Advances in Aeronautical Science and Engineering
ISSN: 1674-8190

ADAPTIVE MONITORING IN 5G: THE ROLE OF INTRA FRAMEWORKS

Olivia Harris[1], Daniel Harris[1], Grace Moore[2]
Department of Digital Systems, University of Piraeus, Greece

ABSTRACT

Following the rapid development and innovations of mobile networking technologies, an entirely new era of mobile communications systems is evolving, namely the fifth-generation mobile technology (5G). Among the goals of 5G networks, one is to meet predetermined Quality of Service (QoS) requirements, in different application scenarios (e.g., such as data transmission rate and latency). To this end, the demand for always-accessible applications and services is continuously growing, rendering the monitoring of network availability a fundamental requirement for modern infrastructures. Network monitoring systems provide a first level of reliability in case of application failure or performance deterioration. Meanwhile, a monitoring framework should accumulate the process and deliver the monitored data at every requested aggregation level and frequency, without introducing a substantial overhead into the system. In this paper, we focus on the trade-off between the running services and the monitoring framework itself. We propose an adaptive scheduling algorithm, aiming at the removal of static time interval applied in the monitoring system. The main goal is to provide highly accurate information in real time without producing unnecessary traffic to the network.

KEYWORDS

NFV/SDN, 5G Networks, Network Services, Cost-efficient Monitoring

1. INTRODUCTION

Software Defined Networking (SDN) is an emerging architecture, providing dynamic, manageable, cost-effective, and adaptive characteristics. The benefits of this technology make it ideal for the high-bandwidth and dynamic nature of today's demanding applications and services. Researchers are currently exploring different architectures to exploit the main concepts of this new technology (Lin, 2015). SDN has been proposed as a promising technique for these networks, having the role of a key component in the design of 5G infrastructures. In recent years, SDN has emerged as a promising technology facilitating network programmability and an ease in the management of tasks. SDN proposal includes the decouplement of the data plane from the control plane functionalities. While the data plane functionality of packet forwarding is built into switching fabric, the control plane functionality for controlling network devices is placed into a logically centralized software component, namely the controller (Salisbury, 2012). What needs to be mentioned is that the control plane functionality provides a programmatic interface for developing management programs, as opposed to providing a configuration interface for tuning network properties. From a management point of view, this added programmability opens the opportunity to reduce the complexity of distributed configuration and to ease the network management tasks. The monitoring framework can be defined as a continuous service, aiming primarily to provide the efficient management and the sequential sampling of the progress and the state of the network, in the achievement of results (Van Adrichem, 2014). Thus, monitoring and its evaluation system are considered as important management tools to track the overall progress, introducing reliability and the ability of further exploitation through Complex Event Processing (CEP) in 5G architectures.

In more detail, a network monitoring system is capable of detecting and reporting general failures of devices or connections. Under normal conditions, it is responsible for capturing the raw measurements of resource usage or behavior that can be observed and collected throughout the system, such as the measure of the CPU utilization of hosts, the network bandwidth utilization of links, and other aspects of the operations (Lee, 2002). These functions entail the circulation of several messages to every network node in order to verify its responsiveness to requests. In case of failures, unacceptably slow response or other unexpected behavior, these systems transmit additional messages, namely alerts, to designated locations such as management server, email addresses, etc. Such alerts can contribute to the identification of patterns and general insight in system performance, as well as in the demonstration of the requirements into upgrades, replacements or even failures. The functionality of a monitoring network system is based on the establishment of performance baselines for the multiple metrics that are observed. This configuration comprises a useful tool to reveal an insight of the behavior and health of the system (Mitchell, 2018). In that case, the pinpointing of probable failing issues turns into a convenient process, preventing substantial cost and time losses. Bearing in mind that the evolution of the 5G infrastructure is significantly based on the evaluation of the Quality of Experience (QoE), the transmitted information is of paramount importance with an average of 10 Gbps terabytes of data. With this large amount of data being circulated through the high-bandwidth networks, the capture of the traffic speed also becomes an issue.

In this paper, we propose an adaptive scheduling algorithm for collecting and processing accurate data of the network nodes with minimal added overhead, as initially described from (Touloupos, 2018). In general, the proposed addon, provides organization and further processing of the various inputs with the aim of correlating and providing enriched insight of the input information. This approach is under development for the EU-funded project 5GTANGO and its purpose is to support the management of 5G services under the SDN/Network Function Virtualization (NFV) paradigm (5GTANGO Consortium, 2018).

The rest of this paper is organized as follows. Section 2 presents the state of the art with regard to the corresponding monitoring frameworks that can be adapted in 5G/SDNs. In section 3, our proposed approach is described thoroughly, while in section 4 the conclusions and future work are introduced.

2. RELATED WORK

Existing approaches include monitoring platforms or mechanisms for qualifying the requirements of monitoring-based applications in SDNs (e.g., anomaly detection, Quality of Service - QoS, network link utilization and throughput measurement). Of the best-known monitoring systems, Payless (Chowdhury, 2014) is a monitoring framework built on top of an OpenFlow (OpenFlow Community, 2016) controller's northbound Application Programming Interface (API). The Payless monitoring framework achieved high levels of accuracy and precise statistics of the monitored signal, while incurring little network overhead. Thus, the Payless monitoring framework provided interoperability between the several configurations while and the hostage of applications with the same set of services available. OpenFlow has emerged as the de facto standard for communication between the controller and switches in SDNs. Apart from providing flow control and communication interfaces, OpenFlow provides a flow level statistics collection mechanism from the data plane. Nagios comprises a different monitoring framework that can be adopted in an SDN paradigm (Nagios Enterprises, 2018). The great advantage of Nagios monitoring system is in the sampling of the entire infrastructure. What Nagios takes into great detail are the network nodes to ensure that systems, applications, services, and business processes are functioning properly. In case of a failure event, Nagios can transmit alert messages with a plethora of relative information, allowing the beginning of remediation processes before outages affect business processes, end-users, or customers. The design is developed as an open source software that can be further configured to meet the user's needs. Based on the functionality of OpenFlow, OpenSketch addressed a new software-defined traffic measurement architecture with generic and efficient measurement regulations (Yu, 2013). What OpenSketch provides is a simple three-stage pipeline, namely the hashing, the filtering, and the counting stage. The implementation of these stages is confronted with commodity switch components and support a sheer amount of measurement tasks. In the level of the control plane functionality, OpenSketch provides a measurement library that automatically configures the pipeline and allocates resources for different measurement tasks. Furthermore, a worth mention monitoring framework is Prometheus (Prometheus Community, 2018). Prometheus is an open-source system monitoring and alerting toolkit originally built at SoundCloud. Among the main features of Prometheus are that it provides a multi-dimensional data model (time series identified by metric name and key/value pairs), including a flexible query language to leverage this dimensionality. Furthermore, there exist no reliance on distributed storage, since single server nodes are autonomous, while time series collection happens via a pull model over HTTP. It should be added that with regards to the pushing time series, they are supported through an intermediary gateway, while targets are discovered through service discovery or static configuration, including multiple modes of graphing and dashboarding support.

3. BASELINE APPROACH

3.1 Static Threshold: Definition and Downsides

The main functionality of a monitoring system, especially in 5G networks, is the attachment of probes across the network and their management, in order to build network-monitoring solutions with greater flexibility and scalability. The probes are responsible for collecting monitoring data from the running services and reporting them to the monitoring server. For computational savings, the reports are returned only if the monitored metrics exceed a set of predefined thresholds. The monitoring server performs an analysis of the data and takes actions, based on the QoS criteria that are set by the user. The proposed addon, which can be adapted by any monitoring framework, as presented in Figure 1, aiming at enabling decision taking at the probe level. The main goal of this functionality is the facilitation of decision making substantially near to the actual time of the event. The back-propagation of the results is performed for the case of detecting an event in real-time, providing less computational necessities from the infrastructure.

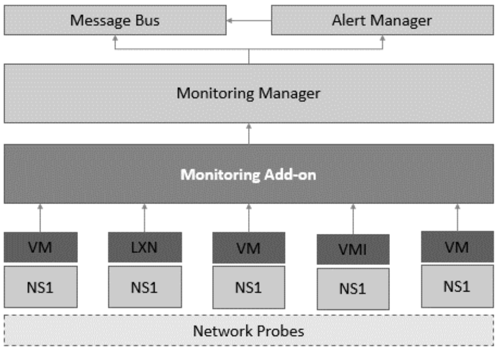

Fig. 1. Monitoring Framework high-level architecture

As probes can be attached in every node, the collection of data from the running Virtualized Network Functions (VNFs) or Network Service (NSs) is enabled. The time interval for the sampling, called the “push” process, is usually a static variable/rule set by the service developer or the monitoring framework itself. The static approach decreases the success rate of preventing a potential service degradation. If a threshold is set high, the detection of a critical event is probable to get unidentified and, at the same time, the success rate of the alerts that are triggered, is decreased. On the other hand, if a threshold is set too low, the volume of the alerts can be overwhelming. In addition, unnecessary traffic is produced caused by the “sensitivity” of the threshold’s value. In the proposed approach, we aim at removing the “static” threshold approach for the time intervals in monitoring frameworks. This comes along with the potential 5G concept and values, as monitoring frameworks will come a step closer to real-time adoptions of the infrastructure as well as massive scale whereby there are so many variations of NS and network requirements. The initial definition of a threshold is also a critical step that affects the overall performance of a monitoring framework. To this end, in this paper we try to dynamically determinate the appropriate threshold of monitored metrics.

3.2 Determining a metric’s threshold

The definition of a threshold in measuring an alert for a specific metric is of paramount importance. Rules and thresholds define how the system represents events in the insight of the data. A simple approach to define a metric’s threshold is by setting a predetermined value as a proportion to the corresponding value of the metric signed in the Service Level Agreement (SLA) (Kapassa, 2018). If the corresponding metric falls below the predetermined threshold, an alert occurs. At the next stage, monitoring data of the running service should be collected, reviewed and assess the normal values for the monitored service. Currently, in 5GTANGO project, monitoring data is collected either from individual VNFs or from NSs, as part of the deployment on a testing Service Platform (SP) infrastructure for the purposes of a testing framework, called Verification and Validation

Platform (V&V). The monitoring results produced by the stress test in V&V, are finally the ones needed to define a metric's threshold before the service is instantiated in the productive SP. Thus, the proposed architecture gains an a-priori knowledge of the thresholds before the actual production environment.

3.3 An adaptive scheduling algorithm

As it was previously introduced, a monitoring framework in a 5G infrastructure needs to continuously measure and assess the metrics or service behaviors in terms of performance, reliability, power usage, ability to meet SLAs, security, etc. The aim is to perform business analytics, for improving the operation of systems and applications, as well as for several other activities (Help Systems Blog, 2017). The vast majority of the scheduling algorithms are defined for coupling to a predefined monitoring framework. The vision of the proposed approach goes beyond the monitoring framework, using a general approach that will be easily compatible with most of the monitoring frameworks in 5G networks, cloud services and IoT applications. Compared to the other approaches, the proposed algorithm introduces the key aspect of linear increase of the data sampling with the attached probes collecting data with linear increase of the time. This rate of increase will provide adequate monitoring decisions with reduced actual overhead. As an example, Figure 2 presents a general case where monitoring data is collected every 1 second. Equally it is depicted that the "pushes" have an endurance as the whole life (in seconds) of the network service (NS).

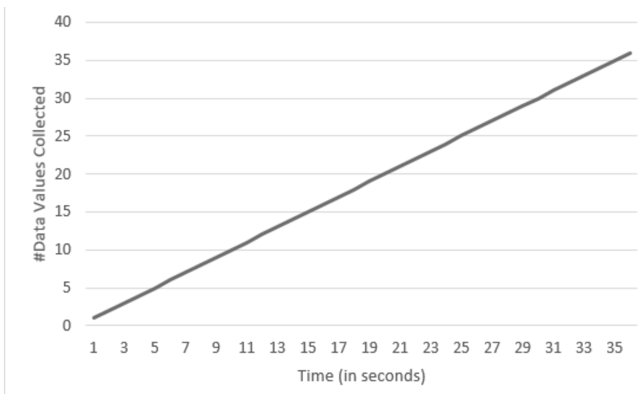

Fig. 2. Amount of data in traditional monitoring

The proposed implementation allows the aforementioned decision to be taken, considering the time intervals for transmitting the monitoring data. The mechanism will obtain from the server the QoS parameters and evaluate the convergence of the NS monitoring data towards the QoS thresholds (i.e. linear change of specific values). In Figure 3, the overall process is described in pseudocode format. The probes collect the monitoring data, adding them to a local list, which along with the data collected, is pushed to the server. It is worth mentioning that the probes are not sending each value itself, but the whole list. Figure 4 depicts the time increase and the average amount of the monitored metrics through simulation results in several NS. Compared to the conventional algorithms in Figure 2, the proposed algorithm obtained 36 values after a time of 36 seconds, equal to 36 "pushed". Our approach shows that the same number of monitored values could be gathered with

```
Max_linear_timeout = X
Time_interval = 1
Final_time = time.now + time_interval

WHILE Time_interval < Max_linear_timeout
        WHILE TIME.NOW < Final_Time
                Value = time.now.get(data)
                VarrayList.add(value)
                Sleep(1)
        ENDWHILE
Mon_Server.PUSH(VarrayList())
VarrayList.clean()
Time_interval += 1
ENDWHILE
```

Fig. 3. Adaptive algorithm - Pseudocode

less “pushes” to the monitoring server. The result of the above approach is having less traffic to the network, while maintaining the same accuracy on the data. On the other hand, the increase of the time interval cannot be into an infinity loop. Therefore, the definition of “timeout” needs to be clarified, as for the time interval where the proposed algorithm should “restart” the overall process and set the time interval to zero. Thus, the collected monitoring data from the testing platform can introduce an insight of the thresholds that should be set, along with the relevant “timeout” value. Supposing that it is set to 60 seconds, the time interval of the “push” process should increase its value until it reaches the timeout’s value. Finally, data is being captured and pushed towards the monitoring server.

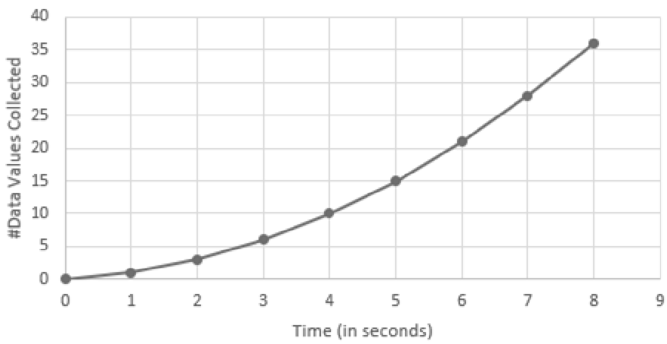

Fig. 4. Amount of data with the proposed algorithm

3.3 Improvement of the Algorithm

In this section, current approaches on improving the proposed algorithm are presented. There are currently two ways of avoiding loss of values for the network’s health while being informed with accuracy for the whole NS life. Inspired from the Transmission Control Protocol (TCP) and Congestion Control algorithms, the two approaches encounter the challenge using a similar approach of the “slow start retransmission” and “fast recovery”. The collection of the monitoring data and the corresponding correlation with the metric’s threshold value set at a previous step, a detection of the level is possible whether there was a critical change in the NS’s lifecycle or an instant change of the metric’s value, or a “false” alert. The approach of “fast recovery” and “slow start” focuses their functionality on monitoring the difference between the threshold of the metric and the corresponding monitored value. In case of this difference is lower than a predetermined value, the “fast recovery” mode is triggered, where the probe pushes the monitored data directly to the monitoring server and set the next time interval to half of its value. The corresponding mode of “fast-recovery” is depicted in Figure 5. Elsewise, in case of this difference exceeding a predetermined value, “slow start” mode is triggered, where the probe pushes the list with the values collected until that time and set the next time interval to 1 second. The corresponding mode of “slow start” is depicted in Figure 6.

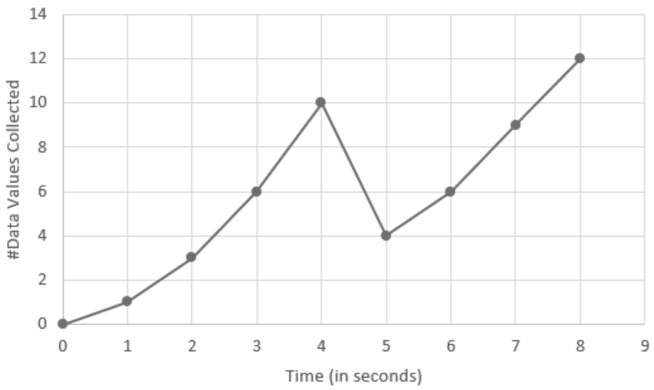

Fig. 5. Time Interval adaptation : Fast recovery

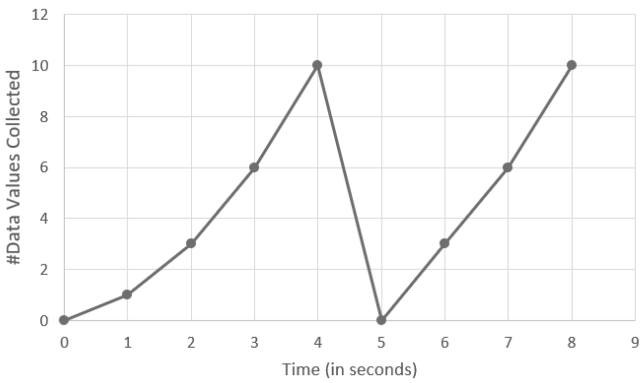

Fig. 6. Time Interval adaptation: Slow start

4. CONCLUSION AND FUTURE WORK

In this paper, we have introduced an adaptable algorithm for monitoring frameworks which can be used in 5G infrastructures, cloud services and IoT applications. The main idea was to efficiently collect monitoring data, using minimal resources of the network. In addition, we have identified that the rationale behind this time interval adjustment, is that we maintain a higher pushing frequency for data that significantly contributes to link utilization, while maintaining a lower pushing frequency for data that does not significantly contributes towards link utilization at that moment.

As a future work, and in the frame of 5GTANGO EU funded project, the introduced algorithm will be adopted by the monitoring framework in the SP. Furthermore, it should be further enhanced in order to facilitate decision making with respect to which data to transmit and how to optimize the overall process. Moreover, at this stage, the introduced algorithm is going to be used for scheduling the collection of monitoring data for one metric per NS. In the near future, our goal is to furtherly enhance our approach for scheduling and collecting monitoring data for all the metrics that are listed in the NS descriptor.

ACKNOWLEDGEMENT

This work has been partially supported by the 5GTANGO project, funded by the European Commission under Grant number H2020ICT-2016-2 761493 through the Horizon 2020 and 5G-PPP programs (http://5gtango.eu). Moreover, this work has also been supported by the MATILDA project, also funded by the European

Commission, under the Grant number H2020ICT-2016-2 761898 through the Horizon 2020 and 5G-PPP programs (http://www.matilda-5g.eu/).

REFERENCES

5G Consortium, 2018, 5G Development and Validation Platform for glo , 2018bal Industry-specific Network Services and Apps, Available at: http://5gtango.eu/

Allman, M., 2009, Congestion Control, Available at: https://www.rfc-editor.org/rfc/pdfrfc/rfc5681.txt.pdf

Chowdhury, S.R., Bari, M.F., Ahmed, R. and Boutaba, R., 2014, PayLess: A low cost network monitoring framework for Software Defined Networks, *"Proceedings of Network Operations and Management Symposium",* Krakow, Polans, pp. 1-9

Help Systems Blog, 2017, Top Benefits of Network Monitoring, Available at: https://www.helpsystems.com/resources/articles/top-benefits-network-monitoring

Kapassa, E, Touloupou, M., Mavrogiorgou, A. and Kyriazis, D., 2018, SLAs in 5G: A Complete Framework Facilitating VNF- and NS- Tailored SLAs Management, *"Proceedings of 32nd International Conference on Advanced Information Networking and Applications Workshops"*, Krakow, Poland, pp. 469-474

Lee, J., Lee, S., Lee, J., Yi, Y., Park, K. 2015, FloSIS: A Highly Scalable Network Flow Capture System for Fast Retrieval and Storage Efficiency, *"Proceedings of USENIX Annual Technical Conference"*, California, USA, pp. 445-457

Lee, H. J., Kim, M. S., Hong, J. W., Lee, G. H., 2002, QoS parameters to network performance metrics mapping for SLA monitoring. *KNOM Review*, Vol. 5, No. 2, pp.42-53.

Lin,Y.D., et al, 2015, An extended SDN architecture for network function virtualization with a case study on intrusion prevention, *Network*, Vol. 29, No. 3, pp. 48-53

Mitchell, B., 2018, Common Home Network Problems, Available at: https://www.lifewire.com/tophome-networking-problems-and-mistakes-817736

Nagios Enterprises, 2018, Nagios Monitoring Framework, Available at: https://www.nagios.org/

OpenFlow Community, 2016, Special Report: OpenFlow and SDN – State of the Union, Available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/special-reports/Special-Report-OpenFlow-and-SDN-State-of-the-Union-B.pdf

Prometheus Community, 2018, From metrics to insight, Available at: https://prometheus.io/

Salisbury, B. 2012, The Control Plane, Data Plane and Forwarding Plane in Networks, Available at: http://networkstatic.net/the-control-plane-data-plane-and-forwardingplane-in-networks/

Touloupou, M., Kapassa, E., Kiourtis, A., Kyriazis, D., 2018, Cheapo: An algorithm for runtime adaption of time intervals applied in 5G networks, *"Proceedings of Fifth International Conference on Software Defined Systems"*, Barcelona, Spain, pp.

Van Adrichem, N.L.M., Doerr, C., Kuipers, F.A., 2014, OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks, *"Proceedings of Network Operations and Management Symposium"*, Krakow, Poland, pp. 1-8

Yu, M., Jose, L.,Miao, R., 2013, Software Defined Traffic Measurement with OpenSketch. *NSDI*, Vol. 13, pp. 29-42