Availability Modelling of Software-Defined Backbone Networks

Gianfranco Nencioni*, Bjarne E. Helvik*, Andres J. Gonzalez†, Poul E. Heegaard* and Andrzej Kamisiński‡
*Department of Telematics, NTNU, Norwegian University of Science and Technology, Trondheim, Norway
{gianfranco.nencioni, bjarne.helvik, poul.heegaard}@item.ntnu.no
†Telenor Research, Telenor ASA, Trondheim, Norway
andres.gonzalez@telenor.com
‡Department of Telecommunications, AGH University of Science and Technology, Kraków, Poland
kamisinski@kt.agh.edu.pl

***Abstract*—Software-Defined Networking (SDN) promises to improve the programmability and flexibility of networks, but it may also bring new challenges that need to be explored. The main objective of this paper is to present a quantitative assessment of the properties of SDN backbone networks to determine whether they can provide similar availability to the traditional IP backbone networks. To achieve this goal, we have completed the following steps: i) we formalized a two-level availability model that is able to capture the global network connectivity without neglecting the essential details; ii) we proposed Markov models for characterizing the single network elements in both SDN and traditional networks; iii) we carried out an extensive sensitivity analysis of a national and a world-wide backbone networks. The results have highlighted the considerable impact of operational and management (O&M) failures on the overall availability of SDN. High O&M failure intensity may reduce the availability of SDN as much as one order of magnitude compared to traditional networks. Moreover, the results show that the impact of software and hardware failures on the overall availability of SDN can be significantly reduced through proper overprovisioning of the SDN controller(s).**

I. Introduction

During the recent years, Software-Defined Networking (SDN) has emerged as a new networking paradigm based on an idea of programmable network devices in which it is assumed that the forwarding plane is decoupled from the control plane [1], [2]. Although programmable networks have been studied for decades, SDN is experiencing a growing success because it is expected that the ability to change network protocols easily and add new services and applications will foster future network innovation which is limited and expensive in current legacy systems. For example, SDN has a number of potential advantages relative to the current technology with respect to the functionality adapted to service requirements, reduced CAPEX and OPEX, resource optimization, ease of maintenance and operation, etc.

In [3], the following potential advantages of SDN were pointed out: *centralised control*, control logic is not distributed as in traditional IP network but centralised in an additional network element (i.e. the SDN controller); *simplified algorithms*, the centralised algorithms are easier than the distributed ones; *commodity network hardware*, expensive application-specific hardware is not longer required; *eliminating middle-boxes*, such as firewalls, that are substituted by network applications; *enabling the design and deployment of third-party applications*, easy to be developed on top of the controller.

A simplified sketch of the SDN architecture presented in the IRFT RFC 7426 [1] is shown in Figure 1. The control plane and the data plane are separated from each other. In addition, the control plane is logically centralized in a software-based controller ("network brain"), while the data plane is composed of network devices ("network arms") that forward packets. The control plane includes both northbound and southbound interfaces. The northbound interface provides a network abstraction to network applications such as an implementation of a routing protocol, firewall, load balancer, anomaly detection module, etc., southbound interface (e.g., OpenFlow) standardizes the information exchange between the control and data planes.

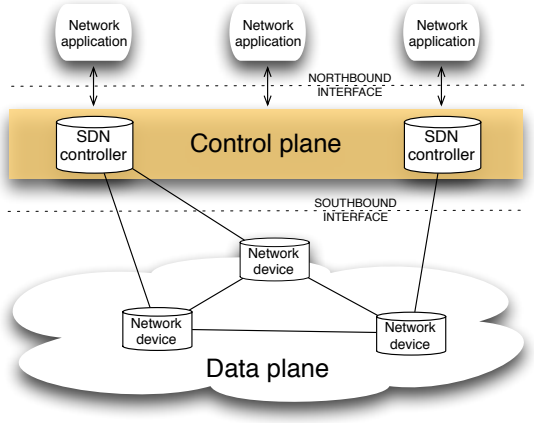


Fig. 1. SDN architecture (excluding the management plane).

The possible introduction of SDN in the backbone network is an extremely important issue. Some features introduced by SDN are of particular interest in this context. However, we should also require that the dependability of an SDN-based backbone network is at least as good as in the case of the current generation of backbone networks. Similar requirements have also been considered during the previous shifts in networking technology, e.g., the introduction of computer control,

which in fact introduced the so called *five nine availability* requirement (two hours of accumulated downtime during 40 years of operation [4]). This is a strong requirement, as the current technology was designed to be inherently survivable due to its distributed nature [5] and has been constantly improved for several decades.

In this context, the dependability (availability and continuous operation) of SDN has received little attention. It was suggested that the centralized and automated management may improve the dependability (e.g., [6]). However, several questions have been raised by network operators and researchers concerning the dependability issues introduced by SDN due to the logical centralization, increased complexity, interdependence between the forwarding plane and the control plane, and other factors [7], [8]. In particular, SDN introduces a set of new vulnerabilities and challenges related to the dependability, compared with traditional networking [8]:

- consistency of network information (user plane state information) and controller decisions;
- consistency between physically-distributed SDN controllers in the control plane;
- increased failure intensities of (commodity) network elements;
- compatibility and interoperability between general purpose, non-standard network elements;
- interdependence between the path set-up task in network elements and monitoring of the data plane in the control plane;
- load balancing (to avoid performance bottlenecks) and fault tolerance in the control plane have conflicting requirements.

The objective of this paper is to compare the overall network availability that may be achieved with SDN with that of a traditional IP-routed network, and to investigate under which parametric conditions one solution is better than the other. Thus, we introduce a two-level modelling approach in which the top layer captures the structural properties of networks, while the bottom layer reflects the dependability characteristics of different network elements/subsystems according to the hardware, software and operational models. To maintain similarities and establish a parametric relationship, the models of network elements/subsystems are developed for both considered types of networks. Extensive sensitivity analysis of a national and a world-wide backbone networks has been carried out to highlight the impact of the components composing the SDN controller on the overall availability of both SDN and traditional IP network.

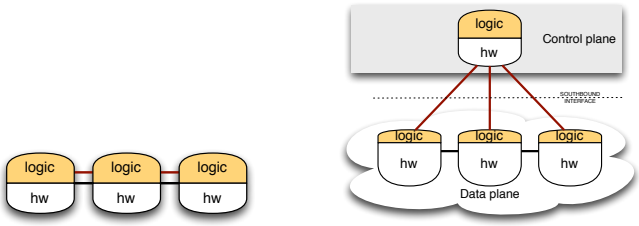
The remainder of this paper is organized as follows: in Section II, we introduce and describe the considered research problem. Then, the two-level hierarchical model to evaluate the availability of SDN is presented in Section III. Finally, in Section IV, we discuss the results of the sensitivity analysis based on the selected set of parameters that will potentially affect the dependability of SDN.

II. Dependability issues in SDN

Traditional IP networks consist of a set of interconnected nodes that include both the data and control planes. Each network node is a complex device that has the functionality of both data forwarding and networking control. To increase the availability and performance of such devices, manufacturers have focused on specialised hardware and software over the past few decades.

As discussed in the previous section, SDN has the potential to change the principles of networking and to enhance network flexibility. This implies moving the control logic from the network nodes to a (virtual) centralised controller, and to open up the controllers to a third party via an API (northbound interface), as illustrated in Figure 2. The transition from a *distributed* network with a focus on establishing and maintaining the connectivity between peering points, to a *centralised* network with a focus on QoS and resource utilisation, will potentially lead to much simpler network nodes with less control logic. The centralised control logic, such as the routing decisions, might be simpler and can even be made more advanced, without making it more complex compared to the distributed solution. The controller has the potential to set up data flows based on a richer set of QoS attributes than in traditional IP networks.

From a dependability point of view, the network nodes might be simple because their only task is now to forward data in the data plane and send information to a network controller and thus the network nodes tend to have a higher availability. At the same time, the controller becomes a very critical component, due to centralization and complexity that requires to be carefully designed. Therefore, the dependability of SDN needs a comprehensive assessment, in particular in comparison to the traditional IP network and considering all the factors that can affect it.



(a) Current IP networks: Distributed logic embedded in the forwarding engine of network nodes.

(b) SDN: Logically centralized control logic combined with simplified network elements.

Fig. 2. Software-Defined Networking moves the control logic from distributed network elements to a logically centralized unit.

Although dependability is rising as an important issue to make SDN a success [9], to the best of our knowledge, limited work on modelling the dependability in SDN has been performed. In [8] the potential dependability challenges with SDN are discussed. In [10], a model of SDN controllers is developed. In [11], the occurrence probability for different failure types is studied. In [12], an hierarchical availability

model is proposed. Similarly to us, their approach is composed of two levels but in both levels they use different modelling approaches: a Reliability Graph and a Stochastic Reward Net. Furthermore their approach is used to compute the availability of a particular application in a small SDN case study. Our approach is instead focussed on evaluating the availability of the network connectivity even in large-scale case study. To the best of our knowledge, our paper is the first attempt of evaluating the overall availability of a SDN backbone. A two-level availability model is presented, which is an extension of a model used in an example in [13]. The proposed model allows us to study how the SDN paradigm modifies the overall availability of the network relative to the traditional distributed IP network and analyse which factors dominate in this new scenario.

III. Two-level availability model

In this section a two-level model is introduced to evaluate the dependability of SDN in a global backbone. In particular, the dependability is measured in terms of steady state availability, henceforth referred to as availability. The two-level hierarchical availability modelling approach consists of:

- *Structural* model of the network topology;
- *Dynamic* model of network elements.

The approach seeks to avoid the potential uncontrolled growth in model size by compromising the need for modelling details and at the same time modelling a (very) large scale network. The detailed modelling is necessary to capture the dependencies that exists between network elements, and to described multiple failure modes that might be found in some of the network elements and in the controllers. The structural model disregards this and assumes independence between the components considered, where a component can be either a single network element with one failure mode, or a set of elements that are interdependent and/or experience several failure modes with an advanced recovery strategy. For the dynamic models we can use a Markov model or Stochastic Petrinet (e.g., Stochastic Reward Network [14]). For the structural model we can use reliability block diagram, fault trees, or structure functions based on minimal cut or path sets.

In the following section, we will demonstrate the use of this approach. We introduce the structural model of IP legacy systems and SDN backbone networks. For the dynamic level, we propose Markov models of SDN and traditional network elements (in particular links, IP routers, SDN switches, and SDN controllers). Finally, we explain how to combine the structural and dynamic models.

A. Structural model

As introduced in Section II, one of the consequences of moving the control logic from distributed to centralised is the increasing of the "connectivity" required to consider the network available. For this reason we focus on the dependability issues for the control plane by investigating the reactive SDN mode. More formally, given a traffic that needs to be routed from a origin node o to a destination node d, the following connections must be considered in SDN:

- *flow triggering*: on arrival of a new flow, a path for the trigger message that should be sent from o to the SDN controller ;
- *network state update and route directives*: a path from the SDN controller to each node in the path from o to d;
- *forwarding*: forwarding path from o to d.

The first two connections are related to the *control plane* in SDN, they concern about the connectivity among the controller and the nodes in the data network. The last connection is associated to the *data plane* and concerns about the connectivity of the forwarding nodes.

For traditional (legacy) IP networks the structure of the data plane and control plane is the same, and identical to the structure of the data plane in SDN.

The structure of the control plane in SDN adds the separate controllers to the data plane structure, and hence increase the complexity compared to the control plane in traditional IP networks.

The critical parts of the connection between the traffic origin and destination (and between the controller and any network node in SDN) can be determined using structural analysis based on either *minimal-cut sets* or *minimal-path sets* [15]. In this paper we use minimal-cut sets:

Definition 1: Minimal-cut set - A system is failed, if and only if, all the subsystems in a minimal-cut set are failed, even if all the other subsystems that are not in the set are working.

The minimal-cut sets form the basis for a *structure function*.

Definition 2: Structure function - Each max-term of the structure function expressed in a minimal product-of-sums form corresponds to a minimal-cut set.

In the case studies of Section IV, further details on how to apply the structural analysis to a SDN scenario will be presented. In [8] a small scale example illustrates the implementation those techniques.

B. Dynamic model

In order to evaluate the availability of each network element, we develop Markov models of each of the links, IP routers, SDN switches, and SDN controllers.

1) Link: The model of a link is assumed to be dominated by physical link failures. Therefore, a simple two-state Markov model is used. The links are either up or down due to hardware failure. We use the same model for both traditional network and SDN. Given failure rate λ_L and repair rate μ_L, the availability of a link is $A_L = \frac{\mu_L}{\lambda_L + \mu_L}$. This model is assumed for each of the link components in the structural model. We don't know the geographical location of the nodes and therefor the distance between them either, which implies that the length of the links connecting the nodes in the network can't be determined. Hence, in our case studies we have to assume that the link failure rate is not dependent of the link length. Note that in general the failure rate is expected to be proportional to the length of the link.

TABLE I
STATE VARIABLES FOR TRADITIONAL IP ROUTER

state	up/down	description
OK	up	System is fault free
O&M	down	Operation and Maintenance failure
CHW1	up	Hardware failure of one controller
CHW2	down	Hardware failure both controllers
COV	down	Coverage state, unsuccessful activation of the stand-by hardware after a failure; manual recovery
FHW	down	Permanent hardware failure in forwarding plane
FHWt	down	Transient hardware failure in forwarding plane
SW	down	Software failure

2) Traditional IP router: The model of a traditional router is depicted in Figure 3, where the states are defined in Table I. In the model we focus on the router functionalities and the related failure sources, each component of the router has not been considered because it would be dependent on a particular router architecture. In any case, we assume 1+1 redundancy of the controller hardware, which is a common best practice in any architecture. Multiple failures are not included in the model since they are assumed to be less frequent and will probably not have significant impact on the expected accuracy of the approach.

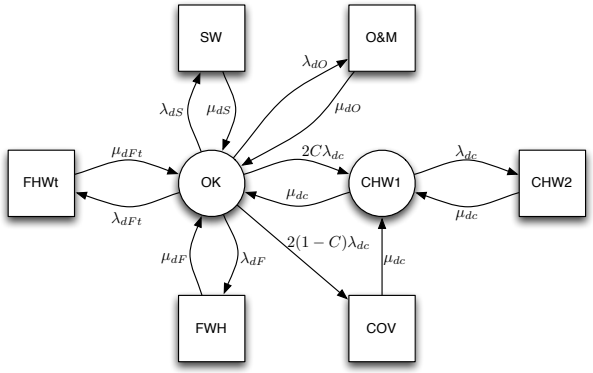


Fig. 3. Markov model of a traditional IP router

All the model parameters are defined in Table II. Note that for sake of simplicity we have assumed homogeneous equipment. The table includes the numerical values used in the case studies and that are inspired by and taken from several studies [16], [17], [18].

3) SDN switch: Figure 4 shows the model of the switch in an SDN, which is significantly simpler than the router in a traditional network. The states related to the control hardware failures are not contained in this model, since all the control logic is located in the controller. O&M associated with the SDN switch has been also omitted because we assume that the complexity of the O&M operations done on a single switch is likely to be small relative to a router and globally in the controller. The software is still present but its failure rate will be very low since the functionality is much simpler.

TABLE II
MODEL PARAMETERS FOR THE IP NETWORK WITH NUMERICAL VALUES USED IN THE CASE STUDIES

intensity	[time]	description
$1/\lambda_L = 4$	[months]	expected time to next link failure
$1/\mu_L = 15$	[minutes]	expected time to link repair
$1/\lambda_{dF} = 6$	[months]	expected time to next permanent forwarding hardware failure
$1/\mu_{dF} = 12$	[hours]	expected time to repair permanent forwarding hardware
$1/\lambda_{dFt} = 1$	[week]	expected time to next transient forwarding hardware failure
$1/\mu_{dFt} = 3$	[minutes]	expected time to repair transient forwarding hardware
$1/\lambda_{dC} = 6$	[months]	expected time to next control hardware failure
$1/\mu_{dC} = 12$	[hours]	expected time to repair control hardware
$1/\lambda_{dS} = 1$	[week]	expected time to next software failure
$1/\mu_{dS} = 3$	[minutes]	expected time to software repair
$1/\lambda_{dO} = 1$	[month]	expected time to next O&M failure
$1/\mu_{dO} = 3$	[hours]	expected time to O&M repair
$C = 0.97$		coverage factor

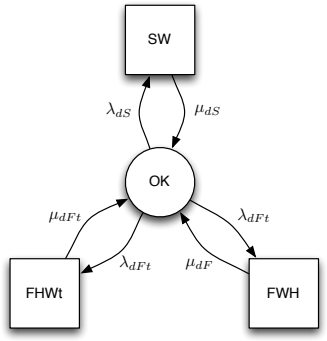


Fig. 4. Markov model of a SDN switch

Table III describes the parameters for modelling the SDN switch.

All SDN parameters are expressed relative to the parameters for the traditional network (Table II). In an SDN switch, the failure/repair intensities of (permanent/transient) hardware failures are the same because failures with the same cause, have the same intensities in both models. However, we assume that the software on an SDN switch will be much less complicated than on a traditional IP router because the control logic has been moved to the controllers, and we have set the

TABLE III
MODEL PARAMETERS FOR THE SDN SWITCH

intensity	description
$\lambda_F = \lambda_{dF}$	intensity of permanent hardware failures
$\mu_F = \mu_{dF}$	repair intensity of permanent hardware failures
$\lambda_{Ft} = \lambda_{dFt}$	intensity of transient hardware failures
$\mu_{Ft} = \mu_{dFt}$	restoration intensity after transient hardware failures
$\lambda_{sS} = 0$	intensity of software failure

failure rate to zero, for the sake of simplicity.

4) SDN controller: The model of the SDN controller is composed of two sets of states. One set captures the software and hardware failures. The second set captures the O&M and coverage failures in combination with the hardware states of the system. We have assumed that the SDN controller is a cluster of M processors and the system is working, i.e., possesses sufficient capacity if K out of the M processors are active, which means that both software and hardware are working. To represent this scenario, each state is labelled by four numbers $\{n, i, j, k\}$, where n is the number of active processors, i the number of processors down due to hardware failures, j the number of processors down due to software failures, and k the state of both coverage and O&M functionality ($k = 1$ if O&M mistake, $k = 0$, if not). Figure 5 shows the *outgoing* transitions from a generic HW and SW failure state $\{n, i, j, 0\}$ (i.e. lower part) and from a generic O&M/coverage failure state $\{n + j, i, 0, 1\}$ (i.e. upper part). The main assumptions of the model are:

- single repairman for a hardware failure;
- load dependency of software failure when the system is working, $\lambda_S(n) = \lambda_S/n$, where the meaning of λ_S is explained in more detail in Section IV;
- when the entire system fails, only processors failed due to hardware failures will be down until the system recovers;
- load independence of software failure when the system has failed, $\lambda_S(n) = \lambda_S$, since the remaining unfailed processors are working at the full capacity.

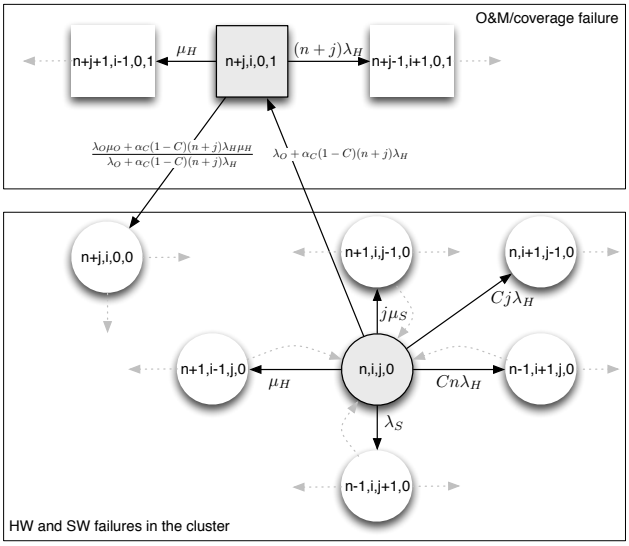


Fig. 5. Generic states of the model of SDN controller

The parameters the SDN controller model are listed in Table IV.

In an SDN controller, all failure rates are N-times larger than in the traditional network, where N is the number of network nodes (10 in the national network and 28 in the world-wide network). This is because we assume that the SDN needs roughly the same processing capacity and amount of hardware than in the traditional network. Therefore, the failure intensity is assumed to be proportional to N, and of the same order of magnitude as the total failure intensity of the traditional distributed IP router system. For the hardware failures the total failure intensity is divided by the number of needed processors $K = \lfloor 0.8 \cdot M \rceil$, where $M = N$ is the total number of processors. Moreover, we added the proportionality factors α_H, α_S, α_O, and α_C, which will be exploited in our sensitivity analysis.

TABLE IV
MODEL PARAMETERS FOR THE SDN CONTROLLER

intensity	description
$\lambda_H = \alpha_H\ \lambda_{dC}\ N/K$	intensity of hardware failures
$\mu_H = \mu_{dC}$	hardware repair intensity
$\lambda_S = \alpha_S\ \lambda_{dS}\ N$	intensity of software failures
$\mu_S = \mu_{dS}$	restoration intensity after software failure
$\lambda_O = \alpha_O\ \lambda_{dO}\ N$	intensity of O&M failures
$\mu_O = \mu_O$	rectification intensity after O&M failures

C. Merging the two levels to evaluate the network availability

The next step is to obtain the overall network availability by merging the structure function and minimal-path sets defined in Section III-A with the availability of the network elements computed by using the Markov models in Section III-B.

The *inclusion-exclusion principle*, which is a technique to obtain the elements in the union of finite sets, is applied. Using the inclusion-exclusion principle on the structure function, we can write the system availability as the probability of the union of all the minimal-path sets:

$$A_S = P\left(\bigcup_{i=1}^{n} Q_i\right) = \sum_{k=1}^{n} (-1)^{k-1} \sum_{\substack{\emptyset \neq I \subseteq [n] \\ |I|=k}} P\left(\bigcap_{i \in I} Q_i\right), \tag{1}$$

where $Q_1, Q_2, \ldots, Q_n$ are the minimal-path sets (see Section III-A), and $P(Q_i)$ is the probability of set Q_i.

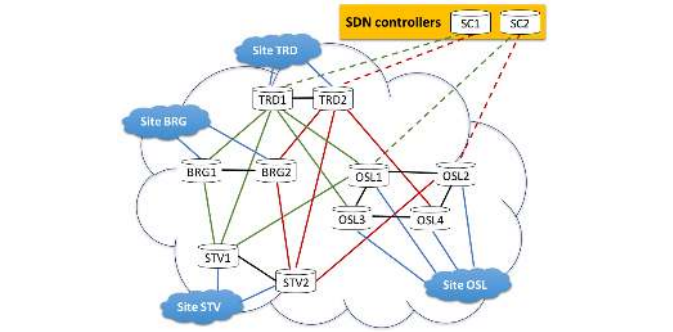
To compute the probability of the intersection of minimal-path sets, we assumed that the failures in each network element are independent thus we just need to know the availability of each network element. To this end, we can calculate the element availability by using the proposed Markov models (see Section III-B).

IV. NUMERICAL EVALUATION

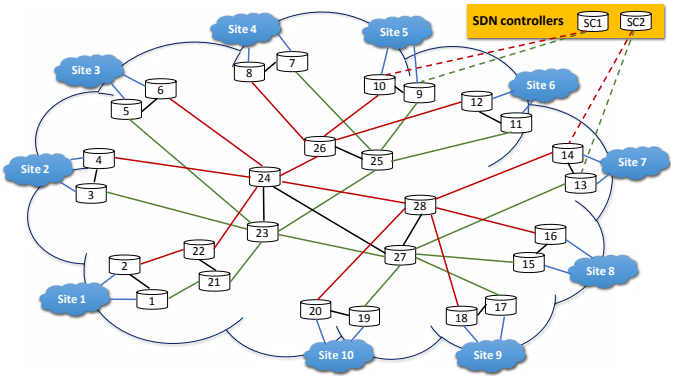
In this section, first we present the national and world-wide backbone networks that will be used as case studies. Secondly, all the parameters related to the two-level model are presented. Successively, the impact of the different failure sources of the SDN controller is analysed. Finally, the overprovisioning of the cluster that composes the SDN controller is evaluated.

A. Case studies

We consider two case studies: a national backbone network, which is a simple but realistic scenario, and a world-wide



(a) National



(b) World-wide

Fig. 6. Topology of the backbone networks

backbone network, which allows us to investigate the availability behaviour when the size of the network topology increases. The related network topologies are depicted in Figure 6. The national backbone consists of 10 nodes and 4 sites, three sites (TRD, BRG, and STV) are connected to two nodes and one site (OSL) is connected to 4 nodes. The world-wide backbone is composed of 28 nodes and 10 sites, all the sites are connected to two nodes and 8 nodes are not connected to any site (i.e. they are core nodes).

In both networks there are two dual-homed SDN controllers (SC1 and SC2). This choice is just an example and has been taken accordingly a study on SDN controller configuration [19], where impact of the location, number, and homing of SDN controllers on the national backbone network is investigated.

One of the objectives of the study is to compare the availability of SDN with a traditional IP network with the same topology of network elements (SDN switches and IP routers). In this study the assumptions are the following:

- nodes, links, and controllers in the system may fail;
- the network is working (up) when all the sites are connected;
- for SDN, at least one controller must be reachable from all nodes along a working path.

Given that the node/link capacity is not considered, the last assumption can be reformulated as follows: at least one controller must be reachable from each site.

B. Evaluating impact of SDN components

The target of this section is to evaluate *which* and *how* the different components of SDN influence the availability of the network. In particular, we investigate the impact of the different failure sources of the SDN controller on the overall availability. For this purpose we use the α_X factors where $X = S, H, O, C$ (see Figure 5 and Table IV), which affect the intensity of the related failure sources (software, α_S, hardware, α_H, O&M, α_O, and coverage, α_C) and are defined as follows:

- $\alpha_H = \dfrac{\lambda_H}{N/K\ \lambda_{dC}}$;
- $\alpha_S = \dfrac{\lambda_S}{N\ \lambda_{dS}}$;
- $\alpha_O = \dfrac{\lambda_O}{N\ \lambda_{dO}}$.

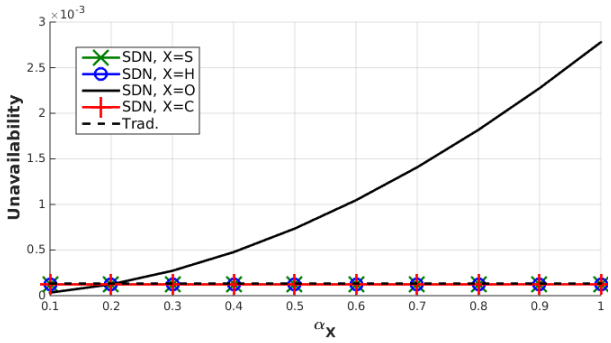
Given the above definition, if $\alpha_X = 1$, the failure intensity, λ_X, in the SDN controller corresponds to the cumulative failure intensity of the elements in the whole distributed IP network, $N\ \lambda_{dX}$. If $\alpha_X < 1$, the failure intensity in the SDN controller is lower than the cumulative failure intensity of the elements in the whole distributed IP network.

Figure 7 shows the behaviour of the overall unavailability for both national and world-wide network when α_X are varied from 1 to 0.1, so that the related failure intensities are reduced of one order of magnitude. In particular, for the world-wide network since the trends are very similar to the ones in the traditional network, to highlight the differences we consider the *unavailability ratio*, i.e. the ratio between the unavailability in the world-wide network and the unavailability in the national network.

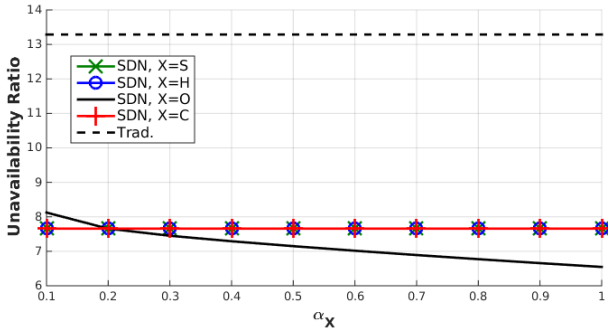
In the figure the availability of traditional networks does not change by varying the diverse α_X because they only affect the SDN controller. For the SDN , the availability is reduced of about two orders of magnitude when α_O increases from 0.1 to 1 and needs to be lower than 0.2 in the national network and than around 0.3 in the world-wide network to have the same availability performance than the traditional IP network. Variation of the other α_X yield far less impact. This indicates that the O&M failure is likely to be the most critical part in the dependability of SDN.

We have further investigated this behaviour by setting to zero the O&M failure intensity, $\lambda_O = 0$. The results, which are not included for sake of brevity, show that, if the availability is still insensitive to software failures, the sensitivity by the variation of hardware and coverage failures is increased and has the same increasing trend for both. This behaviour is caused by the transition rate to the O&M and coverage failure state ($\{n+j, i, 0, 1\}$ in Figure 5) that, given the definition of λ_H, is equal to $\lambda_O + \alpha_C(1-C)(n+j)\alpha_H\lambda_{dC}N/K$ and thus equally depends on α_C and α_H.

Finally, it needs to be noted that the national and the world-wide backbone networks have the same trends, the only difference is that the values are shifted by approximately one order of magnitude (~ 13 for the traditional network, $\sim 7-8$ for the SDN). The motivation is that, if the national network requires 10 connections (6 site-site and 4 one controller-site),

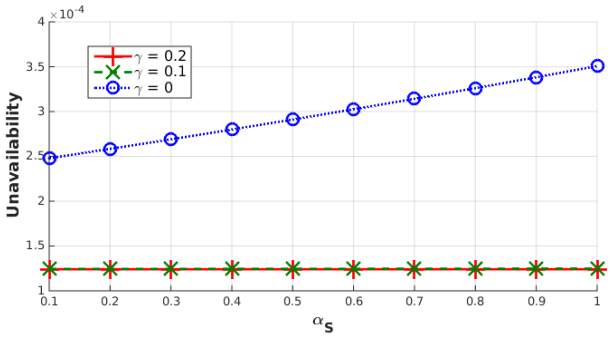


(a) National network

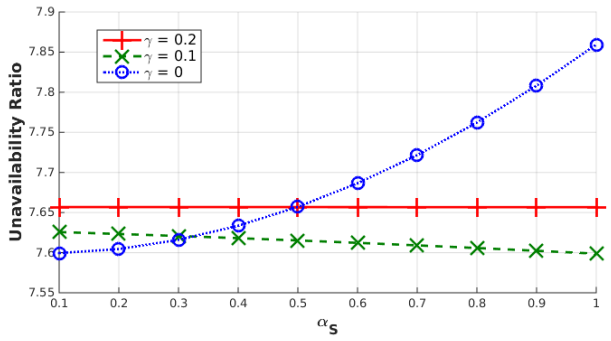


(b) World-wide network vs. national network

Fig. 7. Unavailability of traditional network and of SDN by varying α_S, α_H, α_O, and α_C



(a) National network



(b) World-wide network vs. national network

Fig. 8. Unavailability of SDN by varying α_S ($\alpha_H = 1$, $\alpha_O = 0.2$, and $\alpha_C = 1$)

the world-wide network needs 55 connections (45 site-site and 10 one controller-site). For validating this conclusion we have computed the availability in a version of the world-wide network where the sites 2, 4, 6, 8, 9, and 10 and the related access nodes have been removed and thus the number of sites (1, 3, 5, and 7) is the same as in the national network. The results show that indeed the achieved availability has approximately the same order of magnitude(~ 7 for the traditional network, $\sim 2-3$ for the SDN).

C. Evaluating SDN controller overprovisioning

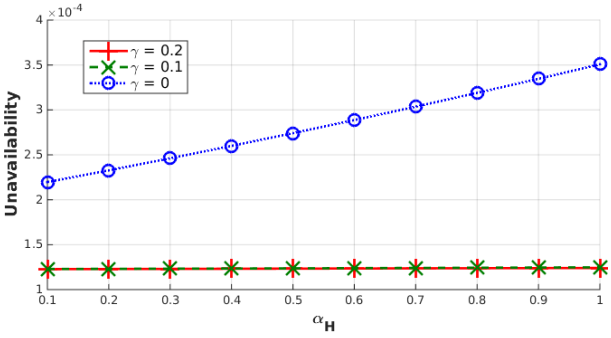
An open question from the previous evaluation is why the overall network availability is not affected by the software and hardware failures.

For this purpose we defined and evaluated the *overprovisioning ratio* as $\gamma = \frac{M-K}{M}$. In particular, given the total number of processors M in the cluster composing the SDN controller, we varied the number of processors in the cluster so that if they fail the global system does not fail K according to $K = \lfloor (1-\gamma)\ M \rceil$.

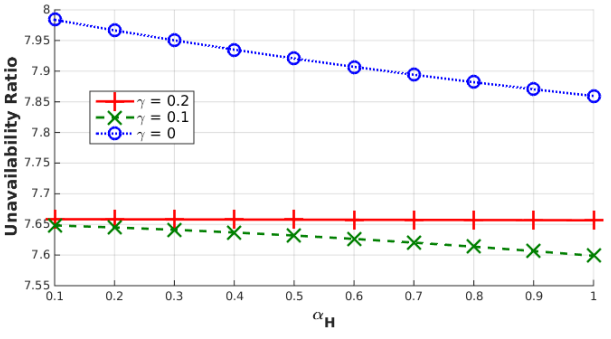
Figures 8, 9, and 10 show the trends of varying the α_X, $X = S, H, C$ respectively, when γ is decreased from 0.2 to 0.1 and 0.

The results highlight how the unavailability is almost the same when γ is equal to 0.2 and 0.1, but increases when there is no overprovisioning in the cluster, i.e. $\gamma = 0$.

Furthermore, Figures 8 and 9 show that the overall availability starts to depend on the software and hardware failure

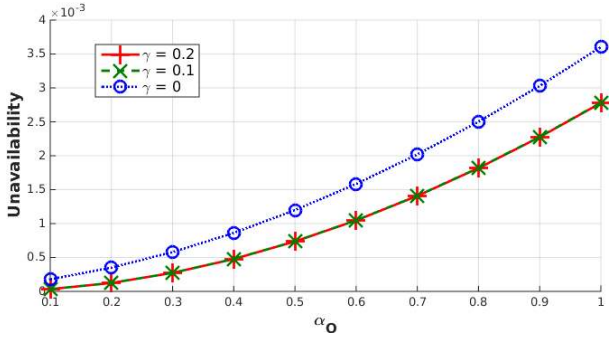


(a) National network

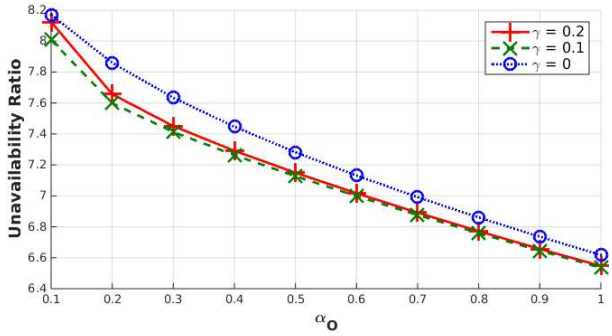


(b) World-wide network vs. national network

Fig. 9. Unavailability of SDN by varying α_H ($\alpha_S = 1$, $\alpha_O = 0.2$, and $\alpha_C = 1$)



(a) National network



(b) World-wide network vs. national network

Fig. 10. Unavailability of SDN by varying α_O ($\alpha_S = 1$, $\alpha_H = 1$, and $\alpha_C = 1$)

intensity for $\gamma = 0$, since the failure of one single processor causes the failure of the whole SDN controller.

Moreover, Figures 10 confirms that O&M is the failure source that has the most significant impact on the overall network availability.

Finally, we did include the figure related to the coverage failure because it shows a constant increasing trend for all scenarios and thus confirms that increasing the coverage failure intensity does not have impact on the availability.

V. Conclusions

In this paper a quantitative assessment has been performed to investigate which are the properties in SDN backbone networks that affect the overall availability. A two-level availability model that includes structural and dynamic models has been formalized and for the dynamic level Markov models of the single network elements have been proposed. A sensitivity analysis has been carried out in both a national and a world-wide backbone. The main outcomes are the following.

- *O&M* failures are likely the most important source of the failure of the whole SDN backbone, an increment of one order of magnitude of the O&M failures intensity coincides with an increment on two order of magnitude of the overall network availability;
- high intensity of *software* and *hardware* failure (likely given by complex software and commodity hardware) does not significantly affect the overall SDN availability, if there are enough spare processors in the cluster composing the SDN controller;
- the impact of *coverage* failures on the network availability is dominated by the O&M failures.

In conclusion, for achieving an overall availability in a SDN backbone comparable (or even better) than in a traditional IP backbone, a reduction of the O&M failures and a proper design of the SDN controller are needed.

References

[1] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-defined networking (SDN): Layers and architecture terminology," Internet Research Task Force (IRTF), Request for Comments RFC 7426, January 2015.

[2] D. Kreutz, F. M. V. Ramos, P. J. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[3] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 3, pp. 1617–1634, Third 2014.

[4] R. W. Downing, J. S. Nowak, and L. S. Tuomenoksa, "No. 1 ess maintenance plan," *Bell System Technical Journal*, vol. 43, no. 5, pp. 1961–2019, 1964.

[5] P. Baran, "On distributed communications networks," *IEEE Transactions on Communications*, vol. 12, no. 1, pp. 1 – 9, march 1964.

[6] ONF, "Software-defined networking: The new norm for networks," Open Networking Foundation, ONF White Paper, April 13 2012.

[7] C. WILSON. Verizon: Reliability a key SDN concern.

[8] P. E. Heegaard, B. E. Helvik, and V. B. Mendiratta, "Achieving dependability in software-defined networking - a perspective," in *7th International Workshop on Reliable Networks Design and Modeling (RNDM'15)*, Münich, Germany, October 5-7 2015, p. 7.

[9] S. Fernandes and M. Santos, "Sdn dependability: Assessment, techniques, and tools," in *SDN Research Group, IETF 93*, Prague, Czech Republic, July 19-24 2015.

[10] F. Longo, S. Distefano, D. Bruneo, and M. Scarpa, "Dependability modeling of software defined networking," *Computer Networks*, vol. 83, pp. 280 – 296, 2015.

[11] J. Wu, Y. Huang, J. Kong, Q. Tang, and X. Huang, "A study on the dependability of software defined networks," in *2015 International Conference on Materials Engineering and Information Technology Applications (MEITA 2015)*, September 2015.

[12] T. A. Nguyen, T. Eom, S. An, J. S. Park, J. B. Hong, and D. S. Kim, "Availability modeling and analysis for software defined networks," in *Dependable Computing (PRDC), 2015 IEEE 21st Pacific Rim International Symposium on*, Nov 2015, pp. 159–168.

[13] P. E. Heegaard, B. E. Helvik, G. Nencioni, and J. Wäfler, "Managed dependability in interacting systems," in *Principles of Performance and Reliability Modeling and Evaluation*, L. Fiondella and A. Puliafito, Eds. Springer, 2016.

[14] G. Ciardo and K. S. Trivedi, "A decomposition approach for stochastic reward net models," *Perf. Eval*, vol. 18, pp. 37–59, 1993.

[15] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing: Probability Models*. To BEGIN WITH, 1975.

[16] A. J. Gonzalez and B. E. Helvik, "Characterization of router and link failure processes in UNINETT's IP backbone network," *International Journal of Space-Based and Situated Computing*, 2012.

[17] P. Kuusela and I. Norros, "On/off process modeling of ip network failures," in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, June 2010, pp. 585–594.

[18] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger, "General availability model for multilayer transport networks," in *Proceedings.5th International Workshop on Design of Reliable Communication Networks, 2005. (DRCN 2005)*. IEEE, October 16-19 2005, pp. 85 – 92.

[19] G. Nencioni, B. E. Helvik, A. J. Gonzalez, P. E. Heegaard, and A. Kamisiński, "Impact of SDN Controllers Deployment on Network Availability," Department of Telematics, NTNU, Tech. Rep., March 2016. [Online]. Available: http://people.item.ntnu.no/~gianfran/SDNctrlDep.pdf