Machine Learning for Autonomic Network Management in a Connected Cars Scenario

Gorka Velez[1*], Marco Quartulli[1], Angel Martin[1], Oihana Otaegui[1], and Haytham Assem[2]

[1] Vicomtech-IK4,
Paseo Mikeletegi 57, 20009 San Sebastian, Spain
{gvelez,mquartulli,amartin,ootaegui}@vicomtech.org
[2] Cognitive Computing Group, Innovation Exchange, IBM Ireland
Shelbourne Road, Ballsbridge, Ireland
haythama@ie.ibm.com

Abstract. Current 4G networks are approaching the limits of what is possible with this generation of radio technology. Future 5G networks will be highly based on software, with the ultimate goal of being self-managed. Machine Learning is a key technology to reach the vision of a 5G self-managing network. This new paradigm will significantly impact on connected vehicles, fostering a new wave of possibilities. This paper presents a preliminary approach towards Autonomic Network Management on a connected cars scenario. The focus is on the machine learning part, which will allow forecasting resource demand requirements, detecting errors, attacks and outlier events, and responding and taking corrective actions.

Keywords: 5G, connected cars, machine learning, network management

1 Introduction

The Internet of Things (IoT) vision promotes the interconnection of objects that have traditionally worked offline, creating new opportunities for more direct integration between the physical world and computer-based systems. However, current 4G technology is approaching its limits, and will not be capable of covering the necessities of the huge amount of devices that are expected to utilize this network [1]. Due to these vast and huge increase in the number of devices, network management challenges are becoming more and more complicated [2]. Hence, the 5G which is the new generation of radio systems and network architecture will need to specially take into account IoT challenges in relation to network management [3]. Apart from being able to make maximum use of available spectrum and data transmission rates, it will need to largely manage itself. Virtualisation will play a key role here. Instead of building a network to

* corresponding author

meet an estimated maximum demand, the network will need to provision itself dynamically to meet changing demands.

Machine Learning is expected to be one of the main key technologies that will enable this vision of self managing network. Machine Learning technologies can learn from historical data, and make predictions or decisions. Instead of following static programming instructions, it can dynamically adapt to new situations learning from new data [4]. This technology has been successfully used in image analysis, language recognition and many other applications. It has also a great potential in the network management area. For example, it can be used to forecast resource demand requirements, detect security threats and error conditions, and react correctly to them.

The transportation sector is one of the key sectors that will be benefited from 5G. Autonomous and semi-autonomous cars, which are expected to decrease significantly the chance of accidents, will depend heavily on connectivity. Cars will become complex electronic devices that have to cope with multiple, high-bandwidth, heterogeneous and asynchronous data sources such as cameras, radars, GPS, etc. So the network must ensure high data rates of big volumes of data at potentially high travelling speeds, what supposes a great challenge.

The aim of this paper is to show the potential of machine learning for its application on the softwarisation and virtualisation of common network functions, in the specific case of a connected cars scenario. This article focuses on the concept and architecture of our approach and also proposes an evaluation criterion.

The rest of the paper is structured as follows. Section 2 describes the state of the art on the main technologies involved in Autonomic Network Management. Section 3 defines a connected cars scenario highlighting the main challenges. Section 4 proposes a generic solution based on machine learning for the problem presented in section 3. Finally, Section 5 concludes the paper.

2 Background

5G breakthroughs will not only lay on the radio access network, by means of additional spectrum bands and higher spectral efficiency, in order to achieve higher capacity for dense deployments but will also bring solutions to empower the core network. These new approaches at the network design will aim to provide connectivity to a increasing number of users and devices [5]. Rising demand for mobile traffic will enforce new ways of enhancing capacity, such as dense deployment, as well as intelligent traffic steering and offload schemes while reducing operational expenditure [6].

The strategy from industrial partners, network operators, equipment vendors and standardization bodies, like European Telecommunications Standards Institute (ETSI), is to decouple hardware from software and move network functions towards software. For example, virtual network functions (VNFs) could virtualize a router a base station, core mobile nodes GGSN, SGSN, RNC, EPC (all-IP mobile core network for the LTE networks), firewalls, intrusion prevention IPS, etc.

Introducing a new software-based solution is much faster than installing an additional specialized device with a particular functionality. It boosts network adaptability and provide elasticity to make network easily scalable. So, with simpler operation, new network features are likely to be deployed or teared-down more quickly.

However, there is a need for mechanisms to manage the network due to the increase of the network complexity. To organize all the VNF instances under common goals and policies, a policy manager and orchestrator is needed for the life cycle. MANO stands for Management and Orchestration setting up, maintaining and tearing-down VNFs. Moreover MANO entity communicates with OSS/BSS(Operation Support System/Business Support System) system of the telco operator. OSS deals with network management, fault management, configuration management and service management. BSS deals with customer management, product management and order management etc.

Open Source projects such as OpenStack[3], OpenMano[4], OpenBaton[5] or OpNFV[6] implement network functions virtualization (NFV) and MANO stacks. While solutions, like OpenFlow[7] open standard, deploy innovative protocols in production networks by means of communications interface defined between the control and forwarding layers of an SDN architecture.

Going beyond, the transformation of operative switching and forwarding into programmable and configurable functions enable autonomous network management. The key challenge is to enable direct access and manipulation of the forwarding plane of network devices (e.g. router, switch) by moving the network control out of the networking switches to logically centralized control software. A logically centralized network intelligence can tune the network control directly without taking care about the underlying infrastructure that is completely abstract for applications and network services. Thus, networks turn into flexible, programmable platforms with intelligence to meet dynamically performance and react to degradation symptoms.

Today, this vision is not yet realized. The is not a reliable solution that addresses the problems for flexible creation by scaling up/down or in/out an elastic network in an automated way [7]. We propose to overcome autonomous network management by means of the application of Machine Learning to data streams originating from the forwarding plane of network devices.

[3] http://www.openstack.org/
[4] https://github.com/nfvlabs/openmano
[5] http://openbaton.github.io/
[6] https://www.sdxcentral.com/listings/opnfv/
[7] https://www.opennetworking.org/sdn-resources/openflow

3 Scenario definition

The connected cars scenario is a complete and challenging scenario to test future 5G capabilities. Vehicles can exchange information with other vehicles (V2V), with the roadside infrastructure (V2I), with a backend server (e.g., from a vehicle manufacturer or other mobility service providers) or with the Internet (V2N), with a pedestrian (V2P), etc. The term Vehicle-to-Everything (V2X) is used to refer to all these types of vehicular communication. Typical automotive use cases for V2X include [8]:

- Advanced Driver Assistance Systems: In this application, connected cars periodically provide either status information (e.g., position, speed, acceleration, etc.) or event information (e.g., traffic jam, icy road, fog, etc.). This information is usually packed into stateless, individual messages or probes which are either locally disseminated to neighboring vehicles, or sent to a central point (base station, backend) where it can be aggregated and then disseminated to other vehicles to make use of it.
- Enhanced navigation: An efficient connectivity is needed to enable a collaborative navigation, where each car can receive information in real-time from other cars or from roadside infrastructure about noticeable events such as road works or traffic congestions.
- Information society on the road: people are demanding high data rate and low latency connectivity when travelling inside vehicles. The introduction of the autonomous vehicle will increase the consumption of data traffic on the move, as drivers no longer need to be focused on driving tasks.

One critical factor in the connected cars scenario is performance. Network performance degradations could potentially impact people lives by affecting the vehicle safety for instance by causing delays in the network. The cause of such issues must be detected in advance, and avoided such that no performance degradations occur in the network.

Moreover, in the current deployments and in order to encompass the connected cars scenario, the system will need to be highly over provisioned (i.e., more resources are available in the network) based on the peak time connectivity requirements. This requires that a large amount of resources can be made available very fast by the system at any time. This continuous time availability is very expensive to maintain. Therefore, an efficient and accurate service demand prediction in the connected cars scenarios is necessary to accordingly provision the network. This should lead to optimizing performance and the use of available network and VM resources, maintaining the Quality of Service (QoS).

4 Proposed solution

4.1 Architecture

The high-level architecture proposed to solve the problems stated in the connected cars scenario is depicted in Figure 1. In a nutshell, we propose defining a Machine Learning cluster, the Data Analyser, and two data flows: an input data flow for measurements and monitoring, and an output data flow for policies. Therefore, we need two APIs to interact with our solution. Using these APIs, the deployment and integration of an existing virtualized infrastructure with the proposed solution is limited to the data ingest and the policy recommendation embodied in two APIs that decouple our solution from the analysed and optimized system. The APIs would be based on HTTP, and would use JSON/Openstack like structured data.

The Data Analyser, which is the core Machine Learning module, is fed by metrics data that aim to describe the state of the network. Some of this data comes from the telecommunications operator, while the rest of the data is conventional data gathered from individual VMs or from the VM management software. The Data Analyser contains different Machine Learning algorithms, that are used to detect different kind of events. The main aim of the Data Analyser is to generate rules and to inform the Policy Manager for resource provisioning.

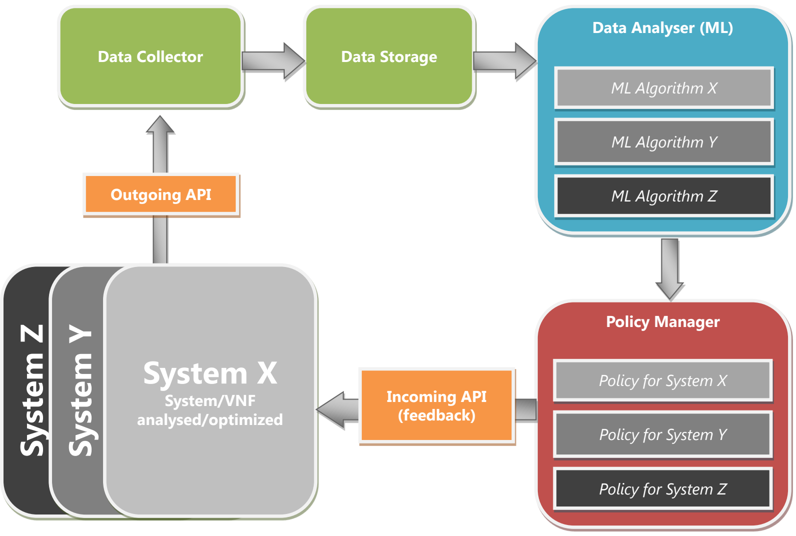

Fig. 1. High-level architecture for machine learning based network management.

As it is evident, the autonomic management of virtualized telecommunication network infrastructures inherently represents a Big Data problem, as it naturally encompasses issues of data Velocity, Variety and Volume: decisions need in principle to be taken in real-time, based on incomplete information represented in large collections of high-dimensional measurements jointly acquired by different layers of the networking stack and therefore residing in largely separated descriptor sub-spaces. This characteristic of the problem needs to be taken into account when devising a working solution to the problem.

If the Machine Learning subsystem needs to execute network load classification and forecasting functions in order to support operational and cost-effective management decisions, then we consider that this effectivity requires both that:

- The metrics acquisition, storage and the initial data reduction and analysis components are physically co-located as close as possible in order to minimize the cost of transferring large volumes of data from one component to another.
- The joint metrics acquisition, storage and pre-analysis elements form a pyramidal decomposition of Composites in the sense of [10] (cfr. figure 2) that progressively reduce and abstract metrics to higher and higher levels, proposing (e.g. horizontal and vertical scaling) remediation measures at a local level while delegating higher-level contextual ones to parent levels in a Map/Reduce pyramid [11].

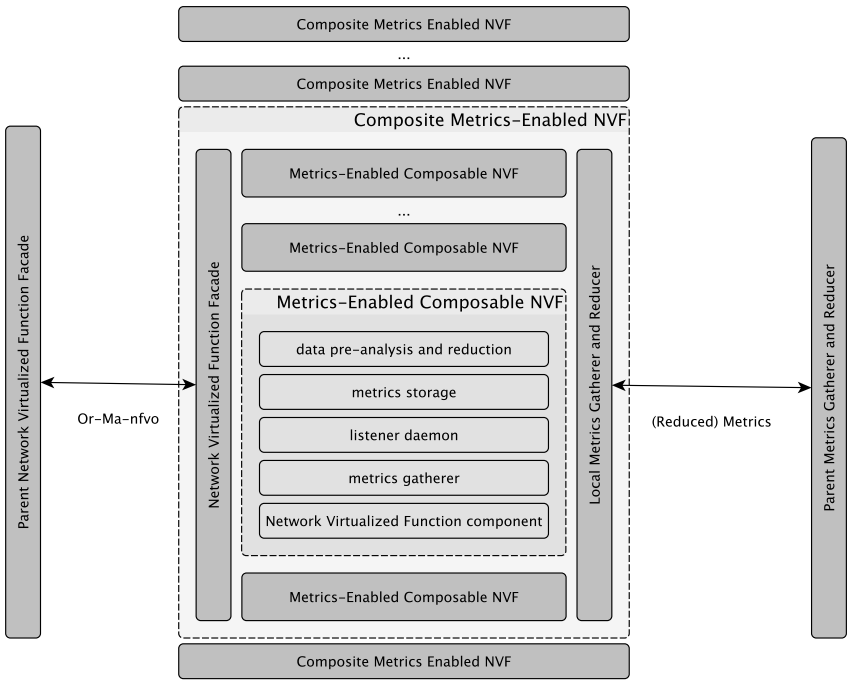

Fig. 2. Joint metrics acquisition, storage and pre-analysis elements form a pyramidal decomposition of Composites in the sense of [10] that progressively reduce and abstract metrics.

Distributed real-time analytics architectures beyond the Lambda need to be considered for the effectual processing of this kind of streaming data.

While large scale near real-time data processing is a crucial component, standard architectures typically only permit batch processing strategies on large scale metrics collections.

As per [12], we propose instead to consider the streaming approach as the standard one for cognitive NVF self-management, adapting large and historical batch data processing to a near-real time scenario by task queueing and smart scheduling. To this end, we propose to consider how the organization of the data on N-dimensional lattices and the locality of access patterns in the spatio-temporal and in the resolution dimensions can allow us to define and exploit a methodology that is based on streaming cluster computing frameworks [13], Hilbert curve scheduling [14] and multi-scale pyramid decompositions for optimizing access to distributed storage and computing resources and maximize perceived processing speed in real-time operations.

In doing so, we propose adapting an architectural approach from the field of Earth Observation data mining to improve on the 'Lambda architecture' [15], the prevalent approach in managing the contradiction between the large sizes of metrics data and the significant data rates their processing involves.

4.2 Required components and proposed methods

Network Sensorization The collection, management and analysis of metrics describing the state of a virtualized telecommunications network requires that components are set up and assembled in a proper architecture (see again Figure 2) to that end.

Requirements for this set of components include:

- The availability of a well-defined querying API allowing standardized access to the collected metrics.
- Fully automatic setup and deployment for the components and the resulting composed architecture.
- Open metric protocol defining the ways in which metrics data is formatted and transmitted through the network.
- Scalability of the resulting component set, especially with respect to scenarios in which the generation, transmission, storage and processing of a large volume of small I/O operations represents the normal mode of operation of the sub-system.

Required components include:

- A metrics gathering component capable of receiving the relevant measures generated by the monitored system, such as StatsD [8]
- Listener daemons capable of aggregating metrics from multiple gathering components. It is in this case fundamental to anticipate and address performance issues at scale, for instance by exploiting highly concurrent implementations

[8] https://github.com/etsy/statsd

- A metrics database capable of locally storing the aggregated measures so that at least short term historical analysis is possible. Again, availability issues at scale need to be addressed.
- A trend analysis component that implements and makes available the Machine Learning core functionality.

Distributed optimization and Machine Learning It is important to note that all problems that are solved through data analysis, particularly through the use of Big Data statistical and machine learning algorithms, share a few key characteristics. First, the datasets are often extremely large, consisting of hundreds of millions or billions of training examples; second, the data is often very high-dimensional, because it is possible to measure and store very detailed information about each example; and third the data can be stored or even collected in a distributed manner. As a result, it is essential to develop algorithms that are both rich enough to capture the complexity of modern data, and scalable enough to process huge datasets in a parallelized or fully decentralized fashion.

For our particular problem, mathematical optimization can be used as an aid to a human decision maker, system designer, or system operator, who supervises the dimensioning of an NFV architecture, checks the results, and modifies the problem (or the solution approach) when necessary. In a number of scenarios excluding on-line applications, this human decision maker also carries out any actions suggested by the optimization problem. As an alternative, for on-line scenarios embedded optimization procedures can be used to automatically make real-time choices, and even carry out the associated actions, with no (or little) human intervention.

4.3 Scenario evaluation

In scenarios dedicated to the management of important telecommunication infrastructures such as the one considered here, is crucial to define and execute appropriate validation activities proving that an operational solution is feasible, applicable to different operational contexts, and that it will bring the expected performance benefits.

Here, performance capturing through probes that enable quantitative assessment becomes a key side of the solution. The KPIs and Quality of Service (QoS) are widely used. KPIs depend in a highly intricate manner on the structure and dynamics present in the network. While QoS score is determined by the transport network design and provisioning of network access, terminations and connections.

The connected cars scenario can be developed in two different configurations, depending on whether the application is based on low-level data (e.g., raw data from camera sensors) or based on high-level data (objects or events detected) [8]:

- High-level data transmission requires a medium data rate (up to 1 Mbit/s) with a very low tolerance on errors (10^{-5}).

– Low-level data (mainly video streaming) requires a high data rate (up to 10-20 Mbit/s) with a medium tolerance on errors (10^{-2}).

The connected cars scenario also requires:

– End-to-end latency of less than 5 ms for message sizes of about 1600 bytes.
– Data is sent either event-driven or periodically with a rate of about 10 Hz.
– Minimum throughput: 3Mbit/s. The system under consideration requires high reliability rather than high throughput [9].

In order to measure these parameters and apply policies there is a set of frameworks and plugins that can be used on top of an OpenStack infrastructure. Neutron[9] is the interface of OpenStack to configure the network that helped by ML2 driver provides network attributes. While other solutions like Ryu[10] or OpenDaylight[11] allows setting QoS policies over Open vSwitches using an Open vSwitch database (OVSDB).

5 Conclusions and future work

Current network management devices are generally proprietary and closed, and have very low improvement potential. The state-of-the-art methods force network operators to implement complex policies and tasks with a limited set of low-level device configuration commands. The softwarisation and virtualisation of common network functions can raise the level of abstraction, simplifying and making more flexible the network configuration process.

This new network management paradigm can also impact on the automotive sector, specially with the future adoption of 5G technology, which is expected to be aligned wit this paradigm. V2X communications will need to deal with a big amount of data in an efficient and reliable way. A software-based self-managing 5G network could more easily fulfill the requirements needed for this purpose. Furthermore, there can be a great number of new applications due to the introduction of technologies that allow improved performances.

In this article, we focus on machine learning as a key enabling technology for autonomic network management. We first identify the features and challenges of a connected cars scenario, and then we propose a preliminary approach towards autonomic network management. However, it does not exist a generic machine learning mechanism suitable for all use cases. Developers need to design a specific learning path for each use case, which may combine multiple approaches or algorithms together. So instead of proposing a specific approach, we propose a generic architecture that can accommodate different machine learning strategies.

The deployment and validation of our solution require further and intensive research and development work, which is planned to be done inside the framework of an ongoing H2020 European Project.

[9] https://blueprints.launchpad.net/neutron/+spec/quantum-qos-api, https://blueprints.launchpad.net/neutron/+spec/ml2-qos
[10] http://osrg.github.io/ryu/
[11] https://wiki.opendaylight.org/

Acknowledgments. This work was fully supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action).

References

1. Wang, Cheng-Xiang, et al.: Cellular architecture and key technologies for 5G wireless communication networks. IEEE Communications Magazine. 52(2), 122–130 (2014)
2. Kim, H., Feamster, N.: Improving network management with software defined networking. IEEE Communications Magazine. 51(2), 114–119 (2013)
3. Shariatmadari, Hamidreza, et al.: Machine-type communications: current status and future perspectives toward 5G systems. IEEE Communications Magazine, 53(9), 10–17 (2015)
4. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of machine learning. MIT press, USA (2012)
5. Sun, S., Kadoch, M., Gong, L., Rong, B. Integrating network function virtualization with SDR and SDN for 4G/5G networks. IEEE Network. 29(3), 54–59 (2015)
6. Hernandez-Valencia, E., Izzo, S., Polonsky, B. How will NFV/SDN transform service provider opex?. IEEE Network. 29(3), 60–67 (2015)
7. Szabo, R., Kind, M., Westphal, F.-J., Woesner, H., Jocha, D., Csaszar, A. Elastic network functions: opportunities and challenges. IEEE Network. 29(3), 15–21, (2015)
8. 5G-PPP: 5G automotive vision. White paper. `https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf` (2015)
9. Yao, Y., Rao, L., Liu, X.: Performance and reliability analysis of IEEE 802.11p safety communication in a highway environment. IEEE Transactions on Vehicular Technology. 62(9), 4198-4212 (2013)
10. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns: elements of reusable object-oriented software. Pearson Education (1994)
11. Dean, J., Ghemawat, S. MapReduce: simplified data processing on large clusters. Communications of the ACM. 51(1), 107–113 (2008)
12. Quartulli, M., Lozano, J., Olaizola, I.G.: Beyond the lambda architecture: Effective scheduling for large scale EO information mining and interactive thematic mapping. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp.1492–1495. IEEE Press (2015)
13. Zaharia, M., Das, T., Li, H., Shenker, S., Stoica, I.: Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. In: Proceedings of the 4th USENIX conference on Hot Topics in Cloud Computing 2012, pp. 10–10. USENIX Association (2012)
14. Drozdowski, M.: Scheduling for parallel processing. Springer, London (2009)
15. Marz, N., Warren, J.: Big Data: Principles and best practices of scalable realtime data systems. Manning Publications (2015)
16. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
17. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning. 3(1), 1–22 (2011)