Doctoral Program in COMPUTER AND CONTROL ENGINEERING (32th cycle)

Doctoral Dissertation
Formal assurance of security policies in automated network orchestration (SDN/NFV)

PhD Candidate

Jalolliddin Yusupov

Advisors

Professor Riccardo Sisto, Polito

Professor Adlen Ksentini, EURECOM

Co-advisors

Guido Marchetto, Polito

Fulvio Valenza, Polito

May 28, 2020

Outline

- Introduction
- Contributions
  - Network function modeling
  - Formally verified network function placement
  - Automatic Firewall Configuration
- Conclusion

Research questions

Networks are evolving rapidly

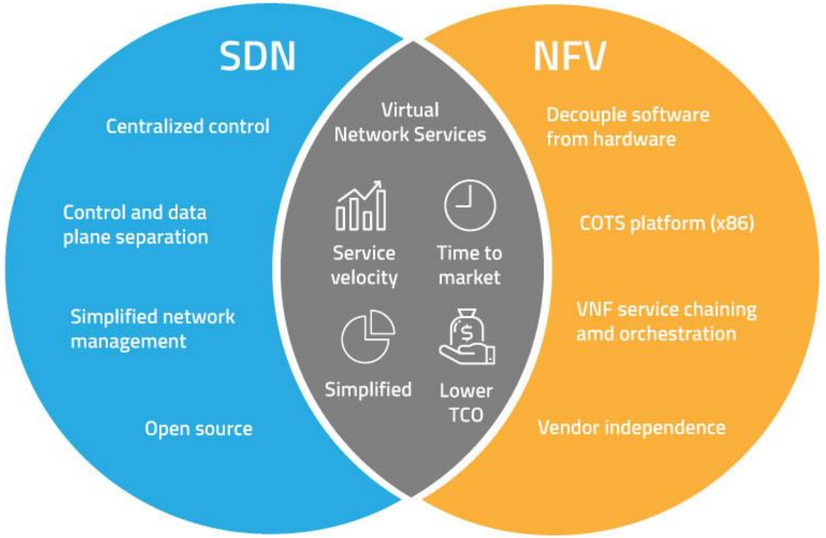
- need for more dynamic and automated network management
  - Software Defined Networking (SDN) and Network Function Virtualization (NFV)
    - undeniable great benefits: scalability, flexibility, efficiency …

However

- misconfigurations, security flaws, underutilization

How to increase **assurance** in the correct behavior of the network?
How to reduce **underutilization** of network resources during orchestration?
How to automate optimal **configuration** of network security functions?

Background

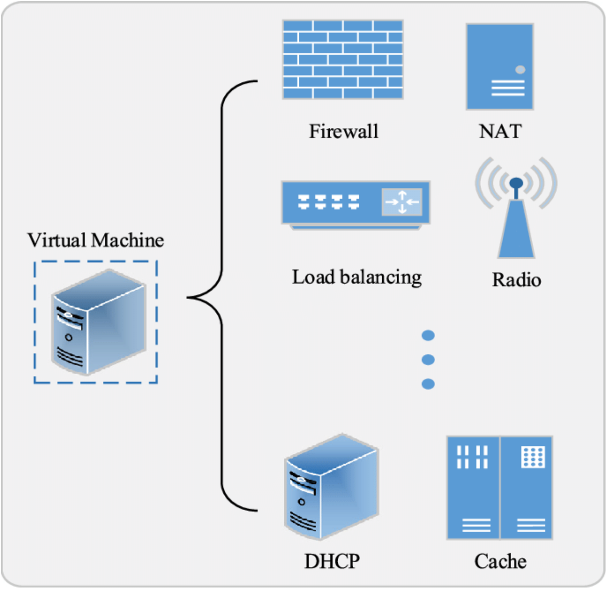


NFV and SDN

SDN is the idea of separating the control plane of a network from the data plane that forwards network traffic. NFV virtualizes network services and abstract them from dedicated hardware

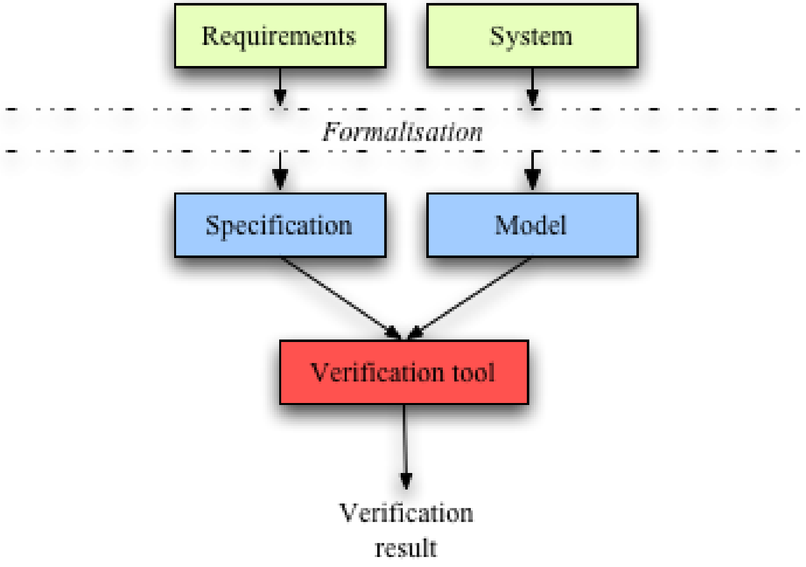
Virtual Network Function (VNF)

handles specific network functions that run on one or more virtual machines (VMs) or in containers



Formal verification

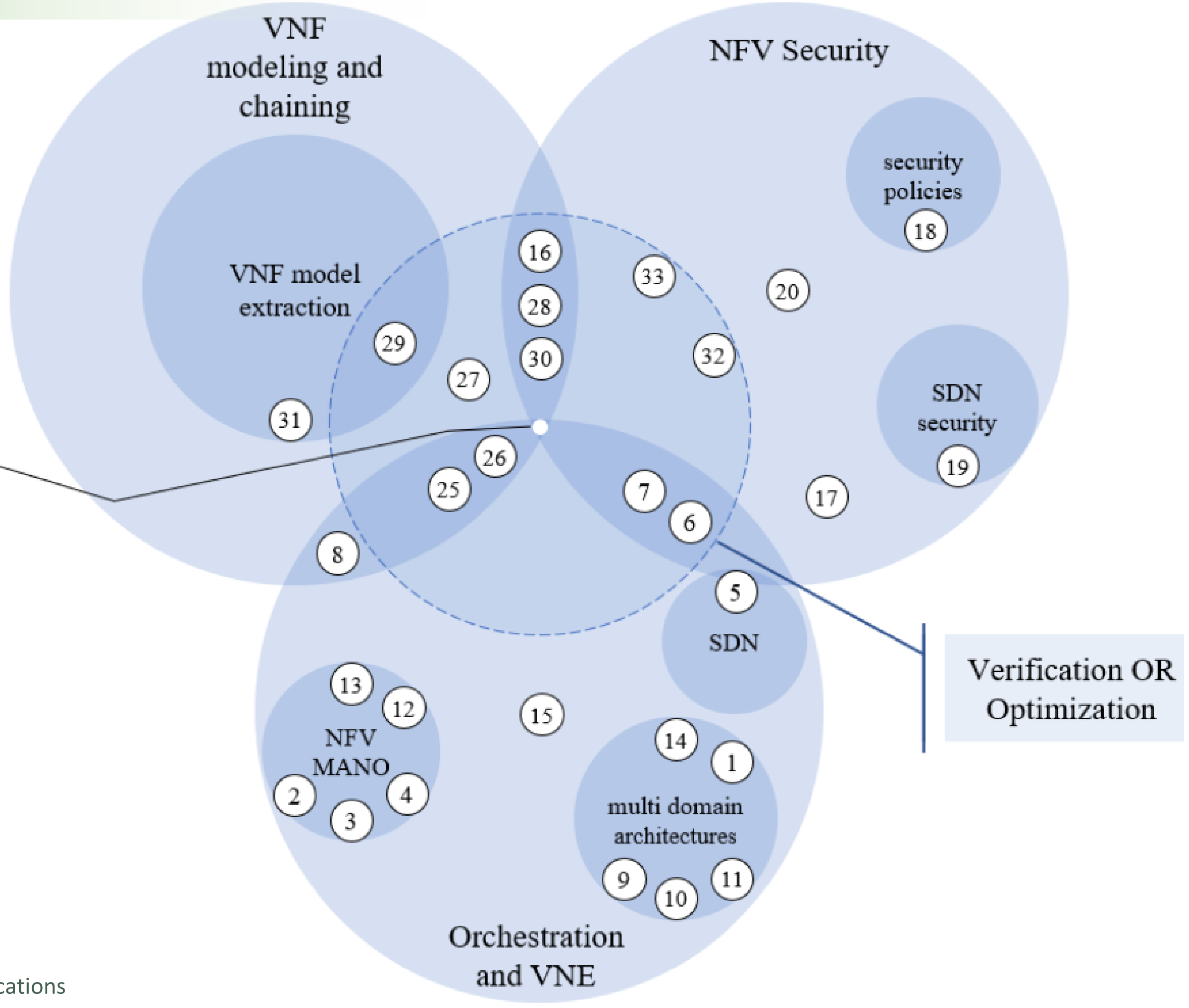
is the act of proving or disproving the correctness of an abstract model of the system with respect to a certain formal specification or property



POLITECNICO DI TORINO

State of the art

To the best of our knowledge, in NFV-based environment, the need for **orchestration and allocation of services**, giving at the same time **assurance** about a number of **safety** and **security-related** properties of the orchestrated virtual networks, remains an open problem to be addressed by the research community



Taxonomy of surveyed techniques with corresponding publications

Objectives

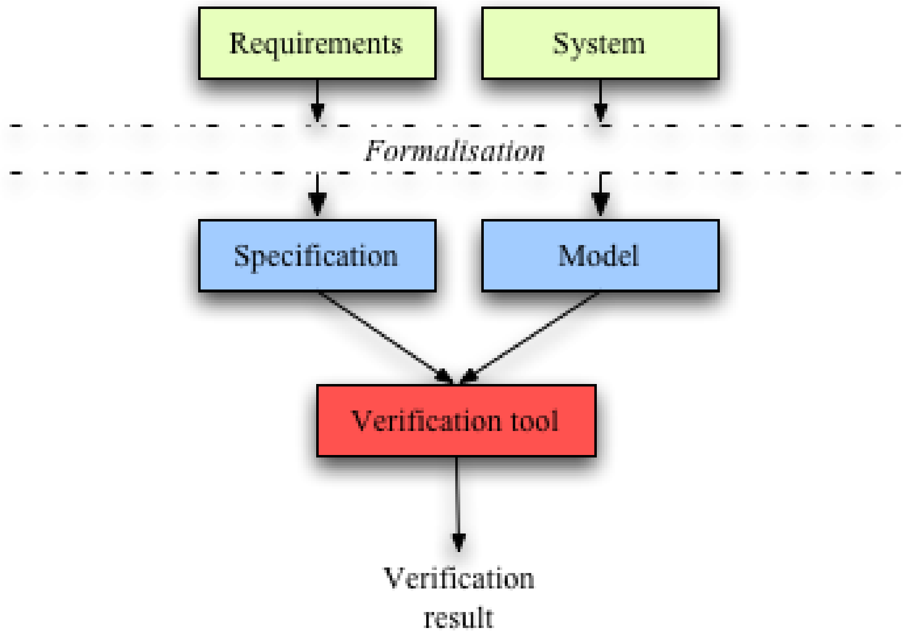
- Simplify the definition of a network function model
  - automatically translate into an abstract formal model for verification tools
- Provide an approach for allocation and formal verification
  - network policies are never violated
  - optimization is achieved
- Automatically and optimally configure multiple firewalls

Network function modeling for verification purposes

Network function modeling for verification purposes

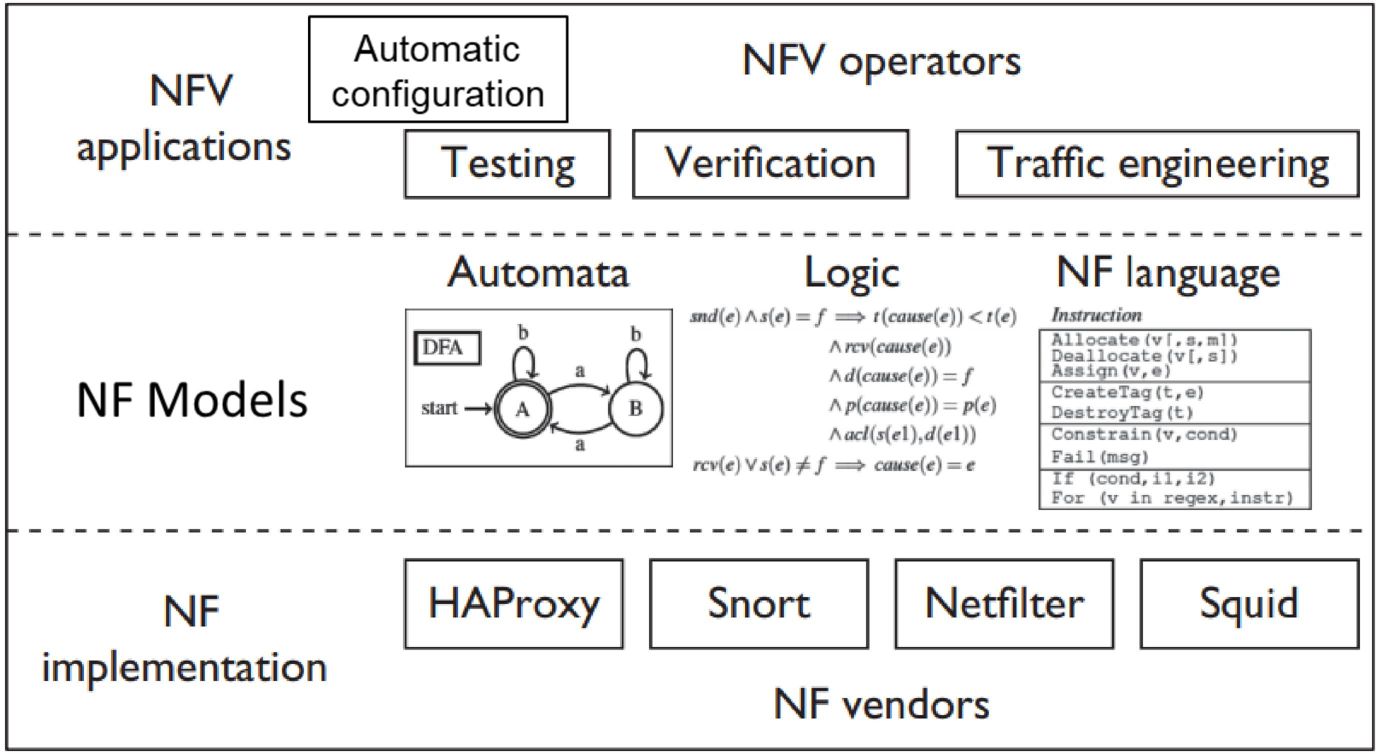
Problem statement

- complex VNF software
- network configuration errors
- need for networks' correctness, safety, and security
- the specific input languages of the verification tools
  - error-prone, time-consuming
  - outside the VNF developers' expertise
  - lack of unified translation pattern
- no standard and structured way of writing network function code
  - lack of specific libraries



POLITECNICO DI TORINO

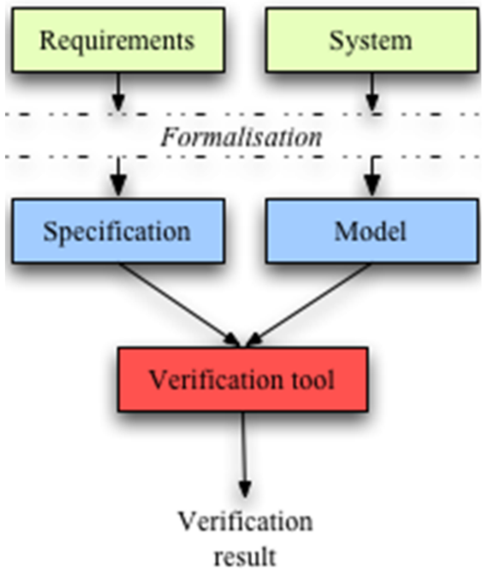
State of the art

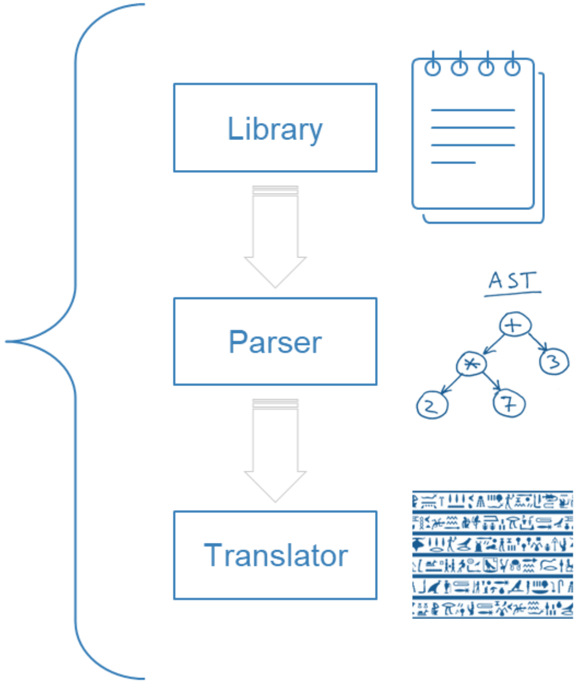


- automation tools based on formal methods accept input in specification languages
- difficult to use for people not experienced in formal methods
- error-prone, time-consuming, and often outside the VNF developers’ expertise

Contribution 1

- To simplify the definition of a NF model in a well-known language
- Flexibility to define the desired behavior for all NF



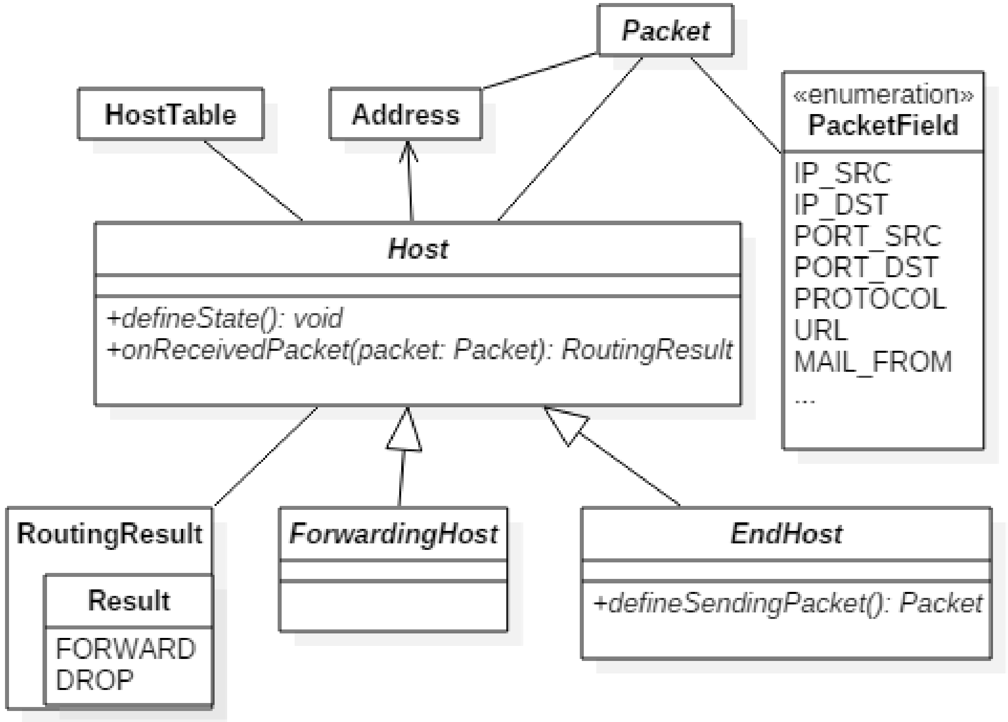


1 A Framework for User-Friendly Verification-Oriented VNF Modeling, 2017
G. Marchetto, R. Sisto, M. Virgilio and J. Yusupov. 41st Annual Computer Software and Applications Conference (COMPSAC)

2 A Framework for Verification-Oriented User-Friendly Network Function Modeling, 2019
G. Marchetto, R. Sisto, F. Valenza, J. Yusupov - IEEE Access

POLITECNICO DI TORINO

Library



- set of high-level operations for describing the network function's forwarding behavior

LIST OF SUPPORTED FEATURES OF JAVA LANGUAGE

Data types: int String boolean
Boolean operators: && || !
Comparison operators: == !=
Statements: return if if-else
Functions, Constructors, Constants, Variable Assignments

Parser

- analyzes the Java source code and extracts an abstract formal model

Tasks

- the identification of the instructions in the Java code that lead to a packet being sent through an interface
- the identification of the conditions (IF statements)

```
@Override
public RoutingResult onReceivedPacket(Packet packet) {
        if(Packet.match(packet.getProtocol(), Constants.POP3_REQUEST_PROTOCOL, Operator.EQUAL)) {
                return new RoutingResult(packet,Action.FORWARD,ForwardDirection.UPSTREAM);
        }
        if(Packet.match(packet.getProtocol(), Constants.POP3_RESPONSE_PROTOCOL, Operator.EQUAL) &&
                        !this.state.hostTableList.get("Blacklist").contains(packet.getMailSource())) {
                return new RoutingResult(packet,Action.FORWARD,ForwardDirection.UPSTREAM);
        }
        return new RoutingResult(packet,Action.DROP,ForwardDirection.UPSTREAM);
}
```

Java description of the behavior of the Antispam network function in response to a received packet

Parser

- analyzes the Java source code and extracts an abstract formal model

Tasks

- the identification of the instructions in the Java code that lead to a packet being sent through an interface
- the identification of the conditions (IF statements)

```
@Override
public RoutingResult onReceivedPacket(Packet packet) {
        if(Packet.match(packet.getProtocol(), Constants.POP3_REQUEST_PROTOCOL, Operator.EQUAL)) {
                return new RoutingResult(packet,Action.FORWARD,ForwardDirection.UPSTREAM);
        }
        if(Packet.match(packet.getProtocol(), Constants.POP3_RESPONSE_PROTOCOL, Operator.EQUAL) &&
                        !this.state.hostTableList.get("Blacklist").contains(packet.getMailSource())) {
                return new RoutingResult(packet,Action.FORWARD,ForwardDirection.UPSTREAM);
        }
        return new RoutingResult(packet,Action.DROP,ForwardDirection.UPSTREAM);
}
```

Java description of the behavior of the Antispam network function in response to a received packet

POLITECNICO DI TORINO

Translation pattern

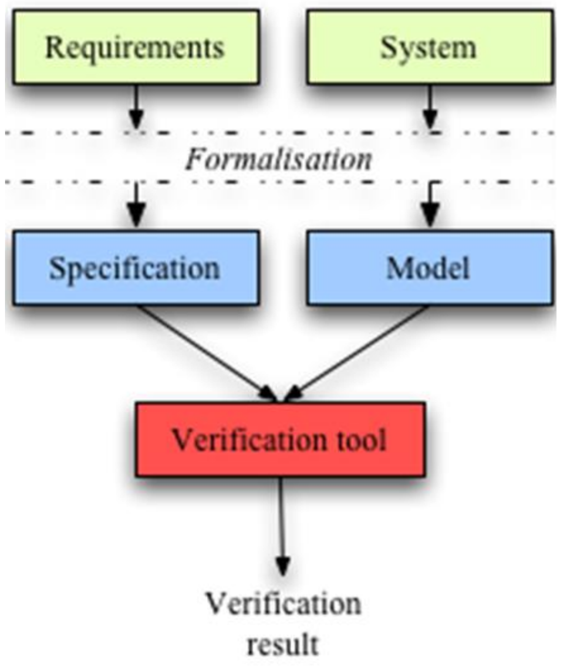
- takes an input from the parser and converts them into first-order-logic (FOL) formulas - Boolean constraints

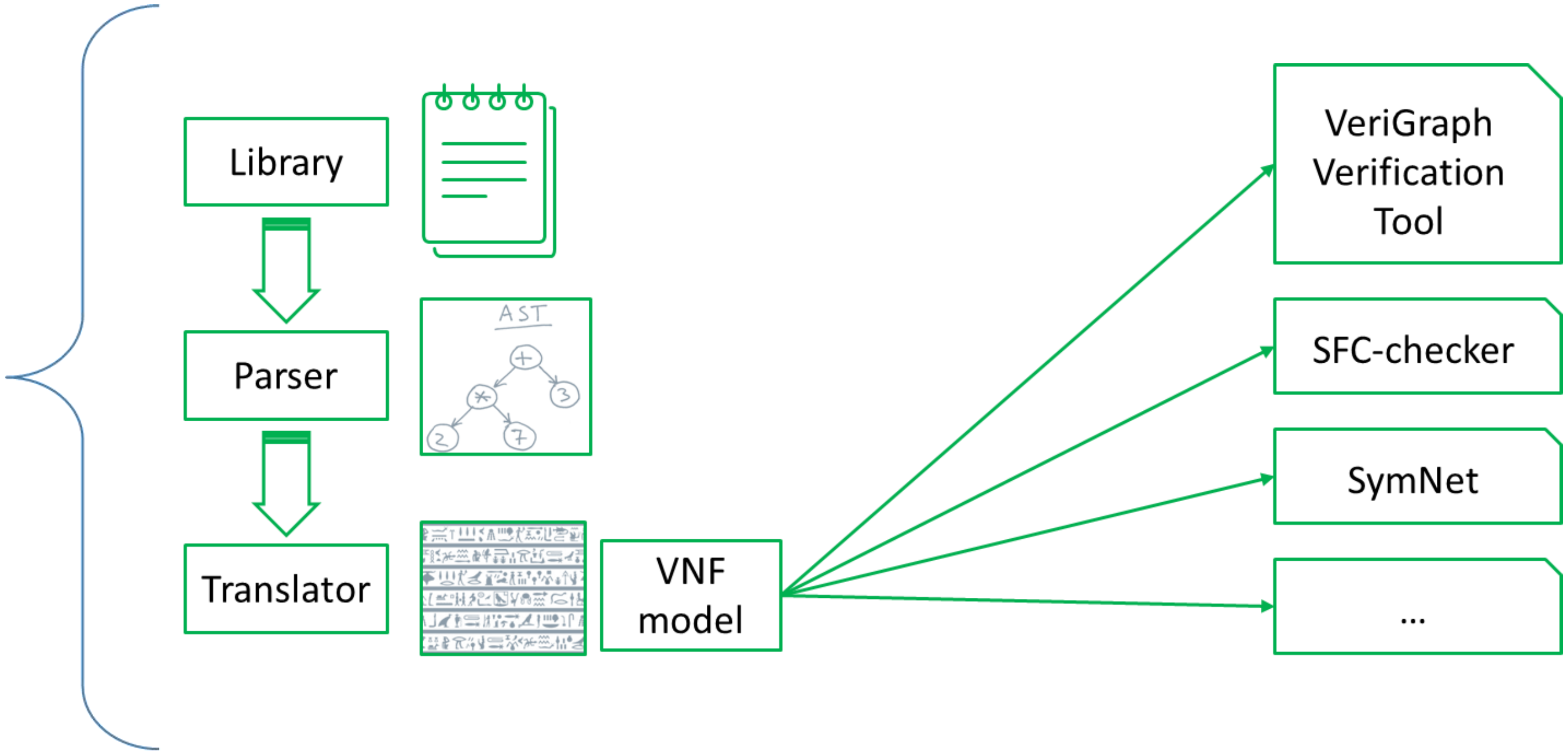
TRANSLATOR OUTPUT FORMAT FOR THE ANTISPAM VNF.

1	send(Antispam, n_0, p, t_0) $\rightarrow$ p.proto == POP_REQ || p.proto == POP_RESP
2	send(Antispam, n_0, p, t_0) $\rightarrow$ nodeHasAddr(Antispam, p.src)
3	send(Antispam, n_0, p, t_0) && p.proto(POP3_RESP) $\rightarrow$ ($\exists$ n_l, t_l : (recv(n_l, Antispam, p, t_l) && t_l < t_0)) && !isInBlackList(p.emailFrom)
4	send(Antispam, n_0, p, t_0) && p.proto(POP3_REQ) $\rightarrow$ ($\exists$ n_l, t_l : (recv(n_l, Antispam, p, t_l) && t_l < t_0))

POLITECNICO DI TORINO

Results





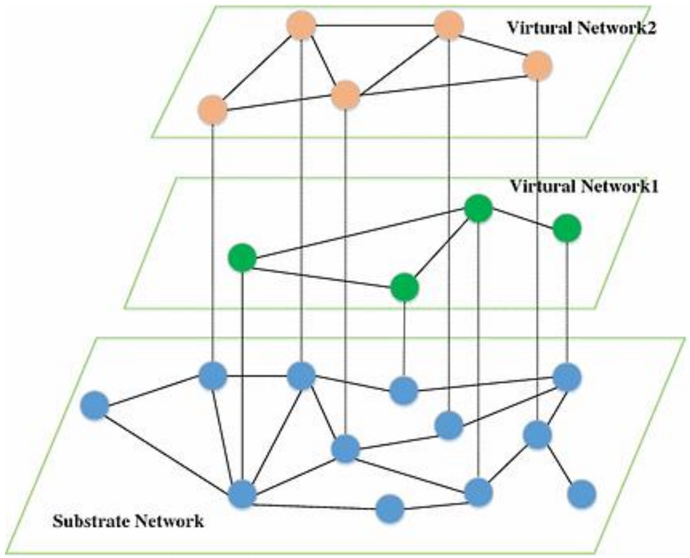
3 A VNF modeling approach for verification purposes, 2019

G. Marchetto, R. Sisto, M. Virgilio, J. Yusupov, INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING

Formally verified network function placement

Formally verified network function placement
Background

- **Network Function Placement or Virtual Network Embedding (VNE)** is the problem of finding an optimum mapping of virtual nodes and links onto a given physical substrate network



Boolean (SAT)isfiability Problem

Given:

- A Boolean Formula

Question:

- Is there an assignment of truth values to the Boolean variables such that the formula holds true?

$$a \vee (\neg a \wedge b)$$

SATISFIABLE
a=true, b=true

Background

Maximum Satisfiability Problem (MaxSAT)

n Boolean variables $x_1, \dots, x_n$

m clauses $C_1, \dots, C_m$ with weights $w_j \geq 0$

– each clause is a disjunction of literals, e.g. $C_1 = x_1 \vee x_2 \vee \bar{x}_3$

- Find an assignment of the Boolean variables that maximizes the total weight of the satisfied clauses or equivalently minimize the total weight of the falsified clauses

MaxSAT + Theory, solvers = MaxSMT

Maximum Satisfiability Modulo Theories Problem (MaxSMT)

- First Order Logic (FOL)
  - theory of arrays
  - free function symbols
  - linear integer arithmetic
  - theory of equality
  - difference logic
  - ...

IP vs MaxSMT

Problem statement:

- Formal Verified Solution
- The existing literature formulates the a problem using Integer Programming (IP) formulation
  - constraints over binary, integer, or real variables
    - restriction does not apply for the MaxSMT formulation
      - which allows us to model the problem more directly and using very **expressive constraints**: forwarding behaviors…
      - to check that network fulfill selected properties with 100% certainty and deliver **formally verified** placement plan

$$\max z = 8x_1 + 11x_2 + 6x_3 + 4x_4$$

s.t.

$$\begin{aligned} 5x_1 + 7x_2 + 0x_3 + 3x_4 &\leq 14 \\ 8x_1 + 0x_2 + 4x_3 + 4x_4 &\leq 12 \\ 2x_1 + 10x_2 + 6x_3 + 4x_4 &\leq 15 \\ x_1, \quad x_2, \quad x_3, \quad x_4 &\in \{0,1\} \end{aligned}$$

State of the art:

- Joint Optimization and Verification can be encoded to IP instance, but…
  - the combinatorial encoding turned out to be impractical[3]
    - in most cases and authors were often not able to generate MaxSMT encodings for many of the instances
  - model generator run out of memory when dealing with larger instances[4]
    - the huge amount of produced clauses required by this encoding

Drawbacks of the Integer Programming

Mixed Integer Quadratically Constrained Programming (MIQCP)

- Placement Constraints

$$\forall v \in V : \quad \sum_{u \in U} m_{u,v} \cdot p(u) \leq c(v)$$

- Path Related Constraints
  - latency requirements

$$\forall (a, a') \in A_{pairs} :$$

$$\sum_{\substack{(v,v') \in E, x,y \in V, \\ (u,u') \in paths(a,a')}} e_{v,v',x,y,u,u'} \cdot l(v, v') \leq l_{req}(a, a')$$

$$\forall (v, v') \in E :$$

  - bandwidth requirements

$$\sum_{(u,u') \in U_{pairs}, \forall x,y \in V} e_{v,v',x,y,u,u'} \cdot d_{req}(u, u') \leq d(v, v')$$

*S. Spinoso, M. Virgilio, W. John, A. Manzalini, G. Marchetto, and R. Sisto,
"Formal Verification of Virtual Network Function Graphs in an SP-DevOps Context," in ESOCC 2015, Taormina, Italy, September 15-17, 2015. Proceedings, 2015, pp. 253–262.

MaxSMT Formulation

Inspired by VeriGraph*

- VeriGraph (z3): hard constraints
- VerifOO (z3Opt): hard constraints

$$\sum_{\forall j | n_j^s \in N^s} to\_int(x_{ij}) = 1 \quad \Big| \quad y_j \implies \bigvee_i x_{ij} \quad \Big| \quad \left(\sum_{\forall i | n_i^v \uparrow n_j^s} storage(n_i^v) * to\_int(x_{ij}) \right) \leq storage(n_j^s) * to\_int(y_j)$$

- VerifOO (z3Opt): soft constraints

$$addSoft(\neg y_1, 1) \qquad addSoft(\neg y_2, 1) \quad \Big| \quad addSoft((route(n_{i+1}^v, l_{jk}^s) \implies x_{ij} \wedge x_{(i+1)k}), -latency(l_{jk}^s))$$

What to verify?
Forwarding behavior

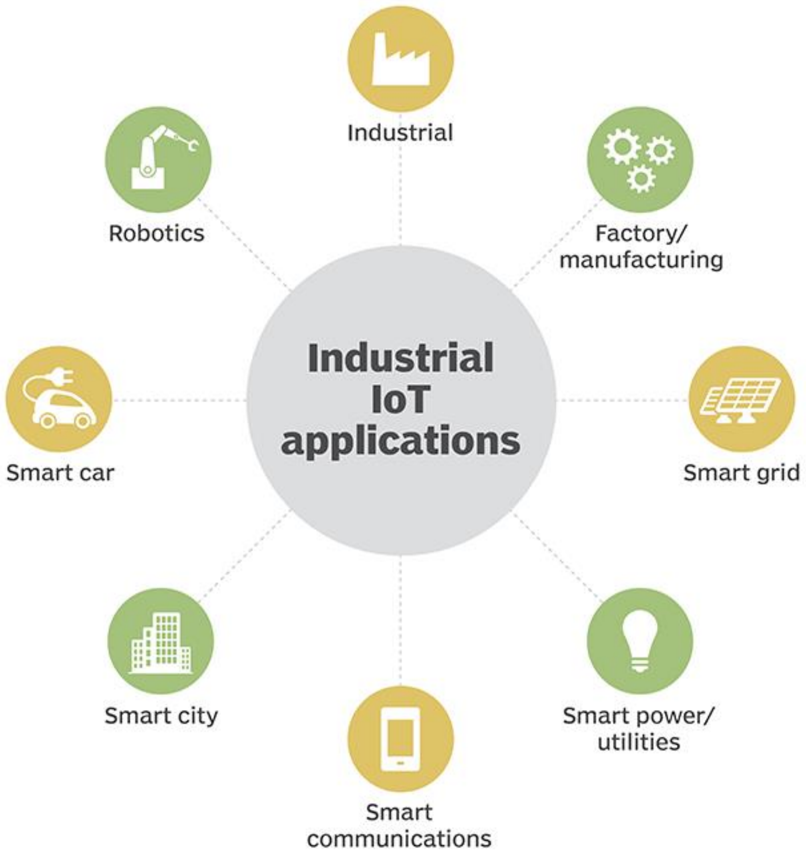
The traffic generated from the source ***s***, must/mustn't reach the destination ***d***

- Reachability: $\exists(n_0, p_0) \mid recv(n_0, dest, p_0) \wedge p_0.origin == src$
- Isolation: $\forall(n_0, p_0) \mid recv(n_0, dest, p_0) \implies p_0.origin \neq src$
  $\exists(n_1, p_1) \mid send(src, n_1, p_1) \wedge nodeHasAddress(dest, p_1.dest)$

```
<PropertyDefinition>
        <Property graph="0" name="IsolationProperty" src="nodeA" dst="nodeB" src_port="
            ↪ 5000" dst_port="80">
            <HTTPDefinition url="polito.it" body="weapons"/>
        </Property>
        <Property graph="0" name="ReachabilityProperty" src="nodeC" dst="nodeD"
            ↪ src_port="3000" dst_port="110">
            <POP3Definition email_from="polito" body="weapons"/>
        </Property>
</PropertyDefinition>
```

POLITECNICO DI TORINO

Application I
Formally Verified Latency-aware VNF Placement



IIoT (Industrial Internet of Things or Industrial IoT)

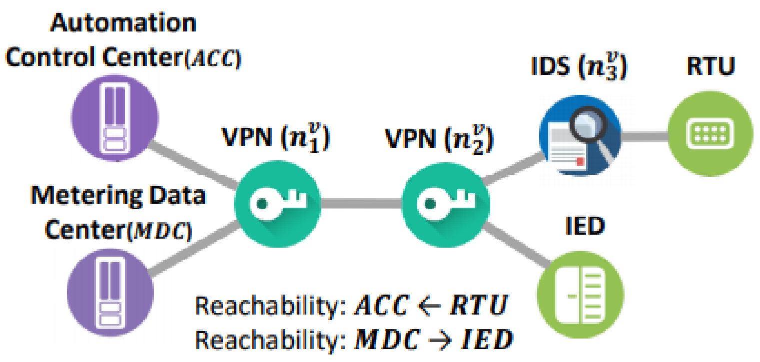
- industrial devices: interconnected sensors, instruments
  - that are now equipped with the capabilities to send data

Problem:

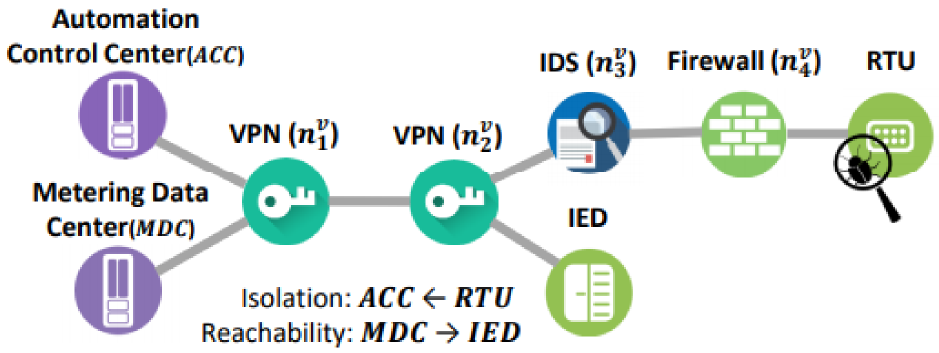
- Smart Grid is safety-critical, very complex and vulnerable to attacks or cyber threats
  - **Causes:**
    - the absence or weakness of security function
    - the misconfiguration of Virtual Network Functions (VNFs)
  - **Solutions:**
    - countermeasures dynamically placed to mitigate security threats (VNE)
    - detect misconfiguration of VNFs with simulation/testing or formal verification techniques

POLITECNICO DI TORINO

Smart Grid Use Case



Original Service Graph (a)



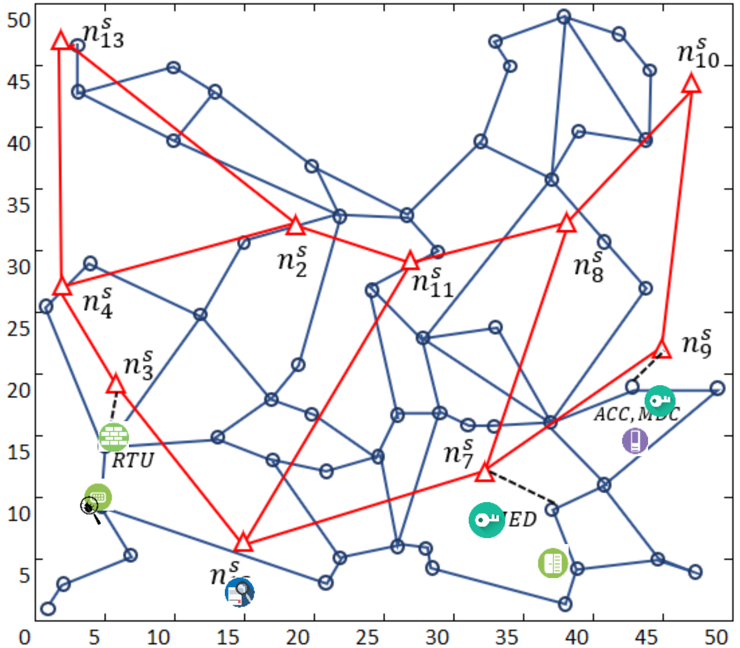
Updated Service Graph (b)

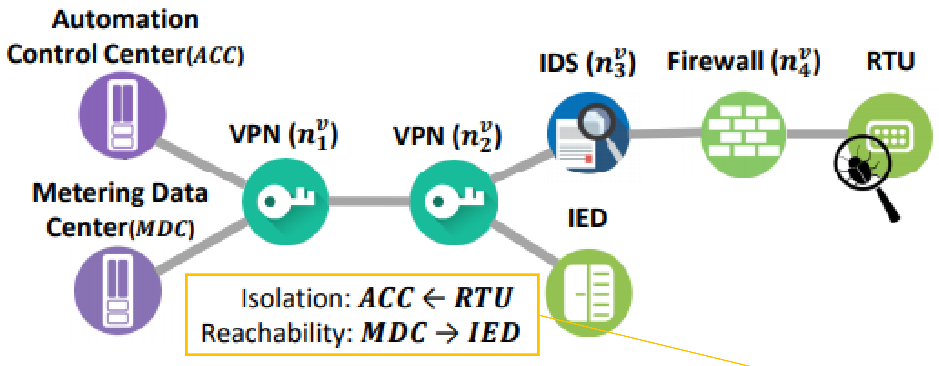
Configurations:

- VPN access n_v^1: IP address of the VPN exit n_v^2 gateway
- VPN exit n_v^2 : IP address of the VPN access n_v^1 gateway
- IDS n_v^3 : not allowed function code {43}
- Firewall n_v^4 : {src:RTU dst:ACC sport:$*$ dport:$*$ proto:$*$}
- ACC: generates a packet with a function code different from {43}

POLITECNICO DI TORINO

Smart Grid Use Case





Updated Service Graph (b)

SAT

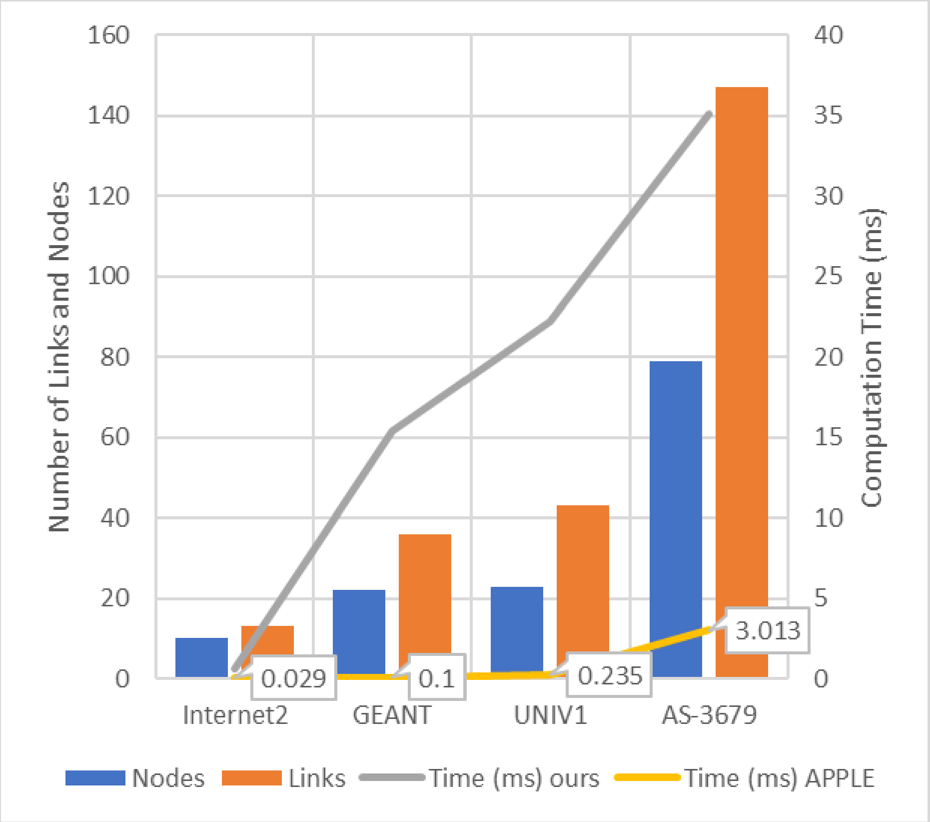
$x_{1,9}, x_{2,7}, x_{3,12}, x_{4,3}, y_3, y_7, y_9, y_{12}$

Configurations:

- VPN access n_v^1: IP address of the VPN exit n_v^2 gateway
- VPN exit n_v^2 : IP address of the VPN access n_v^1 gateway
- IDS n_v^3 : not allowed function code {43}
- Firewall n_v^4 : {src:RTU dst:ACC sport:∗ dport:∗ proto:∗}
- ACC: generates a packet with a function code different from {43}

Discussion and Results

- Model implementation
  - Z3[4] theorem prover from Microsoft Research



Topology	Nodes	Links	Time (O+V)	Time (O)[12]
Internet2[21]	10	13	0.6	0.029
GEANT[21]	22	36	15.4	0.1
UNIV1[31]	23	43	22.2	0.235
AS-3679[32]	79	147	35.1	3.013

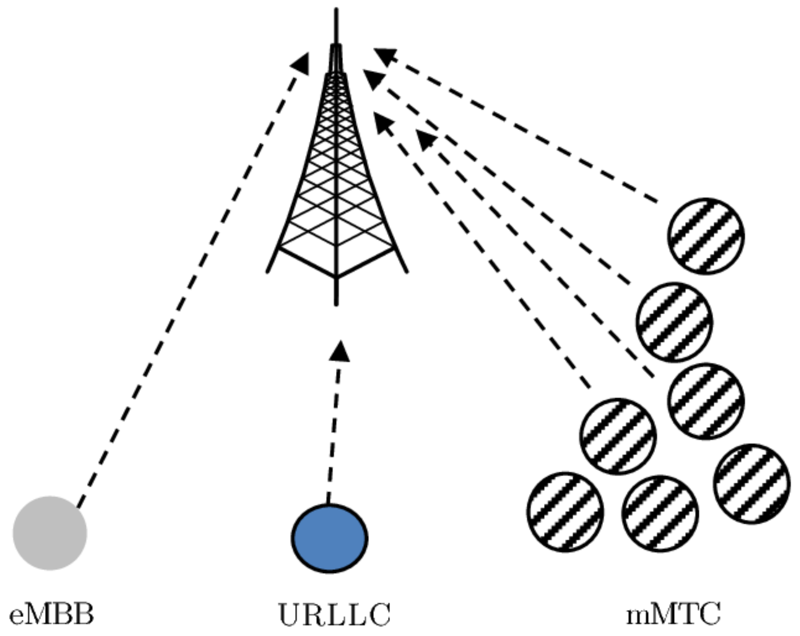
5 Formally verified latency-aware VNF placement in industrial Internet of things
G. Marchetto, R. Sisto, J. Yusupov, A. Ksentini, 14th International Workshop on Factory Communication Systems, WFCS 2018

6 A Formal Approach to Verify Connectivity and Optimize VNF Placement in Industrial Networks.
Under (third) review. IEEE Transactions on Industrial Informatics

POLITECNICO DI TORINO

Application II
Efficient 5G RAN Functional Split
Background

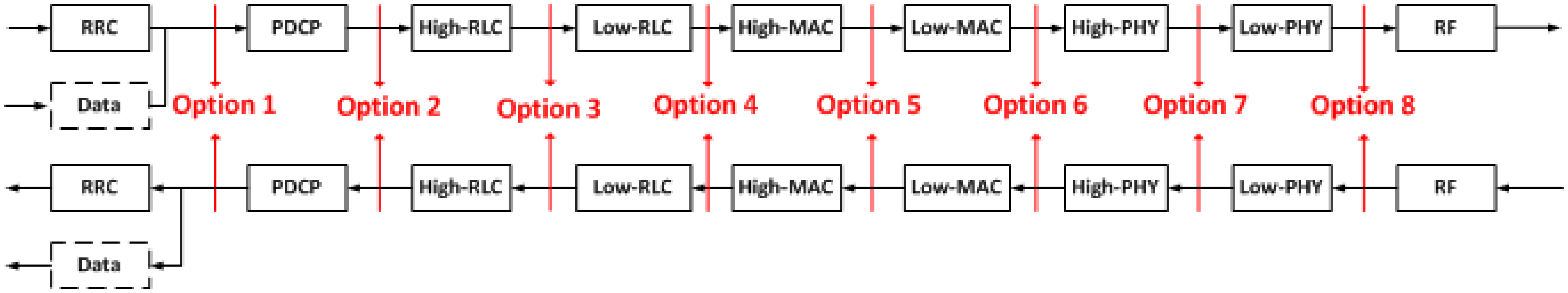
- Maximum latency and capacity required by emerging 5G services (slices)
- Network slicing
  - logical autonomous networks on top of a common infrastructure
    - eMBB - enhanced Mobile Broadband
    - mMTC - massive Machine Type Communications
    - URLLC - Ultra-Reliable and Low Latency Communications
- The logical architecture of next-generation RAN (3GPP)
  - **Central Unit (CU):** logical node that includes
    - Transfer of user data
    - Mobility control
    - Radio access network sharing
    - Positioning
    - …
  - **Distributed Unit (DU):**
  - **Remote Radio Unit (RU):**
    - depends on the **functional split option**



POLITECNICO DI TORINO

Problem statement

- RAN functional split implications/trade-offs
  - service requirements (e.g. latency)
  - fully distributed or fully centralized
  - increased total cost of ownership
  - increased power consumption
  - capability integration
  - …
    - they determine the placement of nodes and distance between



- Exists a need to evaluate the impact of the RAN functional decomposition

Efficient 5G RAN Functional Split

state of the art:

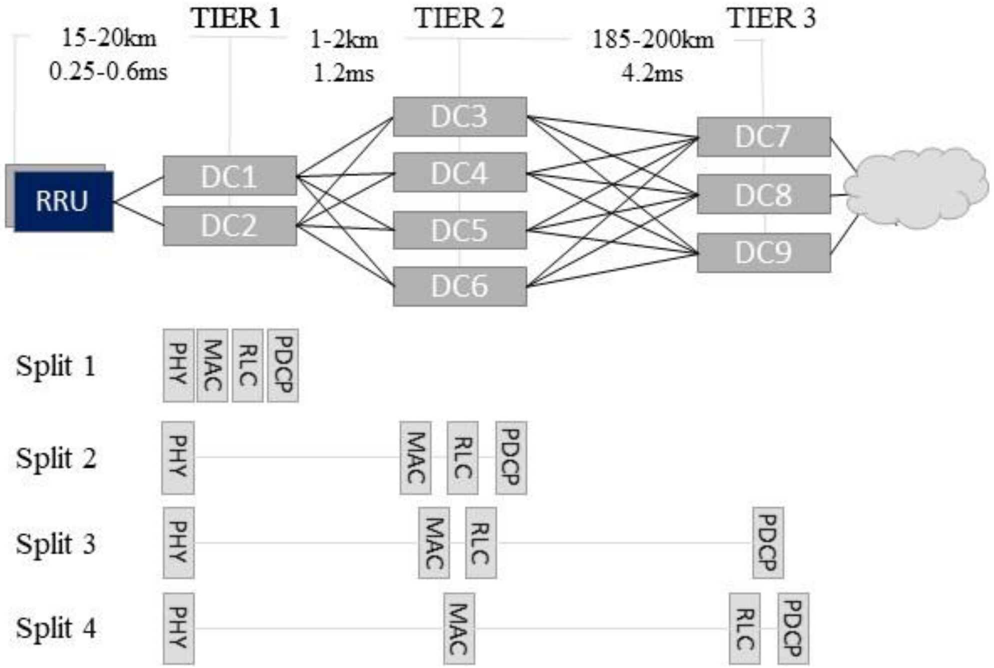
- focus on fully distributed or fully centralized baseband
- an optimal functional split depending on types of slices

objectives:

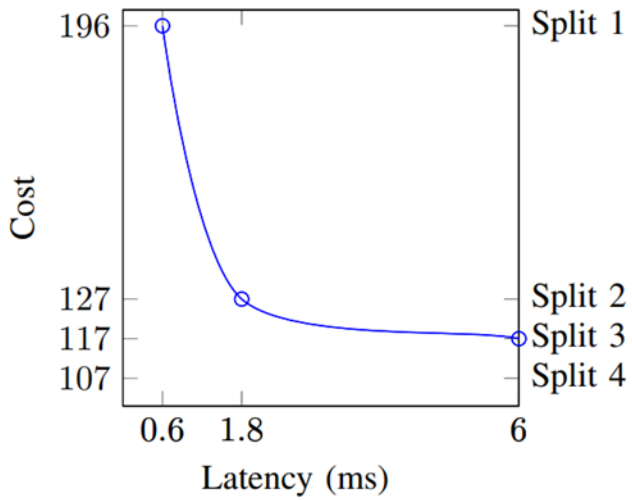
- exploiting the most suitable functional splits
- apply to different use cases and deployment scenarios
  - low-latency and high bandwidth requirements
- assurance
  - function chain correctly implements the reachability policies

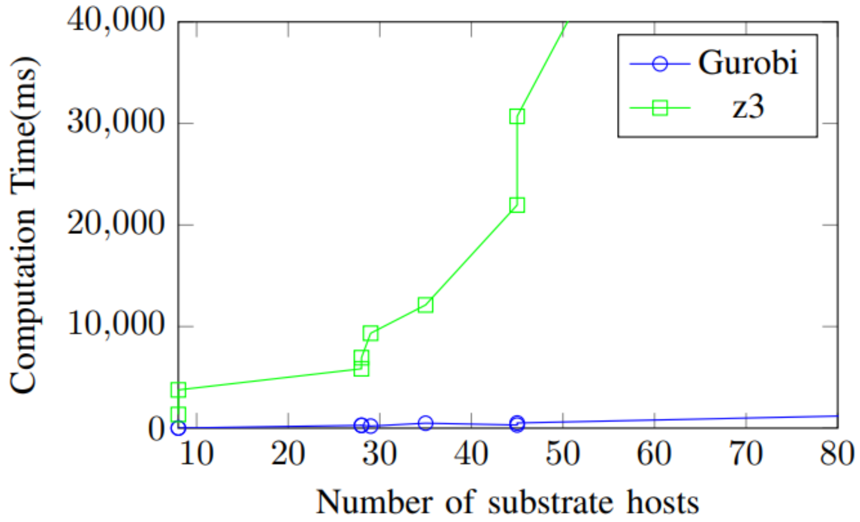
contributions:

- solve VNE problem based on slice requirements
- different traffic classes targeting different objectives
  - focusing on functional split of individual radio functions



Results



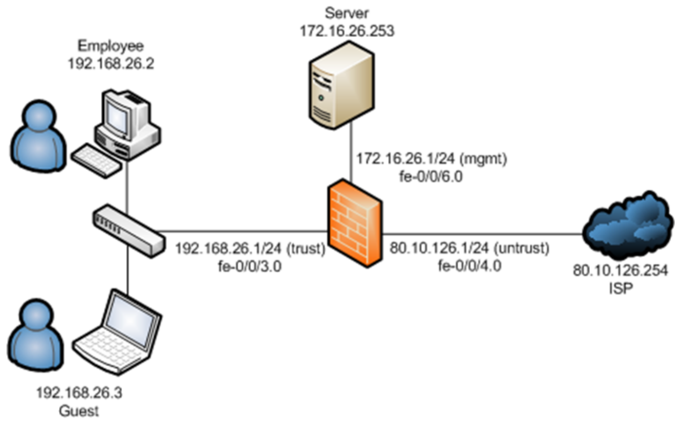


Multi-Objective Funciton Splitting and Placement of Network Slices in 5G Mobile Networks

G. Marchetto, R. Sisto, J. Yusupov, A. Ksentini, IEEE Conference on Standards for Communications and Networking (CSCN), 2018

Automatic Firewall Configuration

VErified REFinement and Optimized Orchestration



Allow all traffic from **Trusted** zone to the **Management office**

Deny traffic from untrusted zone, allow outbound traffic on port 80

Automatic rule generation

		Interface							
State	*Prtcl*	*FWi*	*Ethj*	*In/out*	[illegible]	*Src. Port*	[illegible]	*Dest. Port*	*Decision*
Response	TCP	FW1	Serial 0	out	[illegible]	80	[illegible]	Any	Accept
Response	TCP	FW2	Eth1	in	[illegible]	80	[illegible]	Any	Accept
Response	UDP	FW2	Eth1	in	[illegible]	53	[illegible]	Any	Accept
Response	TCP	FW1	Serial 0	out	[illegible]	25	[illegible]	Any	Accept
Response	UDP	FW1	Serial 0	out	[illegible]	53	[illegible]	Any	Accept
Response	TCP	FW2	Eth1	in	[illegible]	20	[illegible]	Any	Accept
Response	TCP	FW2	Eth1	in	[illegible]	21	[illegible]	Any	Accept
Response	TCP	FW2	Eth1	in	[illegible]	25	[illegible]	Any	Accept

state of the art:

- **a verified orchestration of network services**
- related works consider service graphs consisting of firewalls only
- high level network specifications into low level network configuration
  - lack of optimality in refinement process
  - translation is manual
- problems are typically detected only at runtime

goal: full automation, optimization and formal assurance

VerifOO -> VerefOO

- VerifOO (z3Opt): hard constraints

$$\sum_{\forall j | n_j^s \in N^s} to\_int(x_{ij}) = 1 \quad \Big| \quad y_j \implies \bigvee_i x_{ij} \quad \Big| \quad \left(\sum_{\forall i | n_i^v \uparrow n_j^s} storage(n_i^v) * to\_int(x_{ij}) \right) \leq storage(n_j^s) * to\_int(y_j)$$

- VerifOO (z3Opt): soft constraints

$$addSoft(\neg y_1, 1) \qquad addSoft((route(n_{i+1}^v, l_{jk}^s) \implies x_{ij} \wedge x_{(i+1)k}),$$
$$addSoft(\neg y_2, 1) \qquad -latency(l_{jk}^s))$$

- VerefOO

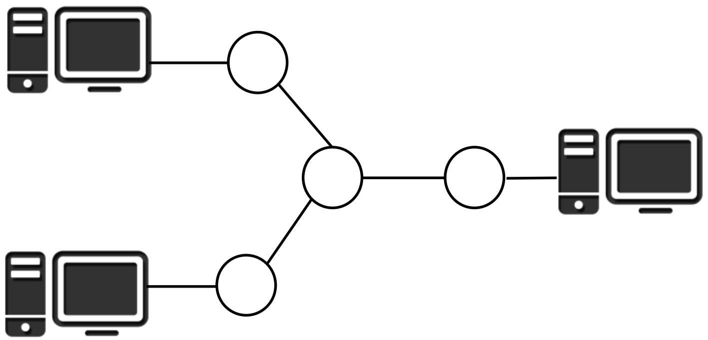
$$\forall \{n0, n1, p0\} :$$
$$recv(n0, n1, p0) \implies$$
$$p0.src\_port.start > 0 \wedge p0.src\_port.end < MAX\_PORT$$
$$\wedge\, p0.dst\_port.start > 0 \wedge p0.dst\_port.end < MAX\_PORT$$

POLITECNICO DI TORINO

Allocation Graph

Middle-Level Security Policies

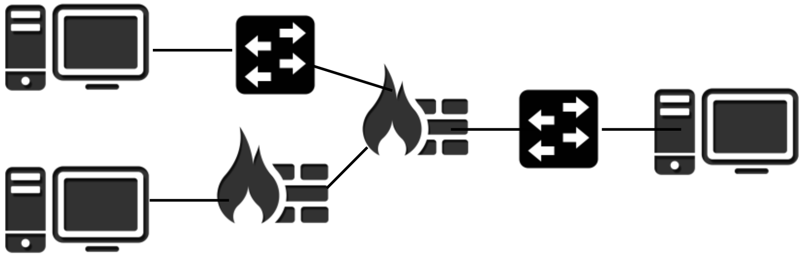
Allocation Graph



VerefOO

Low-Level Security Policies

Service Graph

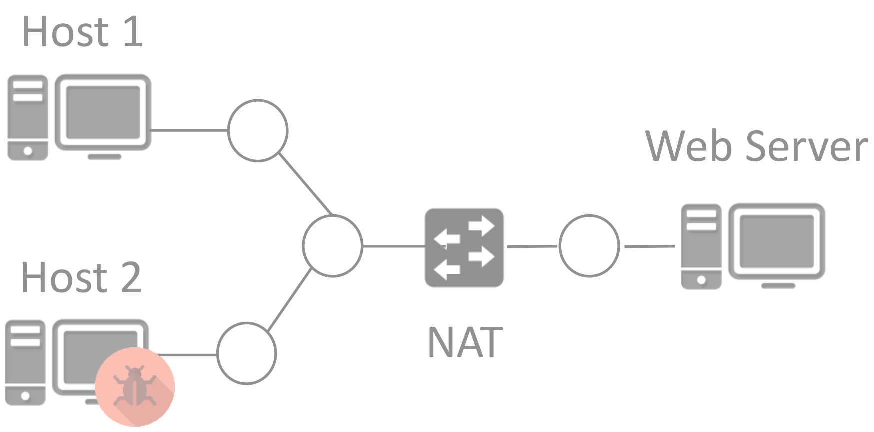


Non-Enforceability Report

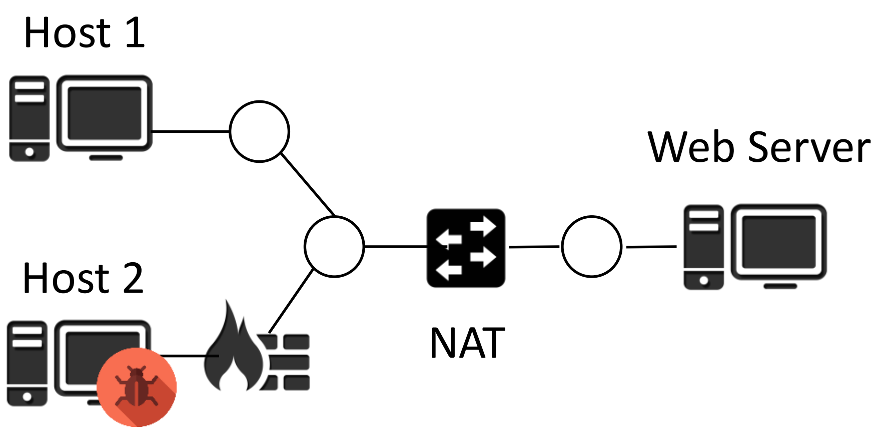
9 | Automated optimal firewall orchestration and configuration in virtualized networks,
D. Bringhenti, G. Marchetto, R. Sisto, F.Valenza, J. Yusupov, Network Operations and Management Symposium (NOMS 2020)

Example. Scenario 1

Objectives: Minimize number of required Firewalls



POLICY	TYPE
HOST1 -> WEBSERVER	REACHABILITY
HOST2 -> WEBSERVER	ISOLATION

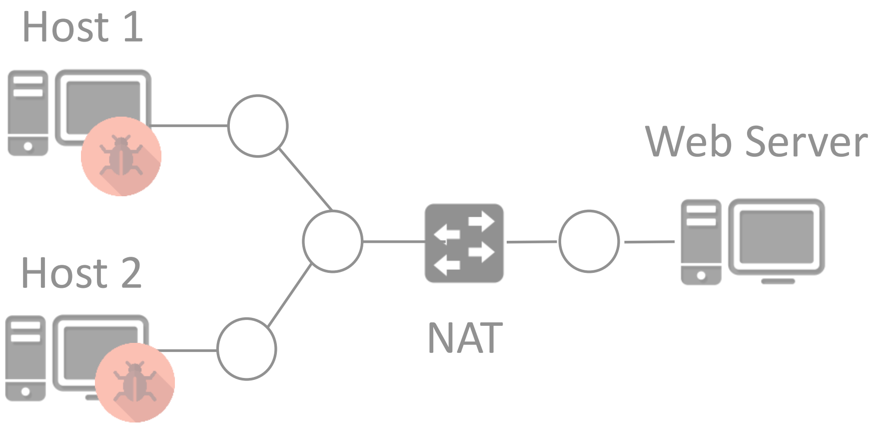


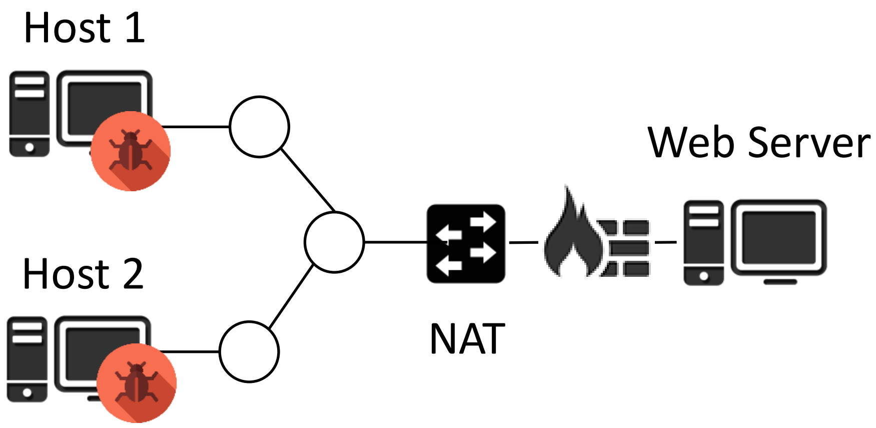
Automatic (deny) rule generation for Firewall 1

SRC IP	DST IP	SRC PORT	DST PORT	PROTOCOL
HOST2	WEBSERVER	*	*	*

Example. Scenario 2

Objectives: Minimize number of required Firewalls
Minimize number of Firewalls rules



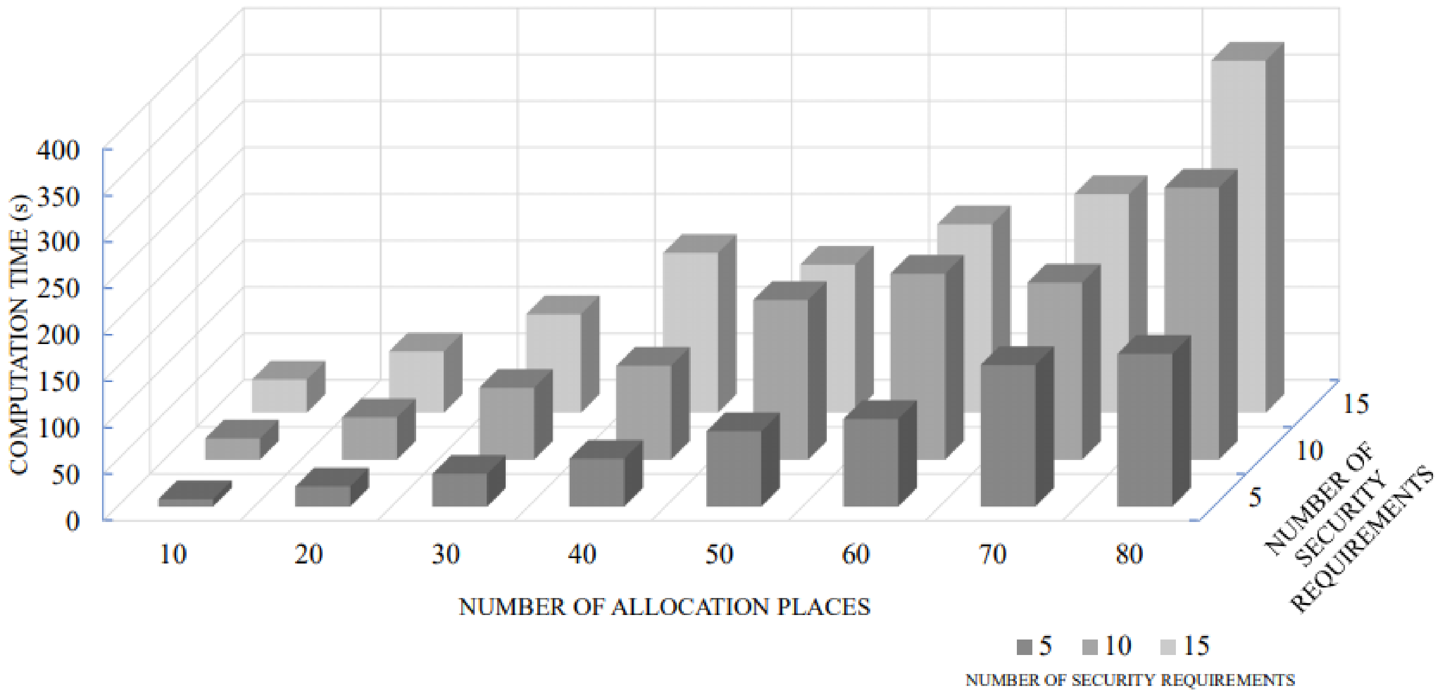


POLICY	TYPE
HOST1 -> WEBSERVER	ISOLATION
HOST2 -> WEBSERVER	ISOLATION

Automatic (deny) rule generation for Firewall 2

SRC IP	DST IP	SRC PORT	DST PORT	PROTOCOL
NAT	WEBSERVER	*	*	*

Results



Towards a fully automated and optimized network security functions orchestration,
D. Bringhenti, G. Marchetto, R Sisto, F.Valenza, J. Yusupov, Optimization in Computing and Networking (OptiComNet 2019)

Automated Security Management for Virtual Services

application of an automatic user-driven security solution operating in a real cloud orchestrator

- User policies are automatically translated and refined
- Formally enforced and deployed in an optimized form
- A well-known and widely used orchestrator has been selected (Kubernetes)

- limited to automatic firewall configuration

Developed within EU H2020 Project ASTRID

10 Automated Security Management for Virtual Services,
M. Repetto, A. Carrega,J. Yusupov, F. Valenza, F. Risso,G. Lamanna, Conference on NFV and SDN - Demo Track.

Summary

- “user-friendly” network function modeling approach
  - using formal analysis tools without requiring strong expertise in formal methods
- joint optimization and verification
  - generates a formally verified optimal placement plan
  - expressive constraints such as forwarding behavior
- automatically define the configuration of network function
  - security policies are correctly and optimally enforced
  - minimizing the number of rules inside each firewall

POLITECNICO DI TORINO

Future research directions

- A heuristic approach for automatic network function selection and configuration

- Formal definition of industrial critical systems for fault diagnosis
- Intrusion Detection, Network Analysis & Verification in the vehicle network traffic
- Formal validation of automobile production systems : PLC software

Publications

- **Proceeding**
  - A Framework for User-Friendly Verification-Oriented VNF Modeling, **2017**
    G. Marchetto, R. Sisto, M. Virgilio and J. Yusupov. *41st Annual Computer Software and Applications Conference (COMPSAC)*
  - Formally verified latency-aware VNF placement in industrial Internet of things, **2018**
    G. Marchetto, R. Sisto, J. Yusupov, A. Ksentini, *14th International Workshop on Factory Communication Systems (WFCS)*
  - Virtual Network Embedding with Formal Reachability Assurance, **2018**
    G. Marchetto, R. Sisto, J. Yusupov, A. Ksentini, *14th International Conference on Network and Service Management (CNSM)*
  - Multi-Objective Function Splitting and Placement of Slices in 5G Mobile Networks, **2018**
    G. Marchetto, R. Sisto, J. Yusupov, A. Ksentini, *IEEE Conference on Standards for Communications and Networking (CSCN)*
  - Towards a fully automated and optimized network security functions orchestration, **2019**
    D. Bringhenti, G. Marchetto, R. Sisto, F.Valenza, J. Yusupov, *Workshop on Optimization in Computing and Networking* (OptiComNet)
  - Automated Security Management for Virtual Services, **2019**
    M. Repetto, A. Carrega,J. Yusupov, F. Valenza, F. Risso,G. Lamanna, *Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) - Demo Track.*
  - Automated optimal firewall orchestration and configuration in virtualized networks, **2020**
    D. Bringhenti, G. Marchetto, R. Sisto, F.Valenza, J. Yusupov, *Network Operations and Management Symposium (NOMS 2020)*
  - Introducing programmability and automation in the synthesis of virtual firewall rules, **2020**
    D. Bringhenti, G. Marchetto, R. Sisto, F.Valenza, J. Yusupov,
    Workshop on Cyber-Security Threats, Trust and Privacy Management in Software-defined and Virtualized Infrastructures (SecSoft 2020)
- **Journal**
  - A VNF Modeling Approach For Verification Purposes. **2019**
    G. Marchetto, R. Sisto, M. Virgilio, J. Yusupov. *International Journal of Electrical and Computer - (IJECE, a SCOPUS indexed Journal)*
  - A Framework for Verification-Oriented User-Friendly Network Function Modeling, **2019**
    G. Marchetto, R. Sisto, F. Valenza, J. Yusupov - *IEEE Access*

A Formal Approach to Verify Connectivity and Optimize VNF Placement in Industrial Networks. **Under (third) review**
Guido Marchetto, Riccardo Sisto, Fulvio Valenza, Jalolliddin Yusupov, Adlen Ksentini. *IEEE Transactions on Industrial Informatics*
Automated optimal firewall orchestration and configuration in virtualized networks, **Under submission**
D. Bringhenti, G. Marchetto, R. Sisto, F.Valenza, J. Yusupov *IEEE/ACM Transactions on Networking*

POLITECNICO DI TORINO

THANK YOU