# [A framework for Virtual Network Functions (VNF) modeling and Service Graph verification in SDN/Cloud context](https://didattica.polito.it/pls/portal30/sviluppo.tesi.dettaglio)

Candidato: Matteo Marchetti

Relatori: Prof. Riccardo Sisto, Prof. Guido Marchetto

# Summary

The growing necessity of flexibility and simplicity of management of the modern Telecommunications networks leads to the creation of new paradigms like Network Function Virtualization (NFV) and Software Defined Network (SDN). These paradigms follow the trend known as “Softwarization”, that includes the replacement of specific hardware for the network functions inside the IT infrastructures with a general purpose one using the virtualization of the function itself.

In order to create and verify the virtual networks before the real deployment, some studies have been already developed. The aims of these previous projects include the definition of Virtual Network Functions (VNF), the creation of models of possible networks and the verification of some properties inside them.

Analysing these previous projects, the definition of the network functions is obtained using First Order Logic formulas. So their behaviour is translated into a set of implications that define the rules of all the actions that the VNF could perform. Also the network models and the properties to be checked are defined using this type of formulas.

For the verification phase all these rules are collected and supplied to Z3, a Satisfiability Modulo Theories (SMT) solver. It can determine if the properties inside the network could be satisfied (result equal to SAT) or not (result equal to UNSAT).

But analysing their structure, these projects result quite difficult to use in a real situation because of the complexity of the VNFs definition and the network testing.

So, starting from these previous results, the objectives of this thesis is the development of a framework that allows Virtual Network Function modelling in a easy way and offer a tool for the verification of the service graph created.

In order to guarantee the easiness of usage, a well-known programming language has been chosen for the VNF’s modelling: Java. Using it, a library has been developed and it represents the set of instructions available for the user for describing his own network function’s behaviour.

It has been developed following three objectives:

1. To allow a simple definition of the network function for the user
2. To leave some general concepts and freedom for the programmer in such a way that he could define the wanted behaviour
3. To can find an automatic translation from the Java definition into FOL formulas

After its definition, I have defined a translating pattern from the Java VNF definition to a set of FOL formulas.

Following these translating rules, I have defined a Parser that takes as input the Java file and creates an internal representation of the data contained. Using it, an other Java file is created and it represents the input to supply to Z3. Its creation and the correct representation of all the data defined from the user inside the source file is one of the main goal of this thesis.

Then all the network functions and all the tests defined in the previous projects have been rewritten. Firstly, using the library, all the definitions of the network functions have been created and through the Parser the input files for Z3 have been obtained. Subsequently all the tests have been rewritten using the new input files.

At the end, in order to prove the efficiency of the framework developed, both the previous and the new tests have been executed and their results have been compared.

For this phase, two Ant files have been created: the first one uses the Parser to rebuild all the VNF input file starting from their Java definitions; the second one runs all the tests using the classes written ad hoc (defined into the previous projects) and the ones automatically created with the Parser and compares the results.

Analysing the output of the second Ant file, it could be possible to see that the results are equal. It means that the whole process could be correct.

During the tests, a difference between the two results has been observed and analysing the structure of the input files and the network models, an error in the behaviour of the previously definitions of VNFs has been detected.

So, not only the tests have been confirmed the efficiency of the framework developed, but using it they have found an error in the previous projects.

In conclusion, considering the results obtained in terms of easiness of project’s usage also for non technical programmers (thanks mainly to the library and to the guide developed) and the reliability in terms of creation of the classes that describe the behaviour of the Virtual Network Functions, I think it is possible to say that the project of this thesis has reached its objectives and I hope that it will be used in other future wider works that really will allow a transformation of the current structure of the network into an other one more flexible, cheaper and simple to manage.