8th July 2015

End-to-end Self-Diagnosis of Programmable Networks

José Manuel Sánchez Vilchez

Supervisor: Imen Grida Ben Yahia, Orange Labs, Issy Les Molineaux, France
Thesis Director: Noël Crespi, Telecom SudParis, Evry, France

orange™

TELECOM SudParis

Outline

Context

Scientific contributions

Results

Conclusion and future lines

Outline

Context

Scientific Contributions

Results

Conclusion and future lines

Programmable Networks

Two challenges concerning management of programmable networks

Resilience

Network dynamicity

Single point of failure: SDN controller
Security and scalability

decentralization

redundancy

controller placement

Existing recovery solutions for SDIs:

- mostly **OpenFlow-based**
- only handle physical **faults in the data layer**
- only a **few solutions focus on the control layer and controller**
- **root cause analysis** is not covered yet

Problem statement

Two challenges concerning management of programmable networks

Resilience

End-to end diagnosis in combined SDN and NFV infrastructures

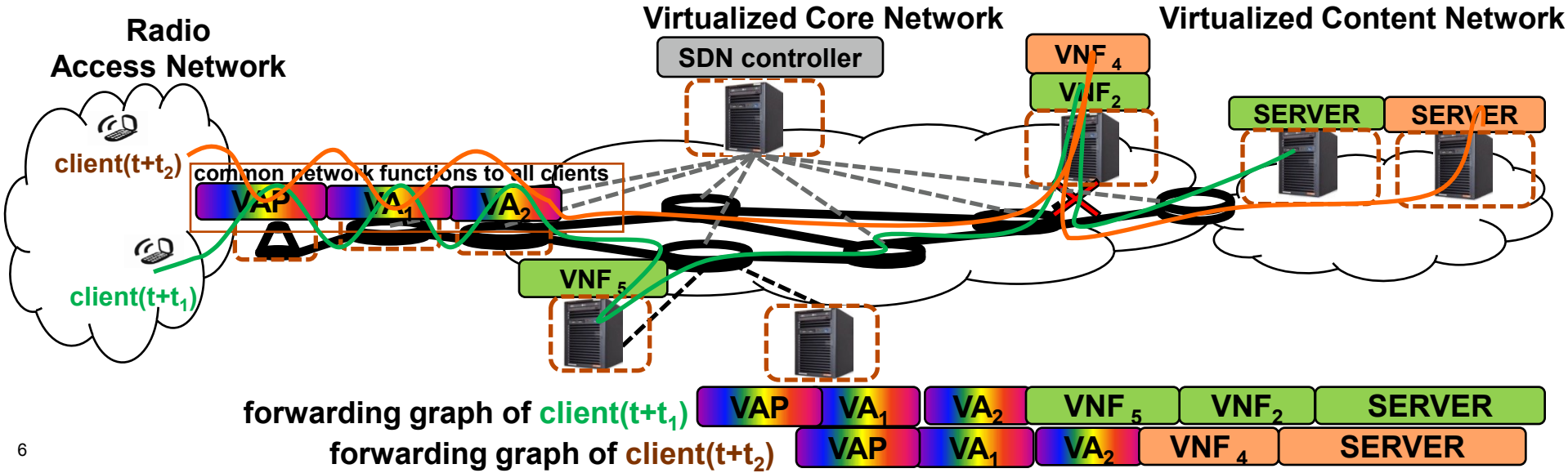
SDN is the underlying layer of NFV based services

The SDN controller dynamically interconnects the VNFs

Network dynamicity

network topology

network functions and services



Outline

Context

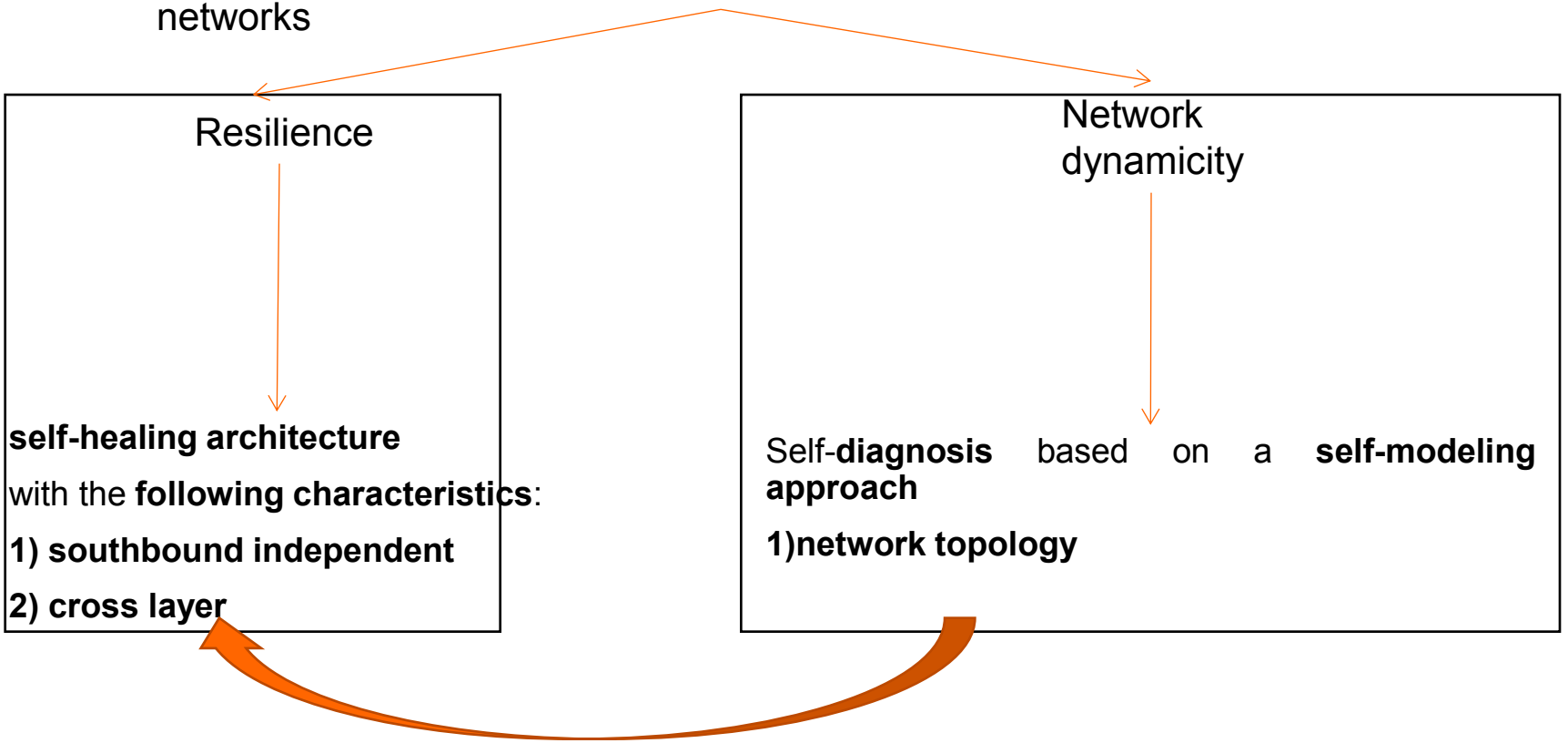
Scientific Contributions
- **Self-Healing arhitecture**
- **Self-Diagnosis module**
- **Proposed templates**

Results

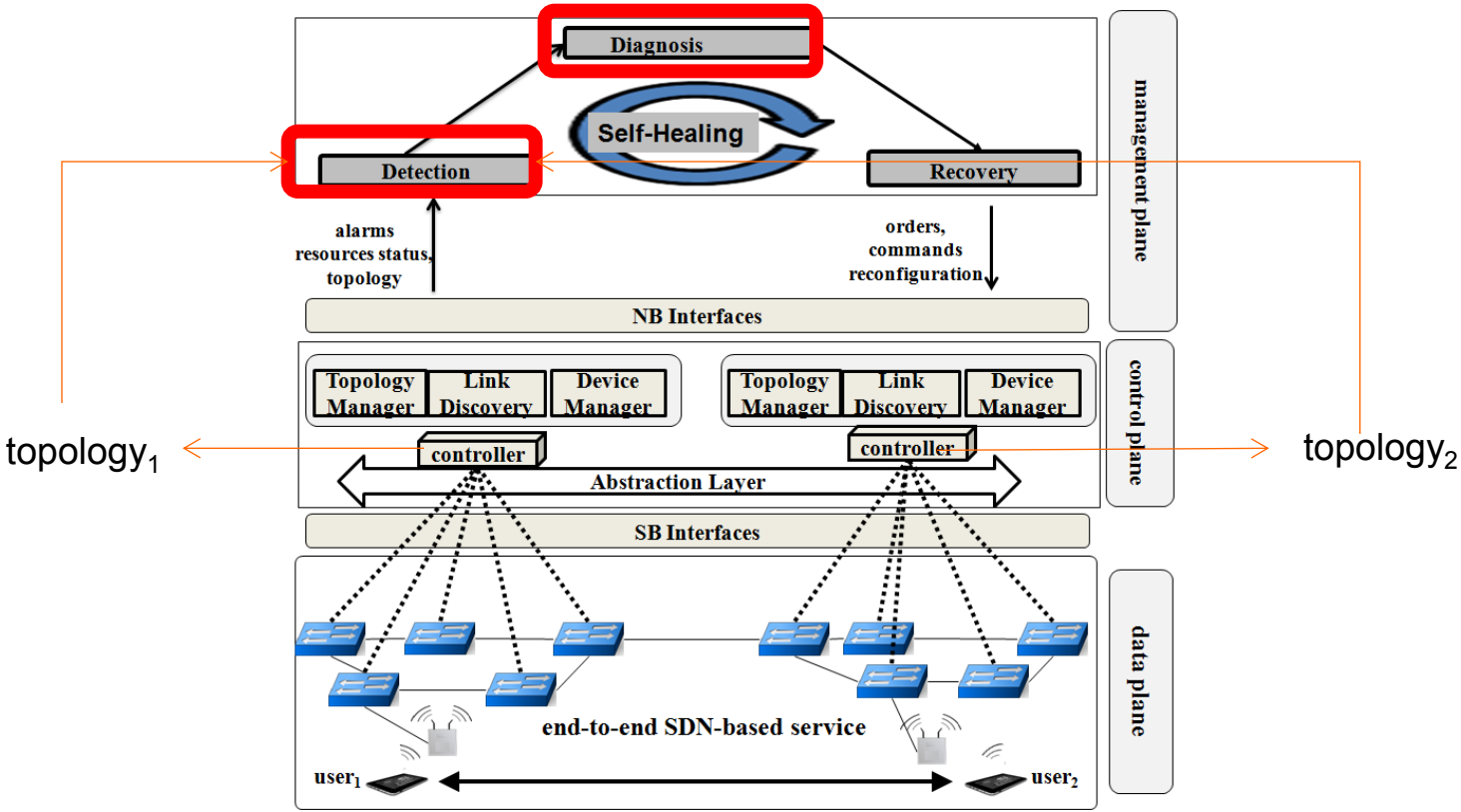
Conclusion and future lines

Scientific Contributions

Two challenges concerning management of programmable networks



Self-Healing cross-layer architecture for SDN



Topology-Aware Self-Diagnosis framework

- Topology-Aware Self-diagnosis:

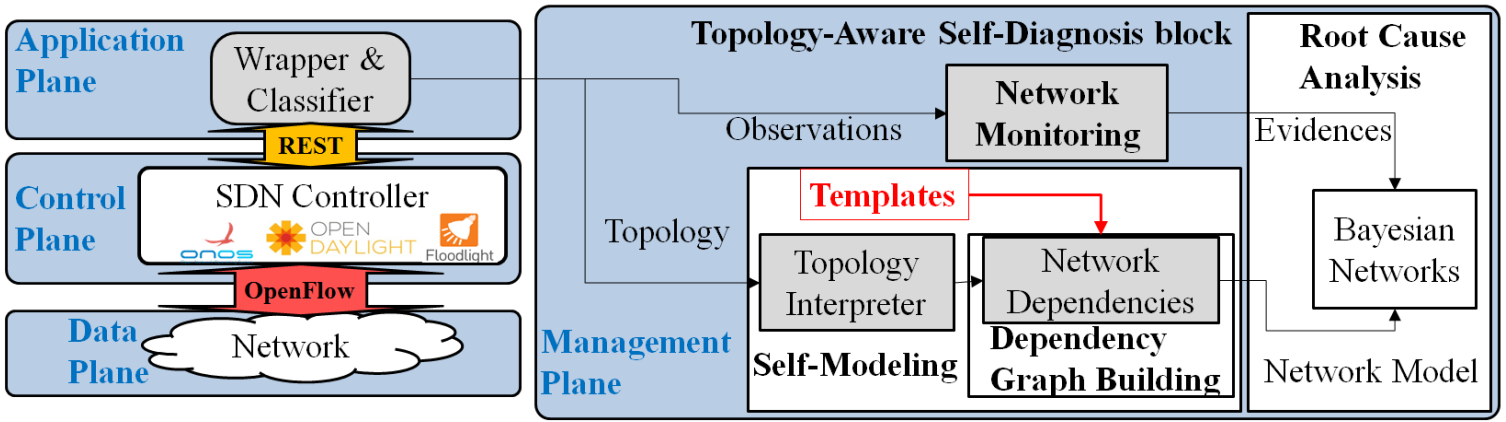
-Root Cause Analysis from the network topology and type of control

-The diagnosis model contains dependencies among network nodes and links

- Based on two blocks:

-Self-modeling: Topology Interpreter + Dependency Graph Building

-Root-Cause analysis based on a model-based Bayesian Networks algorithm



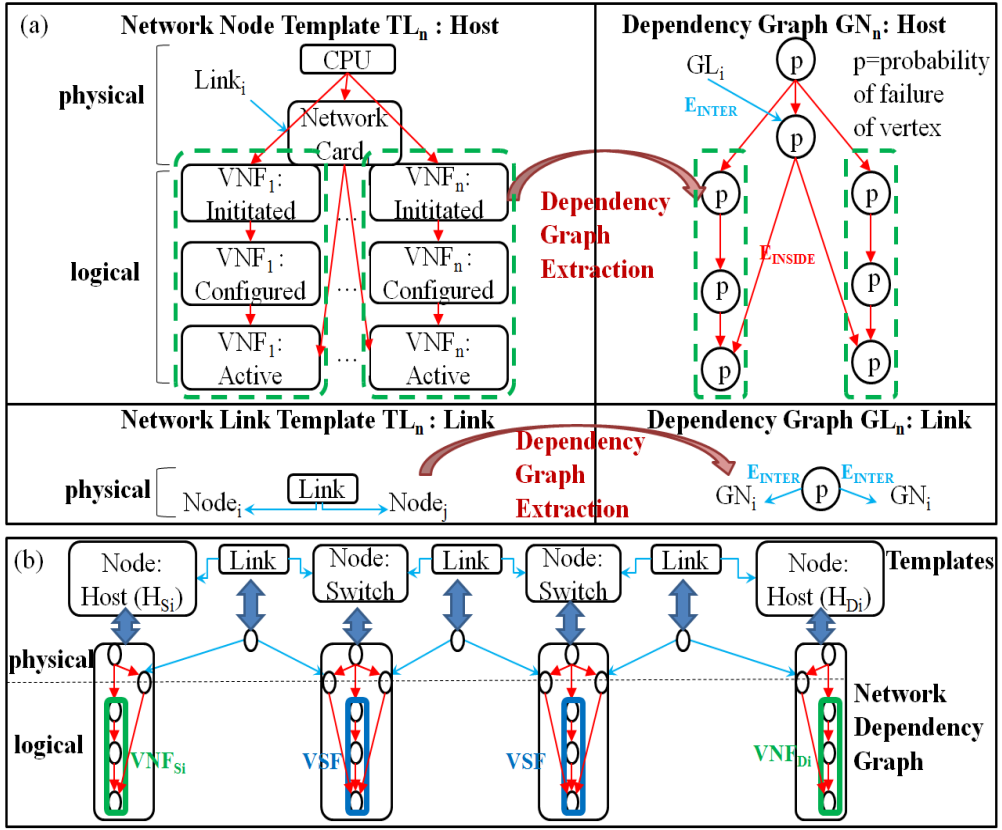
Topology-Aware Self-Modeling approach

Definition of a **set of adaptable fine-grained templates** to model the network dependencies **at physical and logical layers:**

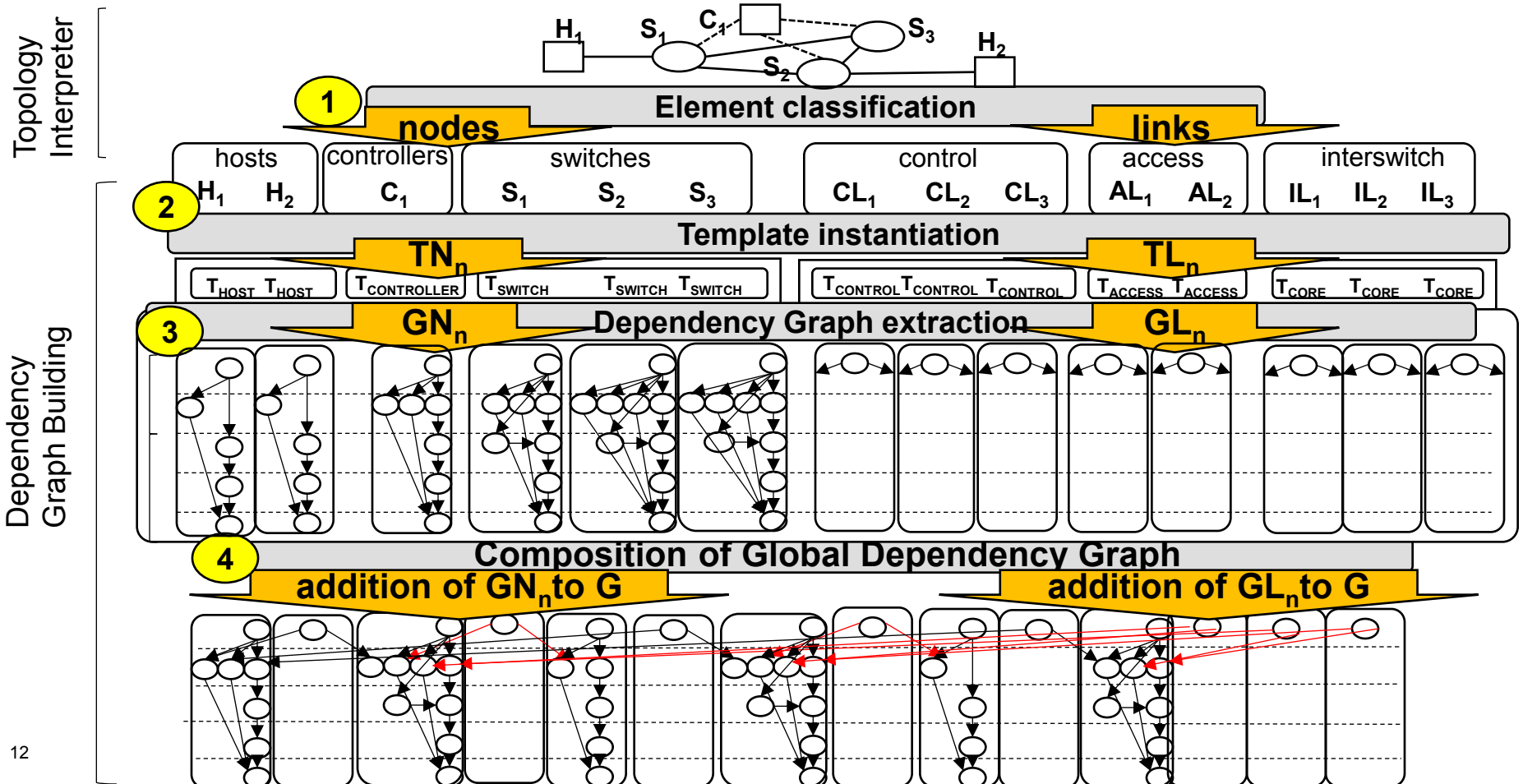
-**Inside the SDN nodes** (sub-components)

-**Among the SDN nodes** (network topology)

The **dependency graph of the network is automatically built** by combining the dependency graphs of the discovered network nodes and links



Topology-Aware Self-Modeling approach



Outline

Context

Proposals

Results

-Topology-Aware Self-Modeling

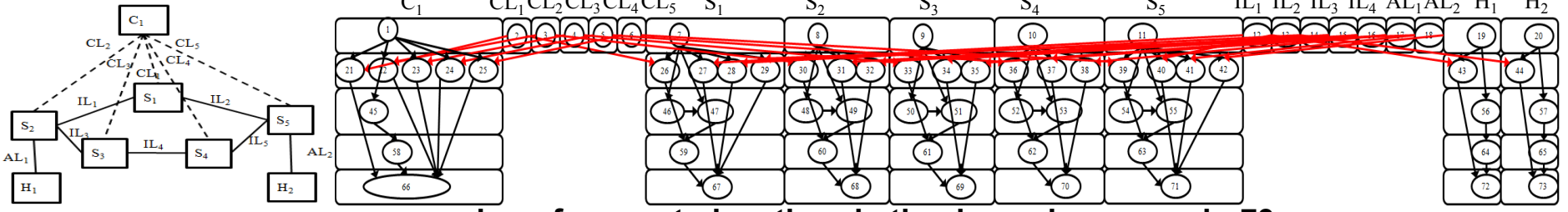
-Topology-Aware Root Cause Analysis

Conclusion and future lines

Results: Topology-Aware Self-Modeling

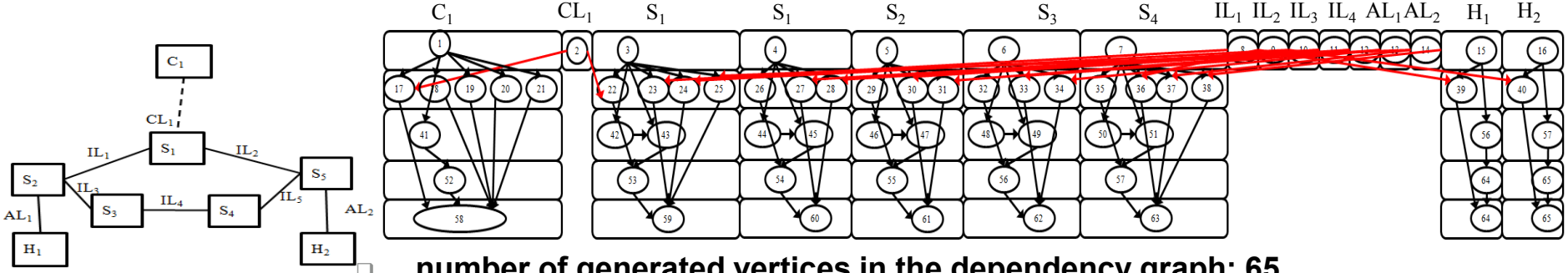
Self-Modeling Validation: dynamic network topologies and types of control

- **control**: out-of-band, **network topology**: ring (5 switches, 2 hosts)



- **number of generated vertices in the dependency graph: 73**

- **control**: in-band, **network topology**: ring (5 switches, 2 hosts)



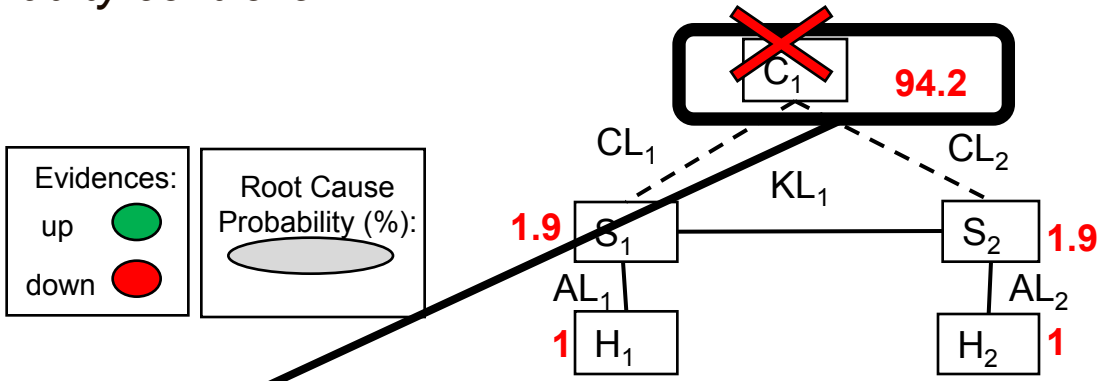
- **number of generated vertices in the dependency graph: 65**

Legend

C_i: controller | S_i: switch | CL_i: control link | IL_i: inter switch link | H_i: host | AL_i: access link

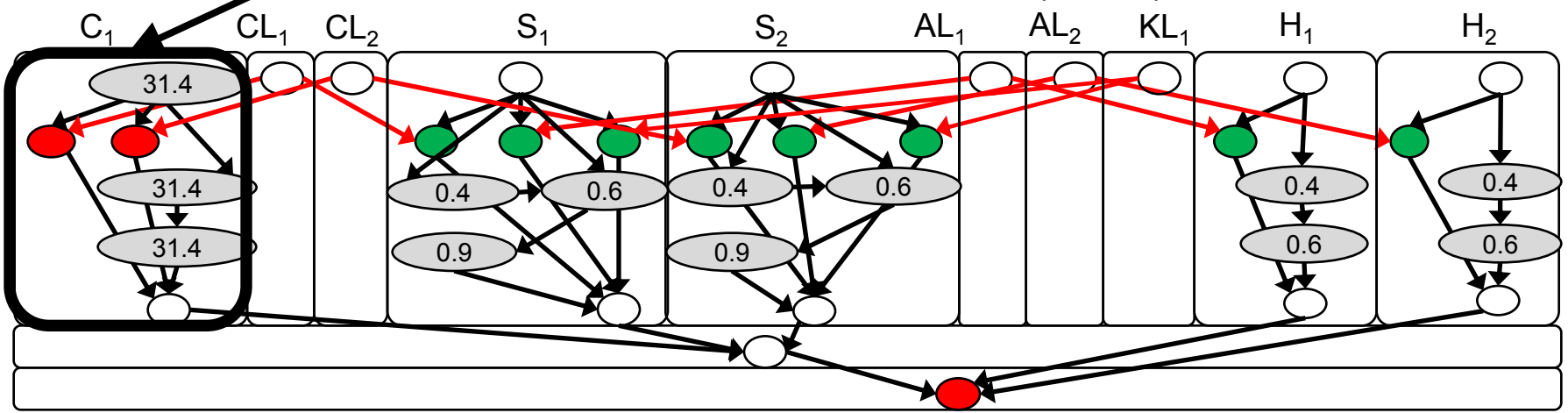
Results: Topology-Aware Root Cause Analysis

faulty controller



Root Cause node : C_1

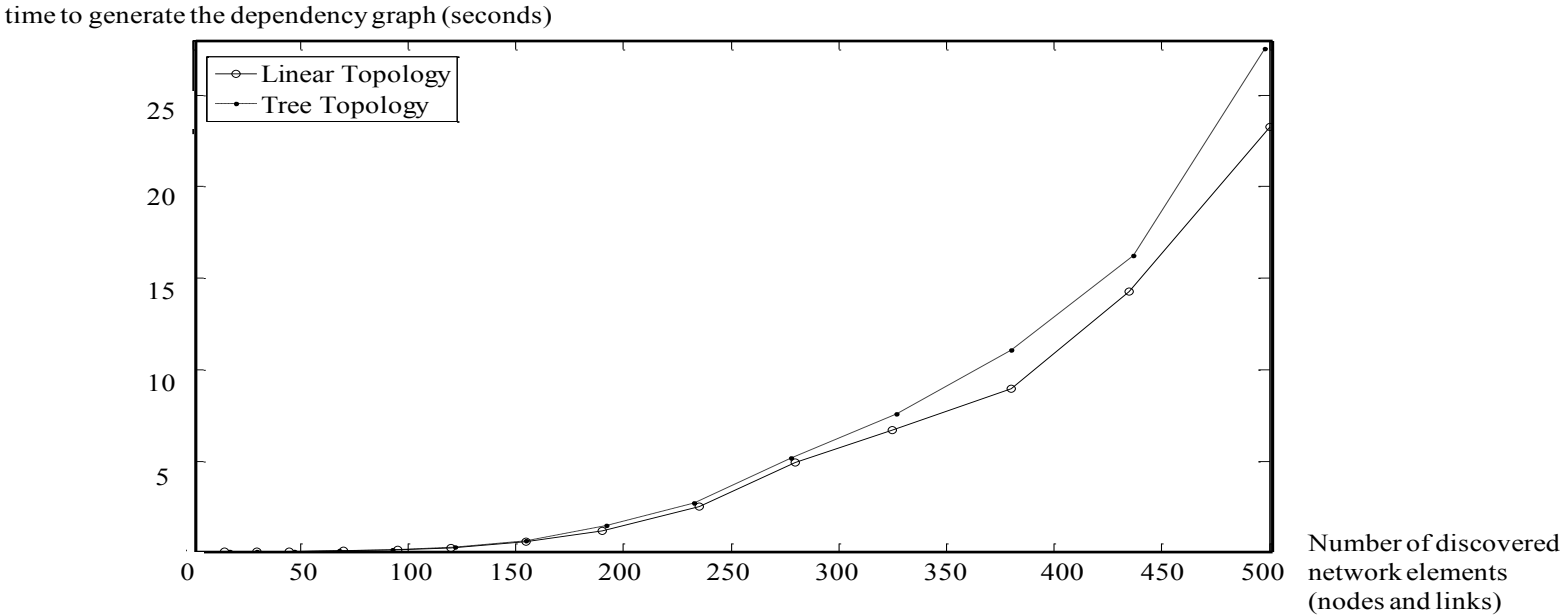
Node's subcomponents:
CPU problem (31.4%)
Controller APP not initiated (31.4%)
Controller APP not configured (31.4%)



Results: Self-Modeling Evaluation

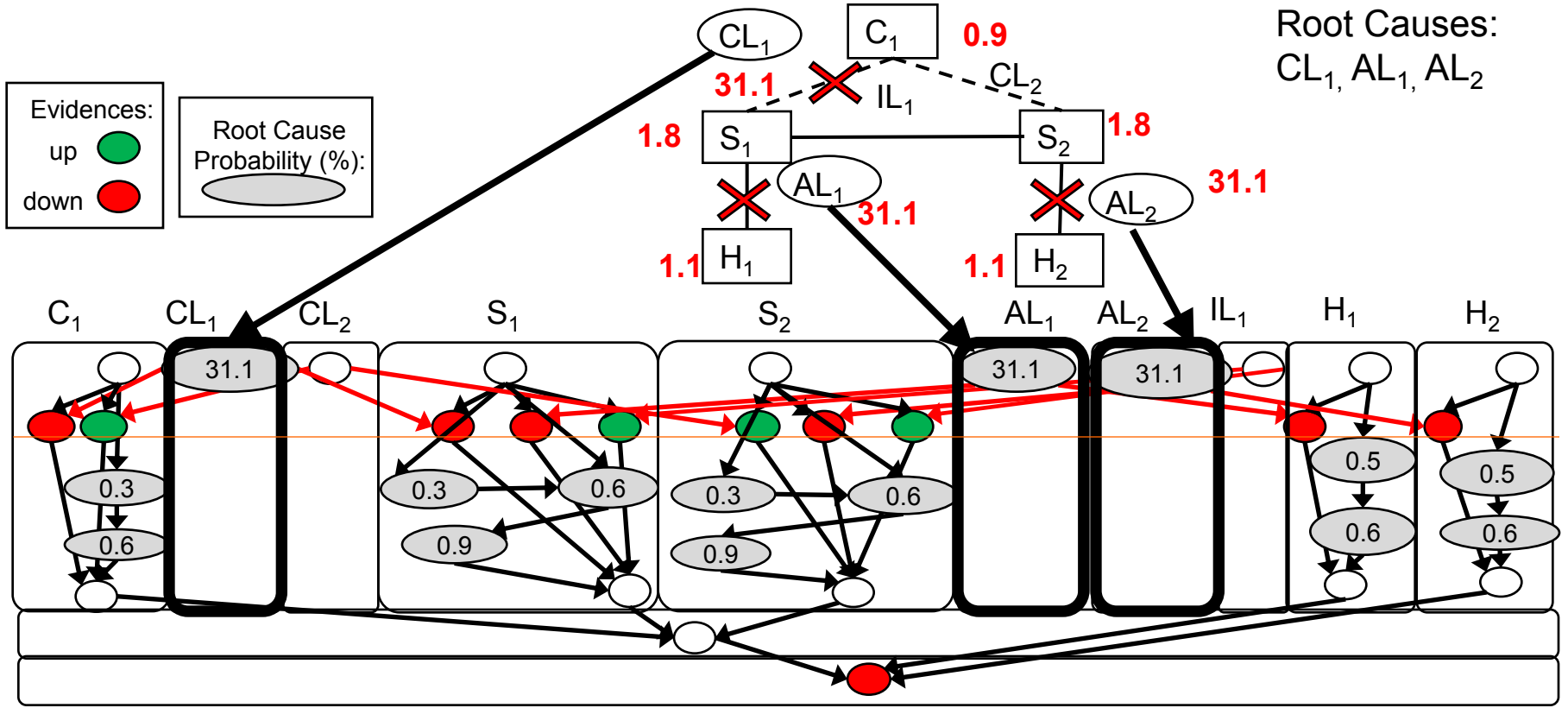
Performance

- Self-Modeling of tree, linear, clos-like, fat tree and ring topologies with in-band an out-of-band control
- Analysis of performance of the self-modeling algorithm as a function of the number of network elements discovered.
- Exponential trend in the growth of self-modeling time with the number of elements for linear and tree topologies (< 30 seconds for both cases).



Results: Topology-Aware Root Cause Analysis

simultaneous link failures in control and data plane



Outline

Context

Proposals

Results

Conclusion and future lines

Conclusion and future lines

Work done

Self-diagnosis framework to empower model-based diagnosis in SDN and NFV scenarios in a controller's domain. It utilizes a self-modeling approach based on a set of predefined fine-grained templates

Results

- Evaluation of scalability of self-modeling algorithm over different topologies until 500 network elements per controller's domain (<30 seconds)

Future work

- Extension of this Self-modeling mechanism to encompass different network topologies of different controller's domains
- Adoption of learning mechanisms for automatically generation of templates of new equipment added to the network

19

Thank you for your attention!

Any questions?