ICN2020 and Internet2 testbeds
Horizon 2020 EU/Japan

Luca Muscariello
Principal Engineer, Cisco Systems
26th of March 2017 – ICNRG - Chicago
Objective

- Deploy ICN testbeds with federation across the world
- High speed backbone that covers US, EU, Japan, anybody could join by providing own hardware/compute
- Based on the Community ICN software
Several partners

- ICN2020
- Internet2
- Géant
- You

- Univ. Gottingen
- Univ. College London
- Univ. Rome
- Ericsson
- Cisco
- SystemX
- KDDI
- KKE
- Univ. Osaka
- Osaka City Univ.
Internet2 and Géant

- Internet2 can use several compute units in different PoPs
  - Cisco UCS
  - Intel DPDK 10Gbps

- Géant Testbed Service (GTS)
  - Make available to ICN2020 a number of compute units to deploy ICN
vICN and testbed deployment

- vICN control run in a central unit
- Can control and orchestrate CICN routers running in Linux containers
- Rest API, SSH, bash, websockets
- Scalable and reliable to failure thanks to reconciliation
vICN

VIRTUAL, REAL AND SIMULATED NODES MANAGEMENT

LIGHTWEIGHT VIRTUALIZATION:
- Linux containers (LXC/LXD): prebuilt images for fast prototyping and deployment

EMULATED WIRELESS MEDIUM:
- 802.11n (WiFi) and LTE wireless channels
- Mobility across APs/nodes

FAST NETWORKING:
- DPDK
- UPP

NETWORK TOPOLOGY MANAGEMENT:
- OVS, OVS/dpdk, VLAN
- OpenFlow for traffic control
- Linux TC for bw shaping

MULTIPROTOCOL SUPPORT:
- SSH, bash, REST, websockets,

FLEXIBLE RESOURCE MODEL:
- Nodes, interfaces, channels, applications
- Resources dependency graph
- Secure access / Slicing
- Consistency check

MULTITHREADED ORCHESTRATION:
- Match specification to current state and perform necessary operations for reconciliation.
How does it work?
Deploy a simple network

A WiFi AP

A consumer

A producer

An LTE+EPC instance
JSON network configuration (1/2)

```json
{
  "resources": [...],
  "settings": {
    "network": "192.168.2.0/24"
  }
}
{
  "type": "Physical",
  "name": "server",
  "hostname": "localhost"
},
{
  "type": "NetDevice",
  "device_name": "br0",
  "node": "server",
  "managed": false
}
{
  "type": "LxcContainer",
  "image": "ubuntu1604-cicnsuite-rc1",
  "name": "cons",
  "node": "server",
  "category": "tablet",
  "x": 1,
  "y": 2
},
{
  "type": "LxcContainer",
  "image": "ubuntu1604-cicnsuite-rc1",
  "name": "lte",
  "node": "server",
  "category": "lte",
  "x": 2,
  "y": 3
},
{
  "type": "LxcContainer",
  "image": "ubuntu1604-cicnsuite-rc1",
  "name": "prob",
  "node": "server",
  "category": "video-server",
  "x": 3,
  "y": 2
}
}```
JSON network configuration (2/2)

```json
{
  "type": "MetisForwarder",
  "node": "cons"
},
{
  "type": "MetisForwarder",
  "node": "wifi"
},
{
  "type": "MetisForwarder",
  "node": "lte"
},
{
  "type": "MetisForwarder",
  "node": "prod"
},
{
  "type": "WebServer",
  "node": "prod",
  "prefixes": [ "/webserver" ]
},
{
  "type": "EmulatedWiFiChannel",
  "name": "wch",
  "node": "server",
  "ap": "wifi",
  "stations": ["cons"],
  "control_port": 30001
},
{
  "type": "EmulatedLteChannel",
  "name": "lch",
  "node": "server",
  "ap": "lte",
  "stations": ["cons"],
  "control_port": 30002
},
{
  "type": "CentralIP",
  "ip_routing_strategy": "spt"
},
{
  "type": "CentralICN",
  "icn_routing_strategy": "spt",
  "face_protocol": "udp4"
}
```
Deploy/control a vICN network (1/2)

```
cd vicn
python3 setup.py install
vicn -s examples/tutorial/tutorial03-hetnet.json

$ lxc exec prod -- producer-test ccnx:/webserver/test
$ lxc exec cons -- consumer-test ccnx:/webserver/test

lxc exec cons -- metis_control -k keystore.pkcs12 -p password list routes [...]
iface protocol route cost next prefix
  3 STATIC LONGEST   1 ----.----.----.----/.... ccnx:/webserver

Done
```
Deploy/control a vICN network (1/2)

cd vicn
python3 setup.py install
vicn -s examples/tutorial/tutorial03-hetnet.json

$ lxc exec prod -- producer-test ccnx:/webserver/test
$ lxc exec cons -- consumer-test ccnx:/webserver/test

lxc exec cons -- metis_control -k keystore.pkcs12 -p password list routes [...] 
iface protocol route cost next prefix
   3 STATIC LONGEST   1 ---.----.---.---/.... ccnx:/webserver

Done
Deploy a vICN containerized network (1/2)

$ lxc exec cons -- metis_control -k keystore.pkcs12 -p password list connections
[...] 
  3  UP inet4://192.168.2.2:6363 inet4://192.168.2.3:6363 UDP
  5  UP inet4://192.168.2.6:6363 inet4://192.168.2.7:6363 UDP
  7  UP inet4://127.0.0.1:9695 inet4://127.0.0.1:32876 TCP

$ lxc exec cons -- metis_control -k keystore.pkcs12 -p password add route 5 ccnx:/webserver 1
vICN federation and testbed interco

• Running one vICN per region
• Federate vICN instances