - Motivation
2) The NITOS wireless testbed
3) Slicing wireless interfaces
4) Discussion
5) **Conclusions - Future Work**

Conclusions and Future Work

- Extensions to the OSM framework for orchestrating VNFs over wireless links
- NFVI is the generic infrastructure in the NITOS wireless testbed
- Slicing is applied for the different wireless networks that can be configured through our solution
- Making use of dedicated services for setting up the wireless links prior to orchestration
- Current developments use OSM Rel TWO and THREE
  - Migrating the functionality to Rel FOUR
  - Integrating all the functionality that is currently implemented in OpenVIM to OpenStack

Thank you for your attention!

Nikos Makris, nimakris@uth.gr
Centre for Research & Technology Hellas, CERTH, Volos, GR
University of Thessaly, Department of Electrical & Computer Engineering, Volos, GR

The research leading to these results has received funding by GSRT, under the act of "HELIX-National Infrastructures for Research" MIS No 5002781 and through the European Horizon 2020 Programme under grant agreement No 732497 (5GinFIRE).