Service availability in the NFV Virtualised Evolved Packet Core

Andres Gonzalez, Pål Grønsund, Kashif Mahmood
Telenor Research
Telenor ASA
Email: {andres.gonzalez, pal.gronsund,
kashif.mahmood}@telenor.com

Bjarne Helvik, Poul Heegaard, Gianfranco Nencioni
Department of Telematics
Norwegian University of Science and Technology
Email: {bjarne, poul.heegaard,
gianfranco.nencioni}@item.ntnu.no

***Abstract*—Network Function Virtualization (NFV) promises to transform the way telecom providers design and operate networks and network services. Virtualized Evolved Packet Core vEPC is one of the Network Function Virtualization NFV use cases that has got most of attention, where dependability is a major concern. In the traditional EPC, functions are deployed in proprietary network elements with proven characteristics, e.g., a defined availability, and corresponding guarantees. Hence, network operators have a firm basis for the design of a robust mobile core network. On the other hand, in the vEPC, network operators face a more challenging environment, where functions, subsystems and requirements are interrelated in a more complex manner. Hence, the assessment of the robustness of the network, and the design to meet dependability requirements becomes hard. In order to address this challenge, we provide initial guidelines and modeling tools to assess system availability in vEPC scenarios, and identify the most relevant factors to be considered in this process.**

***Index Terms*—NFV; virtual EPC; fault tolerance; mobile core reliability;availability modeling.**

I. INTRODUCTION

The mobile networks have evolved significantly, moving from 1G to the current 4G networks. Common to all generations is the *proprietary, mission specific and topologically fixed nodes* that provide the network functions. This has resulted, amongst other issues, in high CAPEX and OPEX for the operators and long lead times for changes in the service delivery. One solution to address this challenge is to virtualize the network functions and provide them on Commercial-Off-The-Shelf (COTS) hardware, i.e. to introduce network function visualization (NFV), where the aim is to run mobile network functions as software instances on commodity servers. An immediate advantage of such an approach is the ability to scale the resources up and down according to the traffic demand. This approach is highly relevant for network functions that do not require specific hardware (ASICs). The evolved packet core (EPC) in 4G networks consists of many such network functions and the potential benefits of a virtualized EPC (vEPC) are huge [2].

One of the main concern for the Telecom operators, and the focus of this paper, is the system availability in the vEPC. A firm control of this aspect is mandatory in order to consider an production scale implementation. To establish availability guarantees, responsibility domains and modeling tools in a vEPC environment is much more complex than in the traditional network design. The main questions addressed in this paper are: *How to assess the availability of a vEPC?* and *What are the main availability concerns to be considered?* For this, we study the potential failure sources in a vEPC environment, discuss the advantages and disadvantages of the vEPC from the dependability point of view, and finally, we propose a model based in stochastic activity networks, to assess the availability of a vEPC.

There has been a considerable effort in dealing reliability and availability aspects in NFV, a significant part of it within the ETSI Industry Specification Group NFV REL WG [5]. They have established a set of requirements and specifications for developing robust NFV based services and have defined a number of techniques and mechanisms to ensure reliability and availability in an operational virtual environment. The work presented in this paper complements the defined specifications and requirements, by providing a Stochastic Activity Network (SAN) system availability model, and a study of the main parameters to be considered in such assessment.

Virtualization is a technology that started in the 1970's [10]. It has acquired significant importance for computational applications with the development of cloud computing, and now, it is extending its scope to the network domain. One of the arguments motivating the implementation of virtualized environments is the independence from the physical hardware, since any affected virtual application may be executed on any working server [15]. Fault tolerance is a key aspect in a virtualized environment wherein two most relevant techniques are active-replication and standby-replication [3], [4].

The availability of the datacenter network is fundamental for the vEPC robustness. There are several studies that have dealt with this. For e.g [9] presents a large-scale analysis of network failures [9]. They distinguish between two types of failures, link and device failures and observe that low-cost, commodity switches (such as ToRs and AggS) are highly reliable, but middleboxes (such as load balancers) experience a high number of software faults. Datacenter networks can be highly reliable, although issues as network control, redundancy and topology design should be carefully planned. Internal network failures in data centers, and network failures between data centers, have a significant impact on the virtualized

applications. In [17] insights on the impact on cloud services are provided.

This paper is organized as follows. Section II describes the NFV framework and illustrates the main features of the vEPC. In Section III, we define the main failure sources in a vEPC scenario, and analyze its pros and cons in comparison with the conventional EPC. Section IV provides SAN model to assess the system availability of the vEPC. Finally, Section V concludes the paper.

II. NFV and the Virtualised Evolved Packet Core

A. Network Function Virtualisation (NFV)

NFV is about transforming the way network operators design and operate networks and network services by applying standard IT virtualization technology to consolidate many specialized network equipment types onto industry standard high volume servers, switches and storage units [7]. The NFV architecture depicted in Figure 1 defines the main architectural constituents, the functional building blocks and interfaces between these.

The NFVI (Network Functions Virtualisation Infrastructure) provides the virtual resources required to support the execution of the Virtual Network Functions (VNFs). It includes Commercial-Off-The-Shelf (COTS) hardware, accelerator components where necessary, and a software layer which virtualizes and abstracts the underlying hardware.

The VNF is the software implementation of a network function which is capable of running over the NFVI. It is the entity corresponding to a main function of today's network nodes, which are now expected to be delivered as pure software free from hardware dependency. VNF can be accompanied by an Element Management System (EMS), as long as it is applicable to the particular function, which understands and manages an individual VNF and its peculiarities.

The NFV Management and Orchestration (MANO or M&O in Figure 1) covers the orchestration and lifecycle management of physical and/or software resources. The NFV orchestrator in MANO is responsible for the global resource management and on-boarding and lifecycle management of the network services. The VNF manager on the other hand is responsible for the lifecycle management of the VNF instances. Finally the Virtual Infrastructure Manager (VIM) in MANO controls and manages the NFVI resources. The NFV MANO also interacts with the OSS/BSS, which allows NFV to be integrated into an already existing network-wide management context containing physical network functions.

The entire NFV system is driven by a set of metadata describing the Service, VNFs and Infrastructure requirements, so that the NFV MANO systems can act accordingly. These descriptors along with the Services, VNFs, Infrastructure and MANO components can be provided by different industry players.

B. Virtualisation of the Evolved Packet Core (vEPC)

The EPC is the all-IP current mobile core network architecture, for providing converged voice and data on a 4G

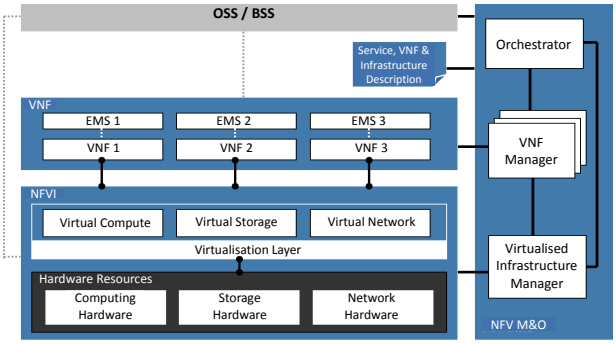

Fig. 1. ETSI NFV ISG architectural framework [7]

Long-Term Evolution (LTE) network. It is responsible for handling the traffic flows from millions of subscribers while providing high availability guarantees. The EPC consists of a set of functions that usually run on propriety and specialized hardware boxes. When traffic or signalling exceeds a certain threshold a new specialized hardware box must be installed.

The three main logical components of the EPC, which mostly are implemented as physical components, are:

-Mobility Management Entity (MME): Is the control node which processes the signalling between the User Equipment (UE) and the core network, selects the Service Gateway for a UE, and it is fundamental in the bearer activation/deactivation process. The functions supported by the MME are related to bearer management, connection management, and to interworking with other networks.

-Service Gateway (S-GW): All user's IP packets are transferred through the S-GW. It serves as the local mobility anchor for the data bearers when the UE moves between eNodeBs. It also retains the information about the bearers when the UE is in idle state and temporarily buffers downlink data while the MME initiates paging of the UE to re-establish the bearers.

-Packet Data Network Gateway (P-GW): Is responsible for IP address allocation of UE, as well as QoS enforcement and flow-based charging. It can be defined as the interface between the LTE network and other packet data networks.

The fact that the main functionalities of the EPC are implemented on expensive proprietary hardware that in some cases may present scalability and flexibility challenges, has motivated the study of vEPC as a potential solution. It is however interesting that software costs usually dominate considerably over hardware costs for EPC. But, when considering the flexibility in implementing new vEPC solutions due to the hardware abstraction and independence brought by with NFV, it is believed that more rapid innovation for and a lower barrier to implement vEPC solutions will lead to significant lower software costs in the near future. Another important point is the potential to increase operational efficiency when using a single NFV platform for all vEPC components and other VNFs in the mobile core network. The automation will enable automatic scaling in and out of hardware capacity for the vEPC, which will lower the operational costs and

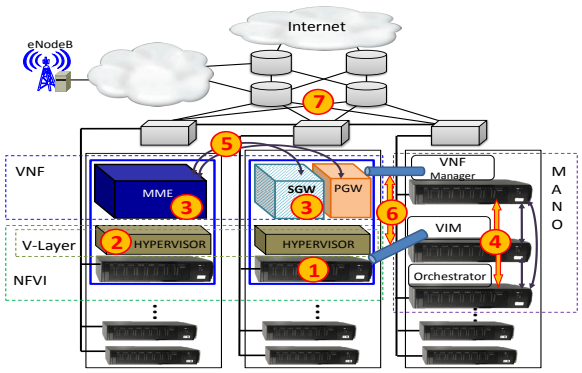

Fig. 2. Generic NFV-vEPC scenario and potential failure sources.

power consumption since idle hardware can be powered down automatically. Finally, it will also be beneficial to spin up new vEPC nodes for specific use cases such as machine-to-machine or for specific customers such as large enterprises. Finally, network resources can be saved by exploiting the flexibility in placement of the vEPC functions in intelligent ways, which according to the scenario in [19] could lead up to 25% network resource savings.

The vEPC is one of the most popular use cases specified by the ETSI-NFV Industry Specification Group [6]. It proposes to implement the MME, S-GW and P-GW as VNFs on COTS servers, running on top a virtualization layer within the ETSI NFV framework. This implies the need of a MANO infrastructure that can administrate the tasks needed for the appropriate performance and management of the VNFs. An abstraction of a potential NFV-vEPC scenario is presented in Figure 2, illustrating the main concepts described here.

III. Service Availability Issues in the vEPC

In order to guide the providers design and implementation of the vEPC, we analyze different sources of failure, and their dependability implications. This section gives more details on how the potential failures can affect the vEPC availability.

A. Failure Sources in the vEPC

Figure 2 illustrates a generic NFV-vEPC scenario and its potential failure sources (yellow circles), numbered as follows:

1) Hardware failure in the COTS servers: As every hardware component, COTS servers will fail. It is recognized that COTS hardware has a higher failure intensity than legacy telecom hardware serving network functions, see for instance [16]. To deal with hardware failures, legacy servers have fault tolerant mechanisms tailored to meet the fault handling requirement of the telecommunication domain, e.g. the availability requirement of five nines, and a mature technology that has been improved over generations of systems. However, to deal with failures of COTS hardware and deliver comparable availability levels, other fault tolerant techniques can be used, e.g. active replication, load-sharing, etc. based on mechanisms implemented in a generic platform, e.g., [1], [14].

2) Software Failures in the Hypervisor: A virtualization environment needs a hypervisor in order to map virtual functions with the respective hardware resources required. The hypervisor is managed from a centralized architecture known as VIM, but it needs to be separately installed on each hardware component used. The hypervisor may be prone to software failures which may affect individual processors or, since the hypervisors on the individual processors are tightly logically coupled, they may affect larger parts of the system wide virtualization layer.

3) Software Failures of the VNF themselves: The VNF is the software with all the logic that allows the implementation of the different parts of the EPC. Like in all type of software, the VNF may also contain logical faults which may cause failures. In principle the code used for implementing those functions is similar for the EPC and the vEPC. Some vendors have hardware independent implementations of this functionality already, and hence it is considered to be a reasonable assumption that the failure rate will not change much by moving these functions from EPC to vEPC. However, the impact of the failures may be different in terms of down times, error propagation and the set of functionality affected.

4) Failures of the MANO: The proper operation of the MANO depends on the hardware, software and even the connectivity between MANO servers, since the Orchestrator, VIM and VNF Manager may be deployed on different physical servers. There are two views concerning failures of the MANO. The first, and pretty optimistic, is that they won't affect the running operations, but inhibit any new operation. Once the vEPC is set, in principle the MANO does not need to be consulted. However, in case it is needed, the absence of the MANO could be catastrophic. The other, and in the authors opinion more realistic view, is that since the MANO may change/influence huge parts of the EPC through misoperation and error propagation, the consequences of MANO failures may become catastrophic.

5) Logic Connection between different EPC functions: Since vEPC functions will most likely be distributed among different physical servers, there is the need of a physical and logical connectivity among each parts as illustrated in failure source number 5 in Figure 2. This connectivity is prone to physical network failures and logical virtual connectivity, and since the topology in a datacenter may be more complex, this part should be carefully planned.

6) Logic Connection between the MANO and the VNF/NFVI: From the connectivity point of view the failure nature is similar to the one presented in the previous case (5). However, the consequences are the same as when a system failure in the MANO has occurred..

7) Failures in the Distribution and Core Networks: This will influence the system availability significantly and must be carefully taken into account. In [11], [13], it is observed high levels of path fluctuation, and considerable number of failures that involve a single link. However, the faults and the effect of the corresponding failures are the same for both EPC and vEPC, and will therefor not be discussed further in this

paper.

B. System Availability Pros and Cons of the vEPC

As listed in the previous section, the virtualization of the EPC involved several challenges with respect to the system availability. It has two important failures sources that did not exist in the EPC. The first one is with regard to the hypervisor while the second source of failure is the MANO. This is compounded by the fact that COTS hardware is likely to have a higher failure intensity than the legacy telecom hardware [16]. This, combined with potential challenging issues related to coordination of the separated software and hardware solutions, strongly motivates the need for a study of the system availability of the vEPC.

New tests and fault tolerance techniques need to be designed and implemented for the vEPCs. It is still an open question how these should be designed in order to meet the strict system availably requirements of NFV. However, it is clear that it poses a more challenging ecosystem that demands the cooperation of multiple stakeholders, where a clear definition of the roles are essential.

It is important to highlight that the MANO is conceived as a very powerful entity that may influence huge parts of the vEPC. So although the probability of having catastrophic failures in the MANO might be small, it would have a tremendous impact that should be addressed.

One the other hand, vEPC could be a potential solution to the scalability and flexibility issues that appear because of the EPC implemented on proprietary hardware. One of the main advantages of virtualizing the EPC is the ability to restore affected VNF on any available hardware device.

In contrast to conventional EPC solutions, where a very high availability is guaranteed by providing highly reliable (and expensive) devices, the virtualized EPC may realize high availability by design smart techniques for fault tolerance and load sharing on the virtualized platform. In conventional EPCs, it is difficult and time-consuming to change and improve the dependability once designed and implemented. In the vEPC, changes and improvement are much easier and quicker to realize, for example for fault tolerance. This give the providers the necessary instruments to constantly improve and adapt their design to the current requirements and needs.

A concrete example of the mentioned vEPC advantage is the handling recovery after a disaster. In conventional EPCs, the operators prepare by proactive measures. However, it is hard to predict disasters and their consequences. Implementing a mobile core solution from zero is a challenging task that can be more manageable by taking advantage of the flexibility and scalability properties of NFV.

IV. Service Availability Modeling and Assessment

In this section we provide an illustrative alternative of how to model the system availability in vEPC by use of Stochastic Activity Networks(SAN) [18], a particular extension of stochastic Petri nets, composed of *places, activities, input gates* and *output gates*. The simulations and models presented here, were developed in the Möbius software tool [8].

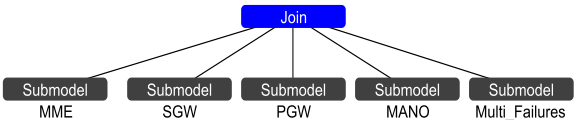

Fig. 3. Composed model for the vEPC availability

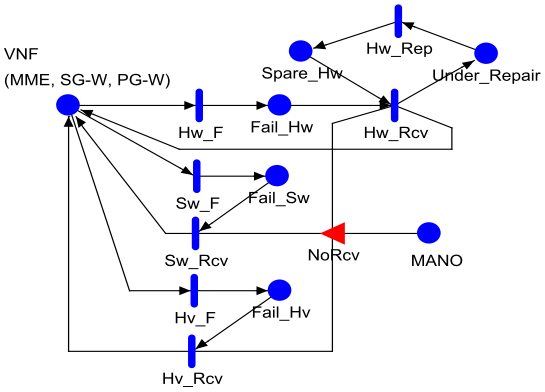

Fig. 4. SAN model of the VNF (MME, SG-W or PG-W)

The SAN model for the system availability of the vEPC is divided in five atomic models: `MME`, `SG-W`, `PG-W`, `MANO` and `Multi-Failure` as presented in Figure 3. In the model, we have concentrated upon the delivery of the network functions to get an insight into the dependability issues related to the delivery of these. To predict the availability of end user services, it is of course necessary to complement the model to included availability provided by the datacenter network topology, which is a complementary research such as the one presented in [12].

Figure 4 presents the SAN model for the VNF, which applies for the MME, SG-W and PG-W, since their dependability behavior is similar (vulnerable to hardware, software and hypervisor failures). Although the software failure intensity may differ due to the difference in the number of lines needed to deploy each VNF function.

The place VNF has an initial Marking (number of tokens) that represents the number of COTS servers (processing units) in charge of running the VNF. This model assumes a cluster based implementation that allows the modeling of high availability fault tolerance techniques, where after the failure of one processing unit, the remaining units can take responsibility of continuing the VNF operations. The VNF will be considered unavailable when the number of tokens on place VNF won't be able to cover the load demands of the system.

Firing of activity `Hw_F` represents a hardware failure in a COTS server. We assume that it follows a negative exponential distribution with failure rate λ_H. After firing, one token is taken from the `VNF` and put in place `Fail_Hw`. A token in `Fail_Hw` demands actions, since it represents a reduction in the VNF capacity, and hence a potential system unavailability.

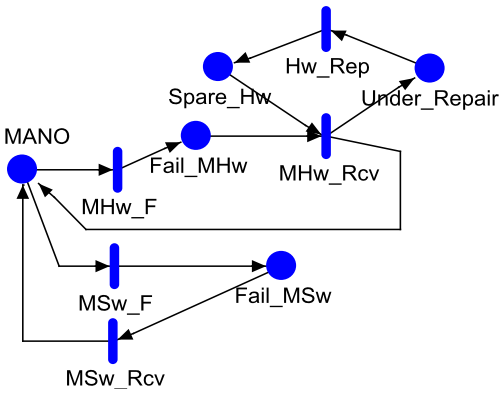

Fig. 5. SAN model of the MANO

The activity Hw_Rcv (Hardware recovery) is in charge of making the respective corrective actions, by taking a spare server to restore the affected VNF. We assume that Hw_Rcv follows a negative exponential distribution with intensity μ_H, and its firing depends on three factors. i) There are tokens in Fail_Hw (hardware units to be repaired). ii) There are spare servers available. iii) The MANO is available, and hence the recovery is possible. The last condition is modeled by using the *shared place* MANO and the input gate NoRcv. The Firing of Hw_Rcv takes one token from Fail_Hw and put it on the Under_Repair place, and at the same time, it takes one token from Spare_Hw and put it on the VNF place. Finally, the Hw_Rep activity represents the off-line repair of the affected hardware, in order to guarantee enough Spare servers in Spare_Hw.

Software (Sw) and hypervisor (Hv) events are modeled as follows. Activities Sw_F and Hv_F represent software and hypervisor failures. We assume that they follow a negative exponential distribution with failure rate λ_S and λ_Y respectively. After the firing of Sw_F/Hv_F, one token is taken from VNF and it is moved to place Fail_Sw/Fail_Hv. The recovery from software or hypervisor failures is modeled using the activity Sw_Rcv (Software recovery) and Hv_Rcv (Hypervisor recovery) respectively. The firing of Sw_Rcv/Hv_Rcv will follow a negative exponential distribution with intensity μ_S / μ_H, and it depends on two factors. i) There are tokens on the respective failure places Fail_Hv/Fail_Sw. ii) The MANO is available, and hence recovery actions can be executed. This is also controlled by using the input gate NoRcv and the shared place MANO. The Firing of Sw_Rcv/Hv_Rcv takes one token from Fail_Sw/Fail_Hv and put it on the VNF place.

Figure 5 illustrates the SAN model of the MANO. As explained before, failures in the MANO in principle won't affect the running operations. However, they put the system in a vulnerable state, which from the modeling point of view is challenging to capture. When the MANO is not operational, the failure recovery activities rcv previously mentioned can not be executed. In order to model this scenario, we use the input gate NoRcv and the *shared place* MANO presented in the

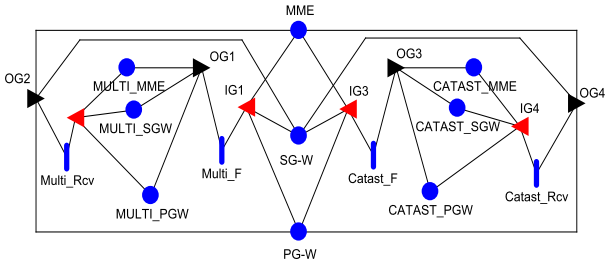

Fig. 6. SAN model for multiple and catastrophic failures (Multi Failure)

SAN models of the VNF (Figure 4) and the MANO (Figure 5).

The initial number of tokens in the MANO place also considers a redundant cluster based implementation that allows the modeling of high availability fault tolerance techniques. The MANO will be considered unavailable when its number of tokens won't be able to cover the current Management and Orchestration Demands.

Hardware failures are in principle the same than in the VNF (same COTS servers assumed), represented by the firing of activity MHw_F. We assume that it follows a negative exponential distribution with failure rate λ_H. After firing, one token is taken from the MANO place and it is moved to place Fail_MHw. The importance here, lies in the shared place MANO that verifies if the tokens at MANO are enough to consider it available. In case of not being enough, the input gate NoRcv will inhibit the recovery processes in the VNF SAN model. The activity MHw_Rcv is in charge of taking the hardware recovery actions in a similar way like in the VNF model. After firing, it puts one token in the MANO place.

Finally, software failures work in the same way, by having the respective failure and a recovery activities MSw_F and MSw_Rcv, keeping track of the MANO place as for MANO hardware failures.

The last SAN atomic model to be described is the one in charge of modeling multiple and catastrophic failures. As described in the previous section, there are events that may generate failures in multiple VNF instances. In addition, catastrophic events that may bring the entire vEPC down may happen (e.g. natural disasters or severe erroneous operations at the MANO). To model those situations, we use the atomic model Multi-Failure presented in Figure 6. In the center of the figure, the three VNF places (MME, SG-W and PG-W) can be appreciated, which are the same *shared places* presented for each VNF SAN model.

On the left side of Figure 6 is the activity Multi_F, which is an activity that considers the event of failures with consequences on multiple VNF instances, by taking multiple tokens from the MME, SG-W and PG-W, using the input and output gate IG1 and OG1, depending on the specific vEPC architecture to be modeled. The places MULTI_MME, MULTI_SGW and MULTI_PGW keep track of the VNFs affected by failures with multiple consequences. The activity Multi_Rcv is in charge of recovering these kind of failures, and through the use of the input and output gate IG2 and OG2,

the respective number tokens are returned to the corresponding affected VNFs.

Finally, the right side of Figure 6 considers catastrophic failures, using the activity `Catast_F`. In this case, the mechanism is very similar to the left side model, but here it is assumed that after fairing of the `Catast_F` activity, all tokens from the VNFs are taken. In the same way, activity `Catast_Rcv` models the recovery process executed after a catastrophic event. After the `Catast_Rcv` firing, all tokens are taken back to the corresponding VNF places.

V. Conclusion

This paper illustrates the most relevant failure sources in a vEPC scenario, presents an analysis of its advantages and disadvantages from the dependability point of view, and propose a model to assess system availability. One of the main issues that this paper highlights is the need of new procedures and mechanisms to assure highly dependable vEPC systems that covers from type of test to be implemented to the business language itself. An interesting paradox is the fact that most of the observed features that may be considered as dependability disadvantages in the vEPC, are the same properties that may represent an advantage under certain conditions (e.g. hardware agnostic property). Therefore, if proper effort is put on individual and joint testing, redundancy planning, and design of smart fault tolerance techniques, the vEPC may deliver acceptable levels of availability.

References

[1] Amir, Y., Danilov, C., and Stanton, S. A low latency, loss tolerant architecture and protocol for wide area group communication. In *Dependable Systems and Networks, 2000. DSN 2000. Proceedings International Conference on* (2000), pp. 327–336.

[2] Basta, A., Kellerer, W., Hoffmann, M., Hoffmann, K., and Schmidt, E.-D. A Virtual SDN-Enabled LTE EPC Architecture: A Case Study for S-/P-Gateways Functions. In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for* (Nov 2013), pp. 1–7.

[3] Cully, B., Lefebvre, G., Meyer, D., Feeley, M., Hutchinson, N., and Warfield, A. Remus: high availability via asynchronous virtual machine replication. In *5th USENIX Symposium on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2008), NSDI'08, USENIX Association, pp. 161–174.

[4] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W. Dynamo: amazon's highly available key-value store. *SIGOPS Oper. Syst. Rev. 41*, 6 (Oct. 2007), 205–220.

[5] ETSI NFV ISG. Network Function Virtualisation (NFV) - Resiliency Requirements. *Tech. Rep. V1.1.1.* (Jan. 2015).

[6] ETSI NFV ISG. Network Function Virtualisation (NFV) - Use cases. *Tech. Rep.* (Oct. 2013).

[7] ETSI NFV ISG. Network Functions Virtualisation NFV - Network operator perspectives on industry progress. *Tech. Rep.* (Oct. 2013).

[8] Gaonkar, S., Keefe, K., Lamprecht, R., Rozier, E., Kemper, P., and Sanders, W. H. Performance and Dependability Modeling with Möbius. *SIGMETRICS Perform. Eval. Rev. 36*, 4 (Mar. 2009), 16–21.

[9] Gill, P., Jain, N., and Nagappan, N. Understanding network failures in data centers: measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM* (2011), pp. 350–361.

[10] Goldberg, R. P. Survey of virtual machine research. *IEEE Computer Magazine* (1974).

[11] Labovitz, C., Ahuja, A., and Jahanian, F. Experimental study of internet stability and backbone failures. In *FTCS* (1999), IEEE Computer Society, pp. 278–285.

[12] Liu, V., Halperin, D., Krishnamurthy, A., and Anderson, T. F10: A fault-tolerant engineered network. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2013), nsdi'13, USENIX Association, pp. 399–412.

[13] Markopoulou, A., Iannaccone, G., Bhattacharyya, S., Chuah, C. N., Ganjali, Y., and Diot, C. Characterization of failures in an operational IP backbone network. *IEEE/ACM Transactions on Networking 16*, 4 (Aug. 2008), 749–762.

[14] Meling, H., Montresor, A., Helvik, B. E., and Babaoglu, O. Jgroup/ARM: A distributed object group platform with autonomous replication management. *Software: Practice and Experience 38*, 9 (25 July 2008), 885–923.

[15] Mergen, M. F., Uhlig, V., Krieger, O., and Xenidis, J. Virtualization for high-performance computing. *SIGOPS Oper. Syst. Rev. 40*, 2 (Apr. 2006), 8–11.

[16] Mo, L. Reliability of NFV using COTS hardware. *ZTE COMMUNICATIONS 12*, 3 (September 2014), 53–61.

[17] Potharaju, R., and Jain, N. When the network crumbles: An empirical study of cloud network failures and their impact on services. In *Proceedings of the 4th Annual Symposium on Cloud Computing* (New York, NY, USA, 2013), SOCC '13, ACM, pp. 15:1–15:17.

[18] Sanders, W., and Obal, W. Dependability evaluation using UltraSAN. In *Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers., The Twenty-Third International Symposium on* (June 1993), pp. 674–679.

[19] Yousaf, F., Lessmann, J., Loureiro, P., and Schmid, S. Softepc; dynamic instantiation of mobile core network entities for efficient resource utilization. In *Communications (ICC), 2013 IEEE International Conference on* (June 2013), pp. 3602–3606.