Testing Technologies to Support Network and Services Testing in a 5G Test Network

Teemu Kanstrén, Jukka Mäkelä, Esa Piri, Jussi Liikka, Atso Hekkala
VTT, Oulu, Finland
email:firstname.lastname@vtt.fi

Abstract—**Trends such as 5G and Internet of Things are driving modern systems towards increasing complexity in diverse configurations of heterogeneous networks, ubiquitous integration of hardware and software, and complex interactions between the different parts. This paper describes the testing technologies developed and deployed in our 5G test network (5GTN), to support development and testing of such systems, next generation services deployed on them, and the underlying network technologies. We describe the 5GTN testing technologies, including the software architecture enabling distributed test generation, monitoring and data collection from test execution. We also describe the 5GTN integrated data analytics services enabling efficient use of the test data, as well as initial results for the first tests in the network.**

Keywords - ***5G, test network, testing technologies, big data, analytics***

I. INTRODUCTION

Internet of Things (IoT), cloud computing, big data processing and fifth generation (5G) networks are all trends currently strongly driving next generation software and service development. They are enabling services such as accurate (indoor) positioning, low latency control, high bandwidth streaming, deep data insights, and large scale computational capacity on demand. However, these are currently fast evolving technologies, and for many actors in the service development space it is difficult to benefit from these opportunities to create such next generation services, due to limited access to suitable environments enabling innovation in this space.

The Finnish national 5G test network (5GTN, [11]) is a joint effort created by VTT, University of Oulu and 15 industry partners. It is designed to support a number of use cases for testing, network management and business development purposes. Some of the main examples include:

- Support testing new applications and services, as well as networking solutions in evolving networks.
- Provide a living lab environment for 3rd party application, service, algorithm, system testing.
- Offer a test network for virtualized services.

Some of the generally identified prime objectives in 5G technologies are increased capacity, increased data rate, lower latencies and higher quality of service [3]. Specific technologies typically associated with 5G are, for example, small cell access points, virtualized network elements/network cloud, and increased IoT traffic and adoption [3]. These provide both opportunities for new types of service development (e.g., higher bandwidth and lower latencies) but also challenges (e.g., different types of traffic profiles in IoT stressing the network in unanticipated ways). Making use of the opportunities and addressing the challenges makes 5G relevant to almost all actors in the software and networking domain.

The 5GTN is an environment intended to enable service innovation in this context by providing a test environment that is constantly incorporating latest technologies available in the 5G networking infrastructure, as well as providing support for different levels of cloud computing (including mobile edge computing [17]), IoT devices and services, and extensive monitoring and big data analytics support. The test network is provided as a service to interested parties working in the area, to provide an environment for developing and testing new innovative next generation services. This both removes the barrier for companies who do not have direct access to such environment themselves, as well as provides a place for network and telecom equipment vendors to test their products with actual end users, customers and next generation services.

The initial version and use cases for 5GTN have been described in [5], and a general overview of its testing technologies was given in [4]. A more recent technical overview of the network elements is given in [9]. In this paper, we focus on describing latest developments in the testing technologies part, as well as in describing the initial use cases/test scenarios.

The rest of the paper is structured as follows. In Section II, we describe related work. In Section III, we present the 5GTN architecture. In Section IV, we briefly illustrate some example scenarios for the 5GTN. In Section V, we discuss these in a broader context. Finally, conclusions sum it all up.

II. RELATED WORK

5G is currently a hot topic and various test networks exist to support different actors in developing 5G products and services. The big players in the field have been running their own specific 5G technology tests already for a long time [2]. However, access to such technology is limited for smaller players. 5G test networks are means for these two types of actors to interact, with the smaller (more software service focused) players having access to a more realistic and state-of-the-art test environment, and the telco actors getting access to real end users to test their products.

We briefly review some of these other 5G networks here to give added context to our 5GTN. Each of these has a specific focus, while we provide a holistic overall test network ranging from 5G devices and virtualized network functionality to software services. 5GTN is also itself part of a broader network of Finnish testbeds related to 5G development, called 5GTNF [12].

The *5G Berlin* [15] is a German test network providing a number of different testbeds for 5G development and testing, such as 5G access technologies, optics, core network technologies and virtualization. Some examples of the 5G Berlin work include the air-interface related topics as described in [8]. The *5G Dresden* [13] another German effort, focusing on research in the area of *Tactile Internet*, which refers to near-realtime interaction of people with physical and virtual objects [1]. The *5G Innovation Centre* [14] is a test network located in Surrey, UK. It focuses especially on new air-interface technologies.

Many test network system issues related to these types of networks are discussed in [21], and with our test network we aim to address also these issues. In comparison to the other 5G test networks such as the ones mentioned above, in 5GTN we provide a unified test network allowing a holistic overview for testing of devices and services, while supporting also linking to a larger nation-wide testbed concept as part of 5GTNF. We also provide integration with an advanced monitoring and data analytics infrastructure to provide means to not just run the tests but also to deeply analyze and understand the results and use them to guide and optimize towards better products and services.

In relation to different types of tests, various approaches for integrating performance and function tests, with e.g., behavioral models and their monitoring against large scale test data have been applied [20]. Currently, we perform this in a more qualitative way, as illustrated by our performance test scenario example in Section IV. However, if needed, our approaches in the test network could be extended to include this type of testing more formally as well.

The complexity of building a test environment supporting big data style data analytics and complex integrations of all required parts in test environments is discussed in [22]. In our test network, we aim to make the application of such techniques possible for all interested parties by providing and managing the complexities of the infrastructure as a service.

In relation to the types of traffic profiles and tests we support, many works have also targeted specific areas of the types of testing that we support in our test network, such as such as video Quality of Service (QoS) ([7]). We combine support for these as a holistic platform in our 5GTN.

III. 5GTN

Figure 1 shows a high-level picture of our test network from the testing technologies viewpoint. The macro cell provides extensive outdoor coverage for the relevant test scenarios. A set of small cells is deployed indoors to provide an indoor test environment. The backend system contains the full Evolved Packet Core (EPC) with all the associated components, along with network monitoring components deployed as Virtualized Network Function (NFV) instances on top of the OpenStack platform.

Supporting various types of actors (developers of infrastructure, services, end user devices, etc.) in their testing needs requires the ability to generate and execute tests at different levels of such a network, to collect extensive data about the performance of different elements in the network, and to be able to perform advanced analytics on them. A related architecture called ”Big Data Network Highway”, and associated challenges, is described in [10]. Expanding on the three layers presented in [10], we define several layers for the network, the end user device (e.g., phones, sensors, computers) layer, the (wireless) access (point) layer, the basic routing infrastructure, the core network (e.g., EPC, Content Delivery Network (CDN) servers) layer, and the datacenter layer (here test data and analytics architecture).

Not many actors have access to such complex environments, expertise on using all the advanced testing technologies, executing complex test setups, and performing the advanced analytics. We provide support for all these layers in terms of supporting diverse sets of protocols at the end user and access point layer, a full EPC core network, several test enabling application services such as video streaming CDN servers, IoT sensors and servers, and diverse test tools. Different combinations of these can be combined to create different test scenarios. For the more technical parts of these test network infrastructure components, we refer the reader to [9].

The analytics architecture follows the trend of what is commonly referred to as the Lambda architecture in big data processing [6]. This means we support both batch processing as well as stream processing. Using tools such as Apache Spark Core we provide batch processing support to analyze large scale datasets collected through the different test runs. Using tools such as Apache Spark Streaming and Apache Storm we provide support for near real-time stream processing. With batch processing we can provide support for long term-analysis, finding trends and correlations and doing similar analytics. They can be applied at any time, to explore new topics of interest in existing data sets as new things are learned and hypothesis need to be confirmed. With stream processing we provide support for interested parties to test real-time traffic optimization, network management and similar algorithms, as well as means to guide online test generation.

Apache Kafka in our case forms the ”Big Data Highway” for the measurement data, allowing us to effectively stream data from numerous distributed locations to several different and concurrent distributed processing systems. We call this in the following sections the data collection layer. For example, data is published from test tools, test targets, test generators, IoT gateways and similar system elements into this layer. Data is consumed (subscribed) from this layer by several analytics tools to perform real-time stream processing or to store the data for long-term historical batch analysis. Real-time stream processing systems can also publish additional data in the form of derived measures to the data collection layer, from which it can be further consumed by other stream processors and stored in long-term storage by batch storage consumers.

To enable execution of extensive test sets on top of this infrastructure, we also need to be able to generate various types of traffic and collect extensive monitoring data. We provide an extensive set of tools available in this environment, enabling monitoring of all deployed network elements, as well as of any test generation components and application servers that have interfaces to query relevant information.

The set of available test and monitoring tools is constantly

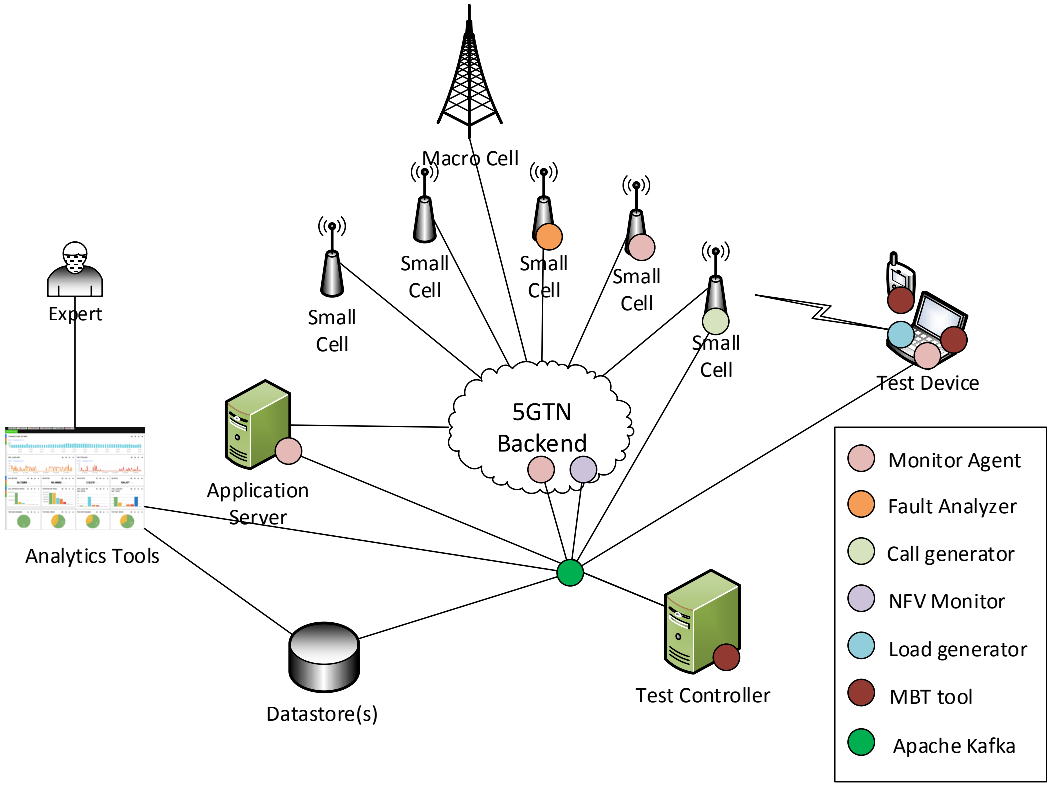

Figure 1. Test and Analytics Architecture overview

evolving, and includes:

- Model-based testing tool: Driving test scenarios to simulate realistic users on test devices, at single device/service as well as overall test scenario level (several devices/services).
- Virtualized (NFV) monitoring tools: Attached to Network Function Virtualization Infrastructure (NFVI) to provide monitoring of network traffic and parameters for the EPC
- IP traffic monitors: Collecting QoS measurements such as packet loss, latencies, ...
- Call generators: Large scale call traffic in the network
- Load generators: Large scale IP traffic in the network
- Test devices: Mobile devices, laptops, IoT devices, ...
- Data store: Used to collect test data such as control information and monitoring statistics
- Analytics tools: Test data analytics, both real-time and long-term historical

For the technical details on these, we again refer the reader to [9].

Our extensive set of monitoring tools enable us to collect data from different parts of the network, and these can be deployed on several network nodes at the same time. For example, QoSMet [18] is a tool capable of measuring detailed QoS network parameters between two endpoints. By deploying this on several of the endpoints at the same time, we can get a detailed view of the QoS for all the different elements. Similar data can be collected from the core network, and its interfaces, using monitoring tools deployed as Virtual Netword Function (VNF) elements with the EPC. Various similar tools can be deployed to monitor different properties as needed, and application specific monitoring interfaces can also be integrated into the data collection layer as needed. The overall data can be accessed through the data analytics layer.

Test traffic can be generated using different devices and services, both with real user equipment (phones, sensors, etc.) and large scale simulators. Specific types of large scale network data and specific service usage sequences can be generated at large scale using general computing resources as part of the network.

Tools such as model-based testing tools are used to generate traffic based on user profiles. These simulate real traffic and user activity in the network and on the service applications deployed on top it. They can be generated either based on recorded real traffic or simulated test models based on a model of the expected behaviour of the end user/sensor in question. The current main test scenarios/user profiles include:

- Video streaming
- Web browsing

- IoT sensors

High quality (e.g., 4k) video streaming is expected to be one of the major usage scenarios for high bandwidth consumption in the future, and provides a baseline for large-scale streaming. This type of stream is high-bandwidth consuming but can typically be scaled down in different QoS levels. IoT sensor traffic is expected to increase at large scale as the current IoT trend continues and the IoT products are increasingly deployed in practice and everything is connected to the network. This provides a specific type of traffic profile, where small burst of traffic are generated but they may be generated in large amounts by the numerous sensors deployed. Some of this traffic may also be of higher priority and must maintain high QoS, such as safety-critical measurements. Web browsing represents a current typical usage scenario that is used to provide a realistic background context for these.

Different profiles can be combined to provide test scenarios for different testing needs. For example, testing network infrastructure components may require generating varying loads of video, browser and IoT traffic, with varying network configurations and analyzing the results using multivariate analysis techniques to identify performance limits, optimization possibilities and problematic configurations and scenarios. From a different viewpoint, testing application services in the test network enables us to see their performance in different network loads, run functional and performance tests across the infrastructure and effectively pinpoint which issues are related to the application server or clients, and which are artefacts of the underlying infrastructure. Also in this context, our test services also enable combining different type of traffic, monitor the overall network, vary the service parameters, and observe and analyze all the results in depth. For end user devices, we can support a number of different protocols (as detailed in [9]), and their co-existence with various other devices and services in the network. In all these cases our aim is to provide a holistic view of the test environment, system under test, its environmental context, and broad analytics support.

IV. USE CASES AND TEST SCENARIOS

In this section, we give examples of the current usage scenarios we are running on the network, and using these to further develop it to be constantly more widely applicable for industrial testing and provide new testing services.

A. *IoT testing*

In this IoT test scenario, we have a Constrained application Protocol application (CoAP [16]) server deployed withing the network. Various actual sensor nodes available in the test laboratory are used to produce test traffic representing real IoT traffic in the network. This is passed through customized service gateway instances in the network edge (which also include the ability to calculate traffic statistics and publish them on the data layer), which forward the data over the test network to the CoAP application server. Several client instances are used over the network to scale up the test traffic over the gateways and other interfaces. More detailed test results for this case are available in [9], where they are shown as examples of measurements in the network, and we do not repeat them here. The important thing to note is how we can provide extensive support for various types of IoT sensor traffic and related protocols. Figure 2 illustrates the base concept of this type of a test case.

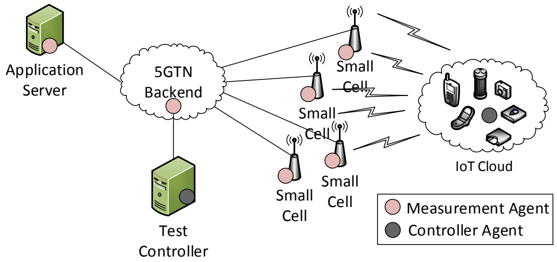

Figure 2. IoT case.

B. *Application server performance*

This test scenario is an example of a mobile service deployed with both mobile clients and an application server as part of the test network. The application server provides location tracking services for several moving nodes that it receives location data for. Any number of clients can be expected to connect to it at any time, and receive continuous streaming updates for sessions of different length. The data is provided as binary streams of protocol buffers messages over SSL encrypted sockets. The application server can be run either as part of the test network or as its own external cloud service (e.g., on Amazon EC2).

Figure 3 illustrates the beginning of one execution for this test scenario. The top row shows the frequency of updates as recorded by the application server (and directly reported to the data layer) in orange, and the average of the receiving frequencies observed at the clients (reported by the tester clients to the data layer) in green. The middle row graph shows the number of SSL errors observed by the clients when connecting to the server (orange for cumulative, green for per frame). The bottom row graph shows the combined number of live sessions by the tester clients during the test execution.

From Figure 3, we can see how at around 1500 concurrent sessions the service quality starts to degrade, with client average latencies starting to fluctuate, and how this fluctuation and number of errors increases as more sessions are initiated. In this case, a single error is an SSL handshake failure, where the client fails to establish a connection to the server, and these are recorded by the customized test client and also reported to the data layer. Running this test causes the system to fail practically all new sessions at around 4000 active sessions, showing a hard limit, where the test system stops after reaching an error threshold for number of failures in a continued sequence.

To better investigate the cause for the issues, we need to understand what is the status with the different system elements. Figure 4 shows the load on the application server, indicating that the server has no issues handing the traffic. This is also visible in top row graph in Figure 3, where the server (orange line) observes a constant result of providing the

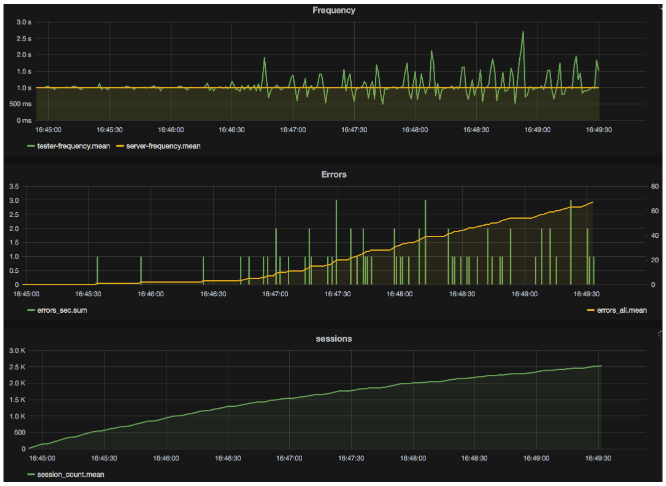

Figure 3. Errors observed in the performance test.

data to the clients on average at 1 second intervals as expected (while the clients show increasing fluctuation). The multiple multi-colored lines on the left side of Figure 4 are load per core, and the single green line on the right is overall load.

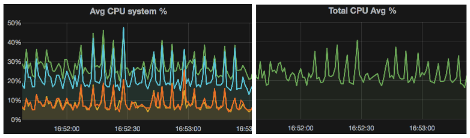

Figure 4. Server load.

We find further insight by looking at the router that is used to connect all the devices together in this case. This load is shown in Figure 5. This is the CPU load on the router collected using Simple Network Management Protocol (SNMP) probes, which again feed the measurements into the data layer. In relation to the server load shown in Figure 4, this has a 5 second resolution as it is the best resolution that the router can provide. This provides some added data analytics challenge due to different granularities of measurements, and automated correlation of failure thresholds to hitting specific limits. However, looking at these, we made the evaluation that the router overload is causing the errors in the test case.

We had high-confidence in this result from looking at the detailed resource use measurements and performance indicators we collected for all the system elements, including the application servers, test clients and router(s). The only one experiencing constant load issues towards the end is the router. For further investigations in this type of scenario we could modify the network configuration to alleviate such bottlenecks but in this scenario the result was enough to provide the needed results for this application server.

Figure 5 shows actually the end of one of these test sequence executions, where the sharp drop indicates the stopping of the test clients. A notable piece of information here is how the load in the router at the the end does not drop to zero but stays at around 25%. This is due to a set of baseline traffic providing a specific traffic profile for a combined test scenario. In this case it is a set of real users streaming YouTube video traffic on the same network, continuing throughout the test scenario and after.

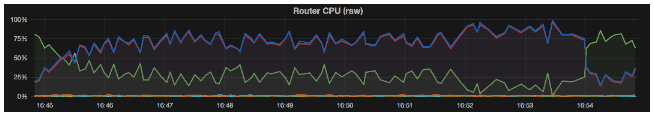

Figure 5. Router load.

Thus we can say that the server can handle larger numbers of concurrent clients but the network in this case would need additional resources. We see testing this type of a scenario as useful for scenarios such as large-scale IoT and mobile service deployments, or large events and similar locations where large crowds are gathering. The next iteration in this type of testing would be to add additional network elements, more distribution and continue to investigate the system limits in different configurations. However, we use this example here illustrate our point of using the test network to also provide a holistic view on the test network and system under test, and we leave these topics for future work.

C. *QoS mapping*

Figure 6 illustrates one of our executed test cases for measuring and mapping the QoS for some of our network devices. This one shows the outdoor area surrounding the test network site, with green parts showing where the observed streaming QoS is above a specified threshold value. Red indicates values where the value is below the threshold. The values can be collected using our tools (e.g., QoSMet [18]) for QoS measurement and mapping these to a map of the area of interest.

Similar measurements can also be provided for indoor areas (e.g., indoor small cell coverage). This is a service we can perform as part of our test environment to provide insight into and compare, what is the strength of the signal using various equipment under different configurations. It also gives us insight into how we might expect the QoS to behave for different services being tested in the network when they are mobile through the network. Such QoS values can further be incorporated into the analytics results.

V. DISCUSSION

The examples we have provided here are only intended to illustrate potential use cases for the test network and the benefits of a holistic tests and analytics architecture. As mentioned in the architecture section, in addition to these basic test execution scenarios described above, The overall testing process with the test network is intended to start with the visual data exploration phase described in the previous section, followed by tuning the testing as new things are learned about the system and its performance in the network, and proceeding to deeper analytics enabled by more advanced algorithms that can be implemented on top of the big data analytics platforms.

As mentioned in Section III, we have also integrated support for these big data analytics platforms in the form of tools such as Apache Spark and Storm. In addition to large-scale historical analysis, output from these tools can be used

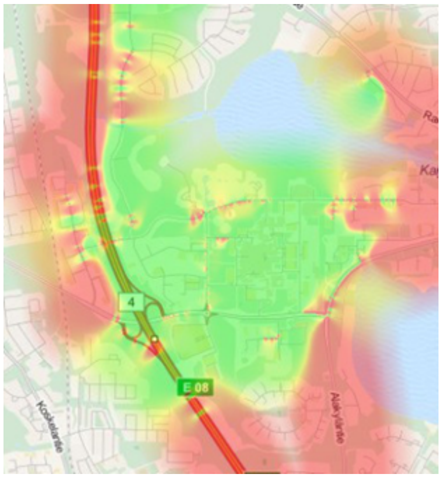

Figure 6. QoS map.

to provide real-time input for algorithms in different domains such as network management or to guide test generation towards interesting goals when specific statistical effects or impacts are observed. Properties of interest to study this way include varying parameters in the test network elements, varying the traffic profiles in relation to these, and analyzing the measurement data to find relations. Besides this current level of integration, one of our long term goals in this relation is the ability to to further link this with more advanced test automation support to enable automated variation, collection and analytics. As our test network constantly evolves and we execute increasingly complex test scenarios we plan to address these as part of our future work.

The application service performance test example given in Section IV focused on the overall performance aspect. In addition to such non-functional properties, we find such tests can also support functional testing through our broad data collection and analytics support. In our performance test example we illustrated how we can integrate any types of service specific measures into the system. In this case they were the application server session counts, and test client and server internal processing latencies. Besides the performance measure, these were also used during testing to identify lingering sessions causing resource leaks in different execution and load scenarios (visible as the live session count not dropping after the tests and high error rates).

In relation to our goals for the test network set in Section III, the architecture and example test scenarios show how we can use this type of a test architecture to support various test goals such as the ones described at the beginning of this paper. We can introduce network elements, including both actual hardware and virtualized (NFV) software appliances, into the network, test their functionality and impact separately and as part of the larger network. Similarly, we can provide a testing platform for next-generation software-based services making use of the features enabled by these fifth generation (5G) networks, and provide a holistic view on their functionality and performance in relation to different elements. All together, we see the 5G test network as providing a holistic innovation platform for next generation services.

Integration of big data monitoring and analytics provides some specific issues to be addressed. Fast reactions and real-time analytics need special attention as telecommunications systems are real-time systems where situations happen quickly and need fast reactions also from analytics and the operations it can trigger. Building such a heterogeneous test network as described here also requires integrating multitude of heterogeneous devices and services, as well as all of the data they produce. This requires extensive integration over different interfaces, and means to combine them.

It also requires integrating the diverse data formats to the diverse set of analytics tools. In our case, we have used data ingestion components collecting the data from different sources and transforming them to the shared binary protocol format. However, besides this basic transformation, the type of data and its meaning requires extensive experience on multitude of domains both within the telecommunications domain and application domains (each different). Within the telecommunications domain alone, specific components alone (e.g., EPC or base stations) can produce hundreds or thousands of parameters. Identifying the relevant elements and their combinations from all this requires diverse expertise from numerous players and takes a lot of time, as well continuous evolution. Our set of diverse project partners is one of the enablers for addressing these needs.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we described the testing technologies in the 5G test network. The 5GTN is an ongoing project focused on building a platform for next generation services. With a comprehensive test generation, monitoring, and analytics architecture it enables extensive testing of both related devices and software, as well as provides a platform for building innovative services targeting next generations of networks. We continue our work and expand the network and its services, including addressing the issues identified and discussed in this paper.

In the future we will continue evolving the network as new 5G technologies become available and as we learn new things from the testing performed on the network. We will also investigate additional real application services as part of the network and how to evolve the services to support added use cases. For the analytics part, emerging technologies such as edge computing provide interesting options to distribute and optimize the overall analytics architecture.

ACKNOWLEDGMENT

This paper is carried out in a 5GTN project, which is partially funded by Tekes, Finnish funding agency for innovation. The authors would like to thank the whole consortium for all the help and cooperation.

REFERENCES

[1] G. P. Fettweis, ”The Tactile Internet: Applications and Challenges,”, IEEE Vehicular Technology Magazine, vol. 9, no. 1, March 2014, pp. 64-70.

[2] J. Gozalvez, ”5G Tests and Demonstrations [Mobile Radio],”, IEEE Vehicular Technology Magazine, vol. 10, no. 2, June 2015, pp. 16-25.

[3] A. Gupta and R. K. Jha, ”A Survey of 5G Network: Architecture and Emerging Technologies,”, IEEE Access, vol. 3, 2015, pp. 1206-1232.

[4] T. Kanstrén and J. Perälä, ”Testing Technologies in Finnish 5G Test Network”, ETSI User Conf. on Advanced Automated Testing (UCAAT), 20-22, October, 2015.

[5] M. Latva-Aho, A. Pouttu, A. Hekkala, I. Harjula, and J. Mäkelä, Small Cell Based 5G Test Network (5GTN), 12th Intl. Symp. on Wireless Comm. Systems, Brussels, Belgium, 25-28, August, 2015.

[6] N. Marz and J. Warren, ”Big Data: Principles and Best Practices of Scalable Realtime Data Systems”, Manning publications, 2015.

[7] S. Moon, J. Yoo, and S. Kim, ”Exploiting Adaptive Multi-interface Selection to Improve QoS and Cost-Efficiency of Mobile Video Streaming”, IEEE Int'l. Conf. on Mobile Services, June/July, 2015, pp. 134 - 141.

[8] T. Wirth, et al., ”An Advanced Hardware Platform to verify 5G Wireless Communication concepts”, IEEE 81st Vehicular Technology Conf. (VTC Spring), May, 2015, pp. 1-5.

[9] E. Piri, et al., ”5GTN: A Test Network for 5G Application Development and Testing”, European Conf. on Networks and Communications (EuCNC), Athens, Greece, 2016.

[10] X. Yi, F. Liu, and H. Jin, ”Building a Network Highway for Big Data: Architecture and Challenges”, IEEE Network, July/August, 2014, pp. 5-13.

[11] http://5gtn.fi/, ”5GTN - 5G Test Network”, Oulu, Finland, [retrieved: April, 2016].

[12] http://5gtnf.fi/, ”5G Test Network Finland”, Finland, [retrieved: April, 2016].

[13] http://5glab.de/, ”5G Lab Germany”, Dresden, Germany, [retrieved: April, 2016].

[14] http://www.surrey.ac.uk/5gic, ”5G Innovation Centre”, Surrey, UK, [retrieved: April, 2016].

[15] http://www.5g-berlin.org/, ”5G Berlin”, Berlin, Germany, [retrieved: April, 2016].

[16] Z. Shelby, K. Hartke, and C. Bormann, ”The Constrained Application Protocol (CoAP)”, IETF Request for Comments: 7252, 2014.

[17] ETSI,”Mobile-Edge Computing (MEC); Service Scenarios”, GS MEC-IEG 004, rev. V1.1.1, Nov., 2015.

[18] J. Prokkola, ”Qosmet Enabling passive QoS measurements”, [Online], Available: http://www.cnl.fi/qosmet.html, 2016, [retrieved: April, 2016].

[19] P. Gimenez, B. Molina, C. E. Palau, and M. Esteve, ”SWE Simulation and Testing for the IoT”, IEEE Int'l. Conf. on Systems, Man, and Cybernetics (SMC), October, 2013, pp. 356 - 361.

[20] X. Che and S. Maag, ”A Passive Testing Approach for Protocols in Internet of Things”, IEEE GreenCom & iThings/CPSCom, August, 2013, pp. 678 - 684.

[21] P. Rosenkranz, M. Wählisch, E. Baccelli, and L. Ortmann, ”A Distributed Test System Architecture for Open-source IoT Software”, Workshop on IoT Challenges in Mobile and Industrial Systems, May, 2015, pp. 43-48.

[22] M. Yesudas, G. Menon, and S. Nair, ”High-Volume Performance Test Framework using Big Data”, 4th Int'l. Workshop on Large-Scale Testing, January/February, 2015, pp. 13-16.