Availability modeling and evaluation of switches and data centers

Shakthivelu Janardhanan[1], Carmen Mas-Machuca[2]
[1]Chair of Communication Networks, Technical University of Munich, Munich, Germany
[2]Chair of Communication Networks, University of the Bundeswehr Munich, Neubiberg, Germany
shakthivelu.janardhanan@tum.de, cmas@unibw.de

Abstract—Data Center Network (DCN) availability is defined as the ability of the DCN to handle all demands at a particular time. DCN availability depends on the type and the availability of its switches. Similarly, switch availability depends on its hardware (HW) and software (SW) subcomponents. This work presents a detailed analysis of DCN availability by modeling and evaluating the availability of its components like traditional DCN switches, P4 hardware target switches, and P4 software target switches (and their subcomponents). This work is intended as a guideline for the DCN operator (i) to identify the critical subcomponents of the network, (ii) to understand the effect of different DCN topologies on the availability of the DCN, and (iii) to improve the availability of the DCN by replacing some components (and/or subcomponents).

Keywords–reliability; availability; data center

1. Introduction

A data center is a dedicated facility to store and process data. Modern-day data centers are complex structures built to be operable for an extended period. Cisco's Annual Internet Report [1] shows that the number of internet users is expected to increase by 35% from 2018 to 2023. Another Cisco survey [2] shows an 80% increase in the number of hyper-scale data centers from 2015 to 2020, corresponding to a compounded annual growth rate of 13%. This massive increase in internet traffic can be attributed to the growing need for faster communication with higher capacity. Uptime Institute's Global DCN survey [3] also shows strong growth in global digital infrastructure. Operators are increasingly investing in sustainable, resilient, and reliable networks. Thus, DCNs have evolved into the pillars of communication networks.

Uptime Institute's 2022 Outage Analysis [4] has stated that 80% of operators have faced some outage in the past three years. 70% of outages have caused a loss of over $100,000, while 25% of outages have caused a loss of over $1 million in 2022 (as of September 2022). This trend has shown a steady increase in outages and associated costs. Another alarming factor is the duration of the outages. Around 30% of outages in 2021 have disrupted service for over 24 hours. Moreover, networking-related issues of all levels of severity have caused the longest downtime incidents in the past three years. The outages have been closely linked to the increasing complexities of using software-defined networks (SDN), cloud services, and distributed architectures. The outage can be caused by an unreliable HW or SW component in the network.

For example, consider the Samsung Galaxy Note 7 issue [5]. These phones had a design flaw in their batteries, leading to overheating and explosion. Consequently, Samsung was forced to replace 2.5 million sold phones. Though this example is not DCN-related, it shows the importance of reliable HW subcomponents. SW subcomponent issues have also caused significant damage in DCNs. For example, in December 2021, Canada shut down 4000 government websites because of a security issue in a third-party library called Apache Log4j [6]. Another example is the data breach in Equifax servers caused due to the security vulnerability in another third-party library called Apache Struts [7].

In these examples, some subcomponent was unavailable. Here, the term 'unavailable' does not mean inaccessible. It refers to the inability of a subcomponent to perform its functions. These examples advocate the necessity to identify, model, and evaluate the availabilities of critical subcomponents.

With the upcoming 6G services, the focus is on human-centric applications [8]. The most impactful applications in 6G communications include massive machine-type communication, autonomous networks, e-Health, connected mobility, semantic communication, virtual and augmented reality (VR/AR), pervasive computing, ubiquitous computing, and digital twins. All these applications have different agendas and implementation patterns. However, they all rely upon three major factors: (i) storage of vast volumes of data, (ii) latency requirements, and (iii) robustness of their networks.

All these factors invariably point to the need for a highly available DCN. Network availability is the probability that a network offers uninterrupted service at a particular time. In this work, the focus is on building a highly available DCN by modeling the components and subcomponents of the network. This is shown as a top-down approach in Fig. 1. To evaluate the network availability (Top level in Fig. 1), it is essential to evaluate the components' availabilities (Intermediate level in Fig. 1) used in the network. It is possible to go one step lower and argue that the component's availability depends upon the subcomponents' availability (Low level in Fig. 1) used by each component. Therefore, this work aims to calculate the DCN availability by evaluating the steady-state availabilities of the subcomponents used in the DCN.

To provide the DCN operators with comprehensive information, multiple types of switches and different DCN topologies have been considered. The combinations of input choices are discussed in Secs. 3 and 4. This paper follows the workflow shown in Fig. 2. First, the subcomponents of the components in a DCN are identified. They are modeled as Stochastic Activity Networks (SAN). The parameters needed to model

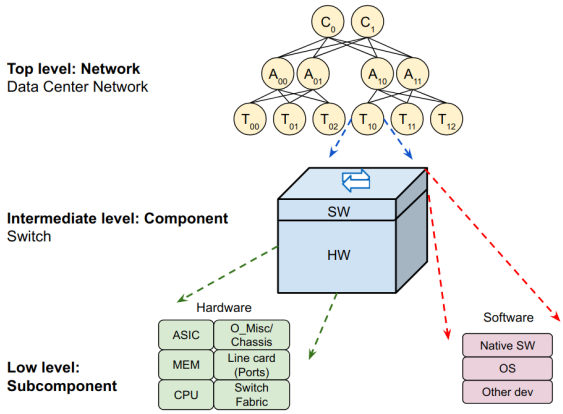


Figure 1: Top-down breakdown of a DCN

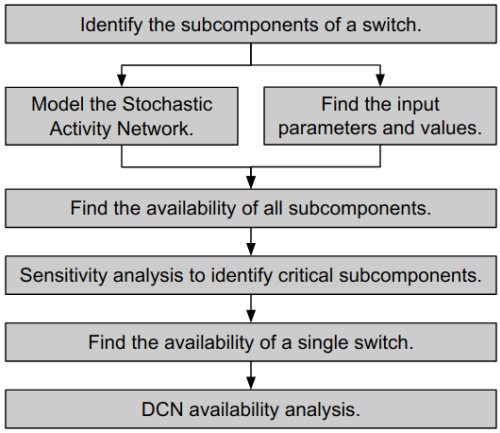


Figure 2: Workflow in this paper

these subcomponents' availability are determined, and their values are obtained from an extensive literature survey. The subcomponents are then simulated to get their steady-state availabilities. These values are later used to find the switch availability. Consequently, the switches' availabilities are used to find the DCN availability.

This paper aims to provide guidelines to DCN operators on identifying critical subcomponents of the network and how to improve DCN availability. This paper is arranged as follows. Sec. 2 discusses the related work. Secs. 3 and 4 explain the switch and DCN modeling, respectively. Secs. 5 and 6 discuss the results of the switch and DCN availability analyses, respectively. Sec. 7 presents the guidelines to the DCN operators on building a reliable DCN.

2. Related Work

Data required to study DCN availability can be gathered from:

(i) DCN statistics on critical issues and their consequences,
(ii) DCN statistics on traffic characterization,
(iii) surveys on relevant modern DCN components, and
(iv) data sheets of said components (and subcomponents) to model failure characteristics correctly.

Several studies like [9]–[11] summarize the failure data from real-time DCNs. In these works, the term 'issue' is used to denote any incident that occurs in the DCN which is unaccounted for and affects the DCN's performance in some aspect. [10] states that an alarming 29% of intra-DCN issues at Facebook from 2011 to 2018 have undetermined causes, and hence it is nearly impossible to prevent these issues from occurring in the future. SW issues occur at a significantly higher rate than HW failures [10], [11]. With the increase in softwarization features of switches, this number is expected to increase. Though SW issues occur more often, the downtime caused by them is smaller than that due to HW issues [9]. Top of the Rack (ToR) switch failures cause most of the downtime [9], [10]. This comes as no surprise because, in most commercial data centers, there exists no redundancy in the connection between the servers and the ToR switches. Fabric-based topologies like the Facebook Fabric (Fb-Fab) are more robust than cluster-based topologies like the Leaf-Spine topology [10], [12]. Bigger networks with more nodes and/or edges take longer to detect and repair failures. [10]. Despite having redundant paths in the upper layers of switches, switch failures still cause a noticeable drop in performance [9].

The authors in [13] predict the failing times of the switches by using a Kaplan-Meier survival estimator. This is a suitable method if the network is small, highly predictable, and consistent in behavior. When the network size increases, the uncertainty in accuracy may increase. Other works like [14]–[17] analyze the robustness of a DCN. However, these works do not consider components' behavioral characteristics and subcomponents' failures. Therefore, these works do not explicitly capture the minuscule failing patterns of the subcomponents. The authors in [18] analyze the DCN behavior using analytical methods. Though this model can be further developed into a Discrete Event Simulator (DES) to simulate failures of different subcomponents, it is unrealistic due to the program's complexity and runtime scaling issues. Most subcomponents have failing times in the order of tens of years. Therefore, a DES that simulates traffic and failures together will never be able to capture the required switch characteristics.

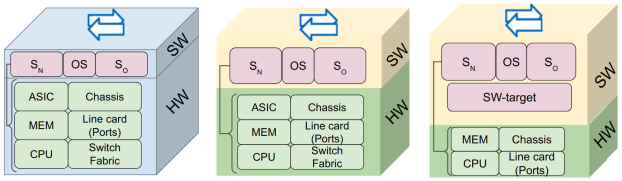
The authors in [19] perform an analysis similar to our work. They model the subcomponents of a switch and use a hierarchical model to evaluate the DCN availability. However, there are some limitations in their setup. The subcomponent models consider that failure reparation is always successful. However, in reality, this may not be the case. After a failure, the reparation may be unsuccessful, and the HW may have to be entirely replaced. Another critical factor is that they evaluate their model for a very small DCN with 16 servers only. Moreover, they consider only two topologies- Fat tree and Leaf Spine. Additionally, the DCN traffic matrix is also not considered. Instead, they consider arbitrary flows connecting different combinations of servers. This is understandable because they did not aim to provide a comparison between topologies or types of switches. Table I summarizes the additional features of our work compared to [19]. Another work [20] considers a similar approach to model an SDN controller's availability. Though we can not use the values from [20], the approach in [20] is a good reference point.

TABLE I: Comparing this work with State of the Art

Comparing criteria	Work in [19]	This work
Topology	3 Tier Leaf-Spine, Fat tree	3 Tier Leaf-Spine, Fat tree, AB-Fat tree, Facebook 4-Post, Facebook Fabric
Size	16 servers	32000 servers
Server availability	Considered	Not considered
Switch availability	Considered	Considered
Traffic matrix	Not considered	Considered
Traffic routing	Not considered	Considered
Probability of failed reparation in subcomponent availability modeling	Not considered	Considered
Different types of switches	Not considered	Considered
Different types of DCN	Not considered	Considered

TABLE II: Input options

Type of switch	Commercially available DCN switch (Traditional), P4 Hardware target switch (P4-HW), P4 Software target switch (P4-SW)
Type of DCN	Traditional DCN, P4-HW DCN, P4-SW DCN, Hybrid
Topology	3 Tier Leaf-Spine, Fat tree, AB-Fat tree, Facebook 4-Post, Facebook Fabric
Size	32000 servers



(a) Traditional (b) P4-HW (c) P4-SW

Figure 3: Switch models and their subcomponents

Depending on the DCN requirements, such as size, traffic, location, latency, and cost, the DCN components vary significantly. Thus, it is not fair to say that all switches have the same behavior. However, the switches have comparable behavior. Note that in this work, we consider the links to have the same availability always as in [19]. The DCN model used in this work is similar to the model used in [12]. In [12], the authors have captured all the critical DCN traffic characteristics.

The subcomponents of the devices have been modeled individually in our work, similar to [19] with appropriate changes in the modeling parameters. The subcomponents considered are only related to the networking capacity of the switch. We do not consider the power supply units of the switches.

3. Switch Modeling

Three types of switches and four types of DCNs are considered in this work, as seen in Table II. In the traditional DCN, all switches are the same commodity switches found in today's DCNs, like the Cisco 9600 series switches [21]. They are referred to as traditional switches (TS) in this work. Next, there are more advanced Programming Protocol-independent Packet Processor switches called P4 switches. P4 switches can program the data plane functionality of the switch. They are considered the future of the SDN paradigm. Several works like [22] discuss the advantages and potential of P4 switches in DCNs. However, to the best of our knowledge, no work has evaluated the reliability of this type of switch.

P4 programs are compiled and executed on P4 targets. These targets can be HW or SW targets. P4 Hardware targets (P4-HW) include FPGAs like NetFPGA, ASICs like the Tofino, and NPUs like Netronome SmartNIC [23]. P4 Software targets (P4-SW) include different SW paradigms like T4P4S [23]. The differences in the architectures and performances of these switches have been evaluated in [24]. In this work, we do not go into the performance details but their availability. For this purpose, we model the switch's and its subcomponent's failures. This work considers an ASIC-based P4-HW switch and a T4P4S-based P4-SW switch. However, this analysis can be extended to any other type of switch.

3.1. Subcomponents of a switch

A switch consists of several subcomponents. This work focuses only on the subcomponents related to networking. The various subcomponents considered for the different types of switches are shown in Fig. 3. The demarcation in HW-SW is a relative indicator of the degree of softwarization in these switches. ASIC, Memory (MEM), Chassis, CPU, line cards (LC), and switch fabric (SF) are the common HW subcomponents (HW_{sub}) found in a switch [19]. Operating system (OS), native SW from the switch manufacturer (S_N), and SW from other developers (S_O) are the common SW subcomponents (SW_{sub}) found in the switch [25]. Note that the HW_{sub} also have some SW associated with them. For example, the CPU is a HW_{sub}. However, the proper functioning of the CPU requires some SW, like the BIOS settings. This SW associated directly with the HW_{sub} is the native SW (S_N). S_O refers to third-party libraries that may be used in the devices. For example, in the Apache Log4j issue discussed in Sec. 1, Log4j is the third-party library used. Therefore, HW_{sub} has a small part of SW involved in S_N. This is represented by the connection between S_N and the HW_{sub} in Fig. 3.

As seen in Figs. 3a and 3b, the TS and the P4-HW switch have all the aforementioned subcomponents. The degree of softwarization in these switches varies clearly. This variation does not change the models of the subcomponents but impacts some of the values for parameters like, the type and impact of failures, mean time between failures (MTBF), mean time to repair (MTTR), and mean time to replace (MTTReplace). However, for the P4-SW switch, there is no dedicated HW_{sub} like the ASIC because the SW target handles the networking functionality. So the ASIC is removed, but the SW target is added in Fig. 3c. The SF subcomponent is also removed in the P4-SW switch because the P4-SW switch is essentially the SW target program installed in a mini-PC or a server like the Nokia NDCS16RM AirFrame Compute Node used in [24].

Failures in any of these subcomponents will affect the switch. Some failures cause complete switch failure (e.g., HW failures), while others cause some flows to fail (e.g., SW failures).

For example, an issue in the MEM subcomponent may cause some flows to fail because some registers were inaccessible for some time. [25] discusses a similar approach to partial switch failures due to SW issues.

3.2. Subcomponent Modeling

The subcomponents are modeled as Stochastic Activity Networks (SAN) using the Möbius modeling tool [26]. For example, the ASIC subcomponent's SAN is shown in Fig. 4. The model starts with one token in the 'ASIC_OK' place. A timed activity, 'ASIC_fail', triggers an ASIC failure. For the HW_{sub}, the failure can be either HW-related or SW-related. However, the probability of having a HW failure in the HW_{sub} is much higher (P(HW failure|failure in HW_{sub}) = 0.95) than having a SW failure in the HW_{sub}. Rarely the failure in the HW_{sub} is related to S_N. This assumption can be justified because the S_N is a tiny portion compared to the actual HW_{sub} itself. Moreover, the proprietary piece of SW that arrives with the HW is well-tested and more bug-free than S_O.

If the failure is a HW failure in the ASIC, it is repaired first. Then, the ASIC unit is restarted to check if the reparation was successful as an instantaneous activity. If the reparation worked, the token is returned to the 'ASIC_OK' place. If not, the ASIC is replaced, and the token is returned to the 'ASIC_OK' place. When there is a SW failure, the failure is repaired, and then the ASIC is restarted. If the reparation worked, the token is returned to the 'ASIC_OK' place. If not, the SW is repaired again. The times taken to fail, repair, and replace are exponentially distributed. The values are given in Table III along with their references and justifications.

The times to fail for subcomponents in data sheets range from tens to hundreds of years. For example, the MTBF for the Cisco Catalyst 9600X Supervisor Engine 1 (C9600-SUP-1), which houses the Cisco Unified Access Data Plane Application-Specific Integrated Circuit (UADP-ASIC), is 271420 hours (∼31 years) [27]. On the other hand, the MTBF for the C9606R chassis used in the 9600 series switches is 4113900 hours (∼470 years) [21]. These numbers may be unrealistic because of the testing conditions of these devices. There are no guarantees that the devices perform this well in real conditions. Previous statistics from DCNs [9]–[11], [13], [28]–[31] also indicate that (i) HW failures occur more often than expected from data sheets, and (ii) HW failures cause most downtime. This may be due to the aging of the HW or SW, or improper physical or networking conditions for the switch. Therefore, each subcomponent has a second SAN, an aging-related SAN, to account for the aging issues.

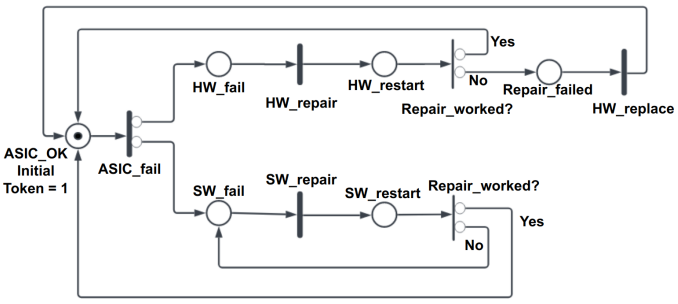


Figure 4: SAN model of the ASIC

The aging-related SAN has the same structure as the SAN discussed in Fig. 4. However, the rate of failures is different. The aforementioned issues are assumed to occur 16 times faster than the rate from data sheets. For example, the ASIC's MTBF is nearly 30 years. According to [32], the ASIC under good maintenance is expected to last around three years. If the ASIC is overclocked or subjected to poor conditions, its lifespan may decrease to a few months. A pessimistic approach would consider an aging-related issue in the ASIC every few months. A realistic approach that accounts for some faults in the management and operational conditions of the ASIC would consider aging-related issues every two years or so. In [33], the authors also consider aging failures of virtual machines (VMs). Here, the ratio of the rate of non-aging failures to aging failures is around 16. Comparing other subcomponents, aging failures occurring 16 times faster seems appropriate. However, this is a variable that can be changed as per the operator's experience. After all, this work only aims to provide the operator with a method to analyze the DCN availability. Apart from the failure rate, the probability of a HW_{sub} producing a SW error is also higher in the aging case. This is because aging affects SW (S_N) more than HW. Therefore, P(HW failure|aging failure in HW_{sub}) is considered 0.9. The SAN is smaller for a SW_{sub} like the OS because the HW failure section in Fig. 4 is absent.

3.3. Subcomponent modeling- Parameters and values

The parameters to model the subcomponents and their values are mentioned in Table III. Here, the availability of subcomponent sub is denoted as A_{sub}. Three sets of parameters and values exist for traditional, P4-HW, and P4-SW switches. The probability of HW being affected due to a failure in the HW_{sub} is lower in the case of the P4 switches. This is due to the increase in the softwarization of modern switches. As a result, more failures are associated with the SW rather than the HW. Therefore, P(HW failure| failure in HW_{sub}) and P(HW failure|aging failure in HW_{sub}) are reduced to 80% of their original values considered for TSs.

When there is a HW failure, the availability of that subcomponent is 0. When there is a SW failure, it might be a partial failure, as discussed in Sec. 3-A. Since no previous works exclusively keep track of the number or type of flows affected due to a switch's SW failure, the availability of the failed SW_{sub} is assumed to be around 0.5. The failure can be severe (availability less than 0.5) or ineffective (availability more than 0.5). This variation is modeled as a Gaussian random variable with a mean of 0.5 and a variance of 0.3. Due to the increased softwarization in P4 switches, two attributes are expected:

(i) The number of SW bugs is significantly high. Therefore, the S_O subcomponent's MTBF is reduced to 75% of the original value assumed for TSs.

(ii) The code and functionalities are modular, well-written, and easy to debug. Therefore, a decrease in the MTTR by 50% is considered to ensure fairness.

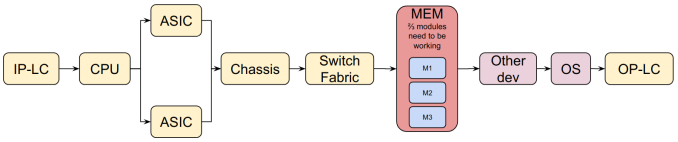
TABLE III: Table with all parameters and their baseline values

Parameter	Base line values			Justification
	Traditional	P4-HW	P4-SW	
A_{sub}, when no failures	1	1	1	Subcomponent is available.
A_{sub}, when HW failures	0	0	0	Subcomponent is unavailable.
A_{sub}, when SW failures (except SW-target and S_O failures)	$\mu = 0.5$, $\sigma^2 = 0.3$	$\mu = 0.5$, $\sigma^2 = 0.3$	$\mu = 0.5$, $\sigma^2 = 0.3$	Subcomponent is partially available.
MTTR for HW failures [hours]	2	2	2	From [19], [20]
MTTR for SW failures [hours]	0.33	0.16	0.16	From [19], [20]. For P4 switches, the MTTR is lesser.
MTTReplace for HW failures [hours]	2	2	2	From [19], [20]. For P4 switches, the MTTR is lesser.
P(HW failure| failure in HW_{sub})	0.95	0.76	0.76	Assumed based on [33] and justified in Sec. 3-B
P(HW failure| aging failure in HW_{sub})	0.9	0.72	0.72	Assumed based on [33] and justified in Sec. 3-B
P(Successful HW repair)	0.9	0.9	0.9	Assumption- Left to the operator to decide based on internal statistics.
P(Successful SW repair)	0.95	0.95	0.95	Assumption- Left to the operator to decide based on internal statistics.
ASIC non-aging MTBF [hours]	271420	271420	271420	From [27] for the Cisco Catalyst 9600X Supervisor Engine 1
ASIC aging MTBF [hours]	16963	16963	16963	0.0625× ASIC non-aging MTBF
CPU non-aging MTBF [hours]	386228	386228	386228	Datasheets of CPUs used in modern switches are unavailable. Thus, from [34], [35], the MTBF of the Intel Server Board S1200V3RP is used. Though this is not for a switch, a switch CPU can be comparable to this CPU in terms of MTBF.
CPU aging MTBF [hours]	24139	24139	24139	0.0625× CPU non-aging MTBF
LC non-aging MTBF [hours]	543000	543000	543000	From [36], the MTBF of the Cisco 9600 series line card C9600-LC-48TX is used.
LC aging MTBF [hours]	33937	33937	33937	0.0625× LC non-aging MTBF
MEM non-aging MTBF [hours]	40000	40000	40000	From [37], MTBF is calculated from 25000 FITs experienced in DRAMs as explained in Sec. 3-C.
MEM aging MTBF [hours]	2500	2500	2500	0.0625× MEM non-aging MTBF
Chassis non-aging MTBF [hours]	4113900	4113900	4113900	From [21], the MTBF of Cisco 9600 chassis C9606R is taken.
Chassis aging MTBF [hours]	257118	257118	257118	0.0625× Chassis non-aging MTBF
SF non-aging MTBF [hours]	422115	422115	422115	From [19], [38], the MTBF of the GSR12-SFC switch fabric in the Cisco 12000 switch is used.
SF aging MTBF [hours]	26382	26382	26382	0.0625× SF non-aging MTBF
OS non-aging MTBF [hours]	18000	18000	18000	From [19], [39], the MTBF of the OS is used.
OS aging MTBF [hours]	1125	1125	1125	0.0625× OS non-aging MTBF
S_O non-aging MTBF [hours]	8760	6570	6570	Assumed MTBF of 1 year because there is no previous data available. Left to the operator to decide based on the SW-target in use.
S_O aging MTBF [hours]	547	410	410	0.0625× S_O non-aging MTBF
SW-target non-aging MTBF [hours]	N.A.	N.A.	4380	Assumed to be 6 months because there is no previous data available. This is a rather optimistic number. Left to the operator to decide based on the SW-target in use.
SW-target aging MTBF [hours]	N.A.	N.A.	273	0.0625× SW-target non-aging MTBF
A_{sub}, when S_O failed	0.7	0.5	0.3	Assumption. With the increase in softwarization, the dependency on SW increases. So, a bug in the code may cause more failed flows.
A_{sub}, when SW-target failed	N.A.	N.A.	0	SW target is critical to the P4-SW switch. If it fails, the switch is unavailable.

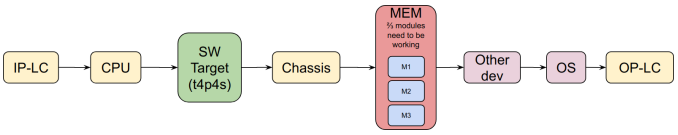
The MEM block can be a combination of high-speed DRAM and a relatively slower ROM. For example, a Seagate SSD memory unit like Nytro 1000 SATA SSD Series is advertised as an enterprise-grade solution for data center applications. It has an MTBF of 2000000 hours (∼228 years). This is unrealistic for a switch memory because it will be extensively used in the data center. An extensive statistical study on DRAM errors by authors in [37] reveals that the DRAM error rates are significantly greater than previously reported. They report 25000 to 70000 FITS. An alarming 8% of all dual in-line memory modules (DIMMs) are affected yearly. 25000 FITs correspond to an MTBF of 40000 hours. This value is taken in our work for the MTBF of memory modules. Note that we have taken an optimistic value. A pessimistic approach would be to consider 70000 FITs (MTBF of 14286 hours).

3.4. Switch model

The availability of a switch is modeled as a Reliability Block Diagram (RBD) as shown in Fig. 5. The RBD for traditional and P4-HW switches are similar because they have the same subcomponents, as seen in Fig. 5a. However, the values for



(a) RBD for traditional and P4-HW switches



(b) RBD for P4-SW switches

Figure 5: RBD for switch availability

the blocks are different. For the P4-SW switch, the necessary changes have been made in Fig. 5b. IP-LC and OP-LC denote the input and output line cards. The ASIC has 1+1 redundancy in this model per the Cisco 9600 series supervisor engine [27] architecture. There have been several memory architectures over the years. In this work, a generalized architecture with three memory modules is used. The MEM block's availability (A_{MEM}) is calculated as shown in Equation 1.

$$A_{MEM} = \begin{cases} 1, & \text{if } 3/3 \text{ memory blocks are functional} \\ 0.5, & \text{if } 2/3 \text{ memory blocks are functional} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The subcomponent blocks are in series in the RBD because any subcomponent's unavailability implies the switch's unavailability. Numerically, the switch availability can be represented by Eq. 2 for TS and P4-HW switches and by Eq. 3 for P4-SW switches.

$$\begin{aligned} A_{P4-HW} &= A_{Traditional} \\ &= A_{IP-LC} \times A_{CPU} \times \left(1-(1-A_{ASIC})^2\right) \times \\ &\quad A_{Chassis} \times A_{SF} \times A_{MEM} \times \\ &\quad A_{S_O} \times A_{OS} \times A_{OP-LC} \end{aligned} \tag{2}$$

$$\begin{aligned} A_{P4-SW} &= A_{IP-LC} \times A_{CPU} \times A_{SW-target} \times \\ &\quad A_{Chassis} \times A_{MEM} \times A_{S_O} \times A_{OS} \times A_{OP-LC} \end{aligned} \tag{3}$$

4. DCN Modeling

The previous section concerned the low and intermediate levels shown in Fig. 1. In this section, the top level in Fig. 1, that is, the DCN modeling, is discussed. The ToR switch architecture [40] is used in this study. The DCN simulator uses Python.

Five topologies are examined in this work, as seen in Table II. Each topology has a core switch (CS), aggregate switch (AS), and a ToR connected to servers. In the 3-Tier Leaf-Spine topology (3TLS), the leaf and spine switches in the lower and upper layers form a pod. Each leaf is connected to each spine inside the pod. Extending the same arrangement to a super-spine layer above the spine layer gives the 3TLS, as seen in Fig. 6a. To improve scalability, a Clos-network-based topology called Fat Tree (FT) was proposed [41]. Here, each

TABLE IV: Infrastructure for all topologies ($32K$ Servers)

Topology	(n_S, n_T, n_A, N_P, N_C)	(L_{ST}, L_{TA}, L_{AC}, L_{CI}, R_A, R_C)
3TLS	(96, 30 8, 11, 6)	(10, 40, 100, 400, -, -)
FT, ABFT	(64, 16, 5, 32, 40)	(10, 40, 40, 40, -, -)
Fb-4P	(44, 48, 4, 15, 4)	(1, 10, 40, 400, 80, 160)
Fb-Fab	(48, 48, 4, 14, 96)	(10, 40, 40, 40, -, -)

TABLE V: Notation used in Table IV

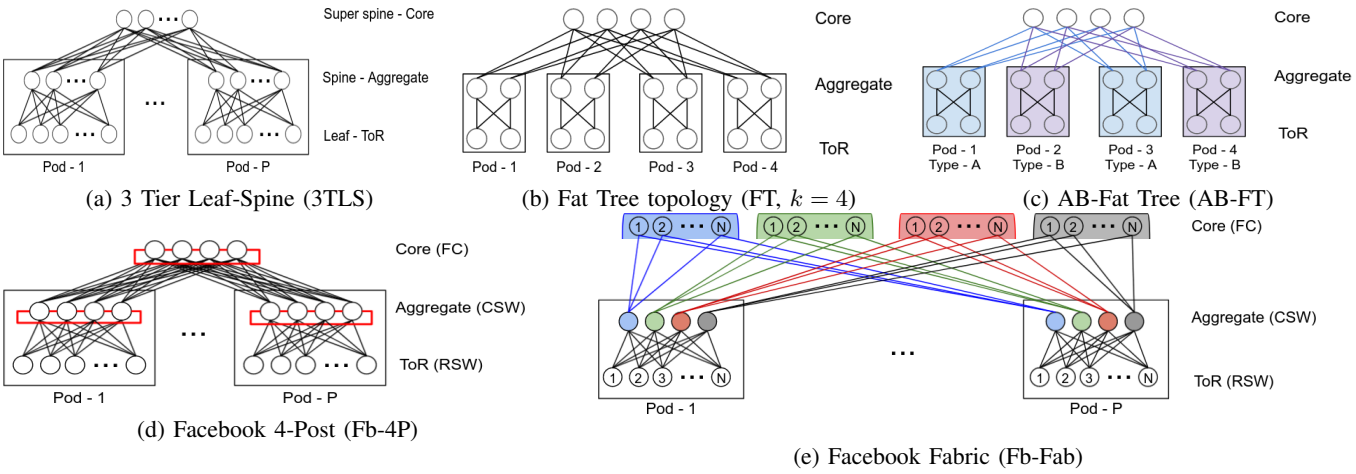
n_S	No.of servers per ToR	n_T	No.of ToRs per pod
n_A	No.of aggregate switches per pod	N_P	No.of pods
N_C	No.of core switches	L_{ST}	Server-ToR link capacity (Gbps)
L_{TA}	ToR-Aggregate link capacity (Gbps)	L_{AC}	Aggregate-Core link capacity (Gbps)
L_{CI}	Core-Inter-DC link capacity (Gbps)	R_A	Aggregate ring capacity (Gbps)
R_C	Core ring capacity (Gbps)		

CS is connected to one AS in each pod. The AB-Fat Tree topology (AB-FT) was proposed at the cost of complicated interconnections to improve the FT's robustness. It is a skewed FT, as seen in Fig. 6c. In Ab-FT, two types of pods are placed alternatively. In type-A pods, every AS is linked to consecutive CSs. In type-B pods, every AS is linked to the core layer in steps of fixed length. Modifying the 3TLS, with a ring in the aggregate and core layers, the Facebook 4-Post topology (Fb-4P) [42] as seen in Fig. 6d was introduced. The ring structure improved reliability and efficient load balancing. The Facebook Fabric topology (Fb-Fab) is the most recent noticeable tree-based topology, as seen in Fig. 6e. There are four core layer planes. Each of the 4 ASs in a pod is linked to all the CSs in a plane. Every CS is linked to one AS per pod.

DCNs vary in size from about 1000 servers to a few thousand servers. In our work, a medium-sized DCN with 32K servers is simulated. The number of switches in different layers changes with the topology. Table IV shows the infrastructure used in this work. The notation in Table IV is explained in Table V.

The traffic matrix is important in this work as opposed to [19] because the DCN availability is calculated based on the generated traffic in the simulation. The key characteristics of the traffic matrix are as follows.

- The DCN is simulated for 1 second.
- The number of flows per server is around 800 for the year 2022 as per [12].
- The oversubscription ratio is the ratio of the bandwidth of southbound ports to northbound ports [43]. It can vary from $2.5:1$ to $240:1$ [44]. However, in our work, similar to [12], [25], the ToR and ASs are oversubscribed at $3:1$ and $2:1$ respectively except for Fb-4P topology. It uses the architecture explained in [42].
- Around 70% of flows are intra-rack [45].
- Around 90% of the flows are intra-DCN [12].
- The flow sizes are modeled from the previous studies in [46]–[48]. Around 80% of flows are smaller than $10KB$, about 95% of flows are smaller than $1MB$, and nearly



(a) 3 Tier Leaf-Spine (3TLS)

(b) Fat Tree topology (FT, $k = 4$)

(c) AB-Fat Tree (AB-FT)

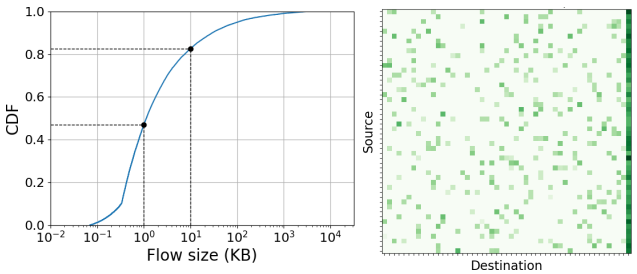
(d) Facebook 4-Post (Fb-4P)

(e) Facebook Fabric (Fb-Fab)

Figure 6: DCN Topologies

99% of flows are smaller than $100MB$.

- The traffic volume is generally dominated by the larger flows that are seen rarely. With this information, the flow size from the simulator can be modeled as a modified Pareto distribution as seen in Fig. 7a.
- As an example, the traffic between the source (Y-axis) and destination (X-axis) ToRs for a small FT topology with 1000 servers plotted in Fig. 7b echoes previous findings [49]. Note that the last column is the inter-DCN traffic.
- The flows are routed using Equal Cost Multi-Path (ECMP) method. The load on the links after routing is similar to that observed in [45], [47].



(a) Flow Size-CDF

(b) ToR-TM for flows

Figure 7: Images from the simulator

Once the traffic is generated and routed, the availability for each flow A_f is calculated as the product of the switches' availabilities in the flow path as indicated in Eq.4.

$$A_f = \prod_{n \in path} A_n, \forall f \in Flows \tag{4}$$

The average DCN availability A_{DCN} is calculated as the weighted numerical mean of all the flows' availability in the DCN. The weight of each flow f, denoted by F_f, is the volume of that flow f while the total volume of all flows is denoted by F_{all}. This is given by Eq. 5.

$$A_{DCN} = \frac{1}{F_{all}} \times \sum_{f \in Flows} (A_f \times F_f). \tag{5}$$

5. Switch Availability Results

Based on the Möbius modeling explained in Sec. 3, the results for the subcomponent availability are generated and discussed in Sec. 5-A. A sensitivity analysis of the various parameters is also done to identify the most impactful parameters in Sec. 5-B. These results are later combined as shown in Sec. 3-D to derive the availability of the different types of switches. This is discussed in Sec. 5-C.

5.1. Subcomponent availability

Each subcomponent is modeled to find its availability. The simulations on the Möbius tool took around 5 seconds to evaluate the steady state availability of a subcomponent for one set of input parameter values. Fig. 8 shows all the subcomponents' steady state availabilities considering the baseline values discussed in Table III. From Fig. 8, the HW_{sub} have higher availability than the SW_{sub}. Note that the ASIC availability is calculated for a single ASIC chip. In the RBD, the ASIC has a 1+1 redundancy that improves its availability significantly. Similarly, the MEM availability is for a single MEM module only. Out of all the HW_{sub}, MEM has the least availability owing to its higher failure rate. The low availability of the SW-target for the P4-SW switch negatively impacts the switch's availability. Notice that the availability of these subcomponents does not vary much. However, when they are all considered in series in the switch RBD, the impact of these subcomponents is noticeable. The comparison between the different types of switches is discussed in Sec. 5-C.

5.2. Subcomponent availability- Sensitivity analysis

It is essential to know which subcomponent has poor availability, as discussed in Sec. 5-A. It is equally important to know which parameter of the subcomponent affects the subcomponent availability the most. This will help the operator to make better decisions to improve the DCN availability. For example, if the most impactful parameter is the failure rate, then the operator needs to look for better subcomponents. On the other hand, if the most impactful parameter is the MTTR

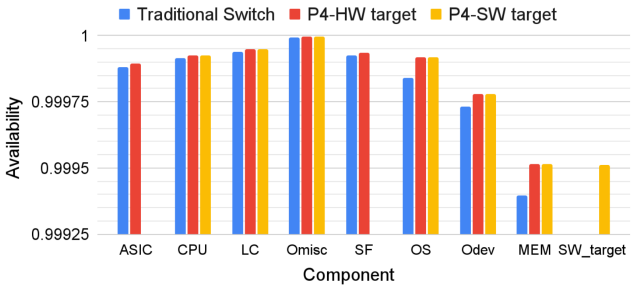


Figure 8: Subcomponent availability comparison

of the HW failure, then the operator may want to train the technicians to perform reparations faster.

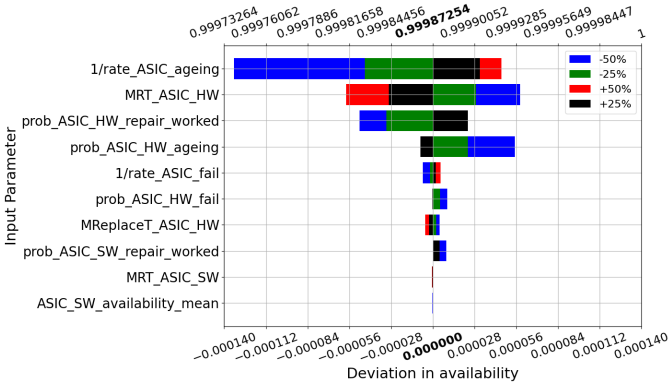
For example, the sensitivity analysis for the ASIC availability for the TS is shown in Fig. 9a. In this figure, along the upper X-axis, the availabilities are arranged. In the lower X-axis, the deviations in availability from the center value are arranged. Note that, for the ASIC in the TS, the center availability ($A = 0.99987254$) is the availability obtained when all the values in Table III are applied. Along the Y axis, all the parameters governing the ASIC availability are mentioned in decreasing order of their impact on the ASIC availability.

This tornado chart is to be understood as follows. When the parameter `'1/rate_ASIC_ageing'` (MTBF of ASIC subcomponent) is increased by 25%, then the ASIC availability increases by 0.00003. On the other hand, if this parameter reduces by 50%, then the availability drops by 0.00013. This is intuitive because fewer failures occur when the MTBF increases and availability improves. When MTBF decreases, failures are more frequent, causing the availability to decrease.

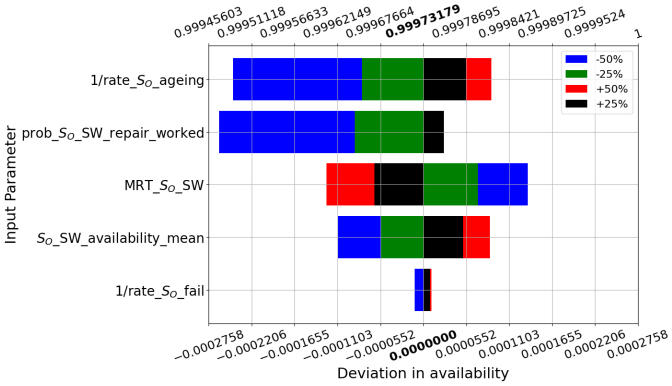
Consider the example in Fig. 9b, which shows the tornado chart for the S_O subcomponent. Here, changing the `'1/rate_`S_O`_ageing'` (MTBF of S_O subcomponent) shows a much more significant impact on the subcomponent availability. On reducing the MTBF of S_O by 50%, the availability decreases by 0.00023. This means that ASIC availability plays a minor role in the switch availability. For a network operator, this means that the operator can focus on other subcomponents that are more vulnerable than the ASIC.

By combining the results obtained from each subcomponent, an overall analysis can be made to understand the most impactful parameters that affect switch availability. Such plots for the TS and P4-SW switch are shown in Fig. 10. The plot for the P4-HW switch is very similar to the TS in Fig. 10a.

For the TS and P4-HW switches, the MEM subcomponent's MTBF and the MTTR of the HW failures are the most impactful parameters. This is because the MEM subcomponent's MTBF is significantly smaller than other HW subcomponents. The reparation time is also crucial because HW reparation is two hours, which is six times the SW reparation time. SW failure rates closely follow these parameters. This is because SW failures occur more often than HW failures. The probability of a SW reparation being successful is also substantial because every time the reparation is unsuccessful, another 20 minutes are required for the reparation. The probability that a



(a) ASIC subcomponent availability



(b) S_O subcomponent availability

Figure 9: Tornado plots for subcomponent availability

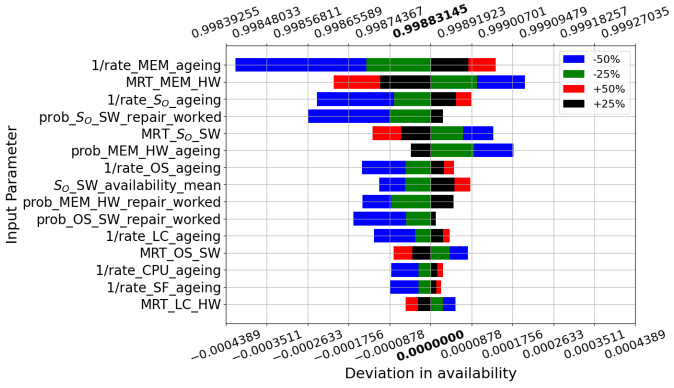
memory failure is HW-related is an interesting parameter. This is because when this value is high, more HW failures occur, causing more significant downtimes.

For the P4-SW switch, the most impactful parameters are all related to the SW target. This is because of two reasons.

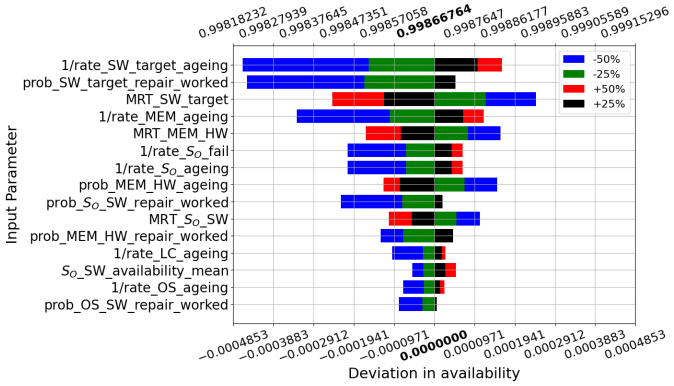
1) A SW target has many bugs (low MTBF) because it is a software program under constant development.
2) A SW target failure causes complete switch failure.

Therefore, the plots discussed above indicate the most impactful parameters and the values that should be assigned to ensure higher availability. It is indeed curious that the deviation in the availability of the most critical parameter in the switch influences the switch availability to a minimal extent. Therefore, multiple subcomponent parameters need to be improved to achieve a substantial improvement. The analysis in these plots is restricted by the values considered in Table III. A DCN operator can use this approach to use their values to understand how the different parameters affect the switch availability in their DCN.

Another comparison approach is to check the frequency of failures of subcomponents with the mean downtime caused by these failures. Fig. 11a shows this comparison for the TS. On the left, the frequency of failures of the subcomponents is shown. The SW subcomponents fail very frequently compared to the HW subcomponents. On the other hand, the downtime due to the HW subcomponent failures is much longer than the downtime due to the SW subcomponent failures. This trend



(a) Traditional switch (TS) availability



(b) P4-SW switch availability

Figure 10: Tornado plots for switch availability

is observed in all three switches consistently. This trend also agrees with previous studies [9]–[11], [13], [28]–[31].

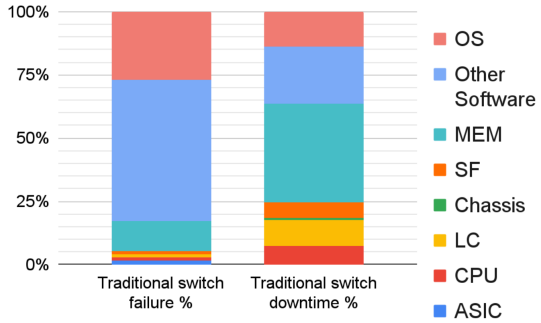
5.3. Switch availability comparison

Fig. 11 shows the comparison between different switches. Fig. 11b shows the availability of different switches based on the baseline values mentioned in Table III. The P4-HW switch availability is the highest. Note that the availability is higher despite this switch having more failures due to S_O and OS subcomponents. This can be attributed to the decrease in the MTTR of SW failures in P4-HW switches. The TS is second in place while the P4-SW switch has the worst availability. This is because of the increased SW failures in S_O and OS subcomponents. Moreover, the SW target switch has a comparatively more vulnerable subcomponent- the SW target. This means the P4-HW switch has the least downtime, while the P4-SW switch has the highest downtime.

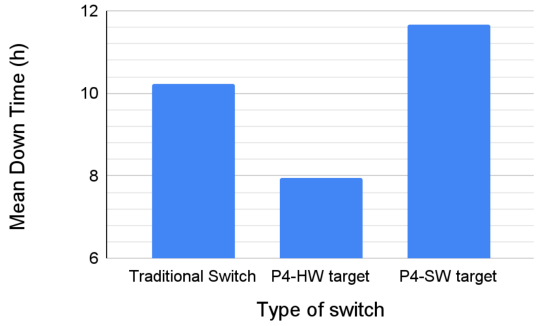
This comparison shows that softwarization has pros and cons. The switch availability varies depending on the subcomponents that are being softwarized, the degree of softwarization, and the reliability of the SW being used. The importance of the target that performs the networking functions in a P4 switch on the switch availability is seen in the comparison between the two types of P4 switches.

6. DCN Availability Results

Based on the results from Sec. 5, the DCN availability can also be calculated. The DCN is modeled as discussed in



(a) Failure-Downtime comparison



(b) Availability of different switches

Figure 11: Results of the switch availability analysis

Sec. 4. The different types of DCN are compared in Sec. 6-A while the potential methods to improve DCN availability are discussed in Sec. 6-B. The DCN availability was calculated using Python. The simulations took around 20 seconds to evaluate the DCN availability for one topology for one type of DCN.

6.1. DCN availability comparison

As mentioned in Table II, five topologies and four different types of DCNs are considered. In Fig. 12, the different constituent switches of the DCN are arranged along the X-axis, while the availability is seen on the Y-axis. Each color in the bar graph denotes a different topology. Therefore, Fig. 12 shows two different comparisons.

Topology comparison: In each type of DCN, the variation in the availability of the different topologies is minimal. This shows that all topologies have an inherent robustness comparable to each other. Therefore, no decision can be made based on the availability of the topologies. A similar conclusion has been derived in [25], where the authors conclude that all topologies have comparable robustness. The choice in topology can be made only based on the size and expected traffic in the DCN.

DCN-switch type comparison: Considering a particular topology, the different types of switches that can be used in the DCN cause a significant change in availability. This graph echoes Fig. 11b. That is, the DCN constructed only using the P4-HW target switches outperforms the TS and P4-SW switches-based DCNs. The fourth column in the graph refers to a hybrid DCN construction where 50% of the components

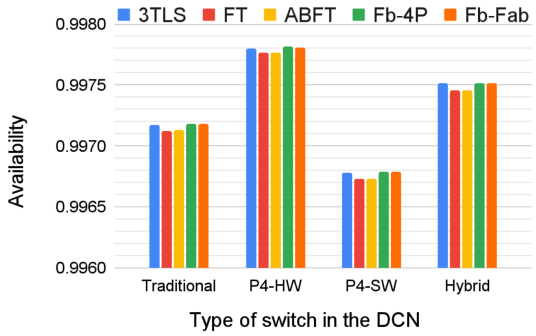


Figure 12: DCN topology comparison

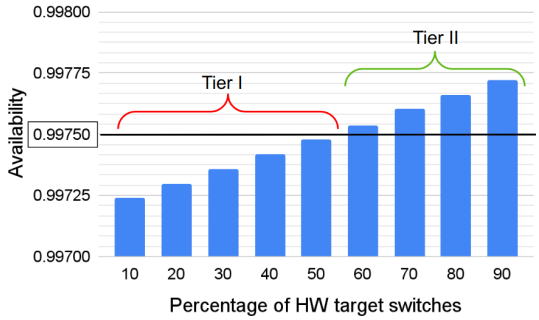


Figure 13: Upgrading traditional DCN with P4-HW switches

are TSs while the other 50% of the switches are P4-HW. This hybrid architecture shows a decisive advantage over the TS-based DCN. This is because 50% of this hybrid DCN consists of superior components than the TS-based DCN. This is a good brownfield scenario when the DCN operator already has a DCN with TSs but wishes to upgrade his DCN with newer components. Sec. 6-B further shows how the availability of an existing DCN can be improved and what significance is associated with this upgrade.

6.2. Towards improving DCN availability

It is clear from Sec. 6-A that changing the topology does not improve the DCN availability. Therefore, other strategies must be tested to improve DCN availability:

(i) Upgrade the switches to switches with higher availability.
(ii) Upgrade the subcomponents in the switches to subcomponents with higher availability.

For example, for a small DCN, the operator may decide that TSs are sufficient. However, after a couple of years, the increase in traffic and the necessity to improve the DCN availability may force the operator to upgrade the DCN. On the other hand, the operator might want to upgrade the DCN availability to be classified into a higher tier in the DCN classification. According to the Uptime Institute's Tier classification [50], the recommended availability for a DCN to be classified as Tier II is 0.9975. From Fig.12, the TS-based DCN availability is lesser than 0.9975.

In such a case, the operator may upgrade the TSs to P4-HW switches. Fig. 13 shows the increase in availability when the traditional DCN (using the Fb-Fab topology) is upgraded with P4-HW switches. From this graph, to be classified as Tier II, at

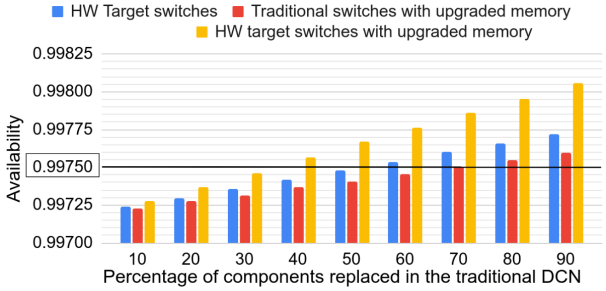


Figure 14: Upgrading the traditional DCN

least 60% of the TSs in the DCN must be replaced by P4-HW switches.

A similar improvement can also be observed by replacing the TSs with upgraded memory subcomponents, for example. This is seen in Fig. 14. The analysis in Sec. 3-C identifies the most impactful parameters and the most influential subcomponents. Improving these subcomponents improves the entire network's availability. In our analysis, the memory subcomponent seems to be the most vulnerable. Therefore, the operator may try to upgrade his memory subcomponent. This is seen by the red bars. If this method is employed, at least 70% of the switches need to be upgraded for the DCN to achieve the recommended availability to be classified as Tier II.

A quicker method to improve availability is using a mix of both approaches. This is shown in the orange bars of Fig. 14. Here, P4-HW target switches with upgraded memory subcomponents have replaced the TSs. In this approach, only 40% of the switches in the traditional DCN need to be replaced to upgrade the DCN classification to Tier II. It has to be noted that the correct approach to improve DCN availability ultimately depends upon the operator's requirements, traffic estimation, and, most importantly, the incurred expenses.

7. Conclusion

This work gives a hierarchical approach to evaluate the availability of a commercial-grade DCN by modeling the switches and their subcomponents. This work is a guideline to the DCN operator on identifying the critical subcomponents of the network. It further provides insight into modeling different types of switches and integrates these findings into DCN topologies of different sizes. The goal of this work is to give an idea of how to improve DCN availability. The following inferences can be made from this work.

(i) Data sheets provide MTBF or related values that are difficult to reproduce in real-world scenarios.
(ii) Subcomponents must be pessimistically modeled to guarantee strict availability requirements.
(iii) For a DCN, the topology does not influence the availability to a large extent.
(iv) The type of switches and the subcomponents greatly influence the DCN availability.
(v) A hybrid DCN with different types of switches is the most practical and feasible solution to improve availability.

Similar work can be extended to cover networks like core and wireless networks in the future. Another direction would be

to analyze the cost incurred due to upgrading the DCN using different approaches. A cost analysis would further help the operator make a well-informed decision to improve the DCN availability.

ACKNOWLEDGMENT

This work has been funded by the Bavarian Ministry of Economic Affairs, Regional Development, and Energy under the project '6G Future Lab Bavaria'.

REFERENCES

[1] Cisco, "Cisco annual internet report (2018–2023) white paper," Cisco: San Jose, CA, USA, 2020.

[2] Cisco Visual Networking Index, "Forecast and methodology, 2015–2020," Cisco, San Jose, CA, USA, pp. 1–41, 2016.

[3] Uptime Institute, "Uptime institute's 2022 global data center survey reveals strong industry growth as operators brace for expanding sustainability requirements," Uptime Institute- Press Release, Jun 2022.

[4] Uptime Institute, "Uptime institute's 2022 outage analysis finds downtime costs and consequences worsening as industry efforts to curb outage frequency fall short," Uptime Institute- Press Release, Sep 2022.

[5] M. J. Loveridge, G. Remy, N. Kourra, R. Genieser, A. Barai, M. J. Lain, Y. Guo, M. Amor-Segan, M. A. Williams, T. Amietszajew, et al., "Looking deeper into the galaxy (note 7)," Batteries, vol. 4, no. 1, p. 3, 2018.

[6] A. Rudolph, "What is log4j and why did the government of canada turn everything off?," 2022.

[7] J. Luszcz, "Apache struts 2: how technical and development gaps caused the equifax breach," Network Security, vol. 2018, no. 1, pp. 5–8, 2018.

[8] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?," Nature Electronics, vol. 3, no. 1, pp. 20–29, 2020.

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

[10] J. Meza, T. Xu, K. Veeraraghavan, and O. Mutlu, "A large scale study of data center network reliability," in Proceedings of the Internet Measurement Conference 2018, pp. 393–407, 2018.

[11] R. Potharaju and N. Jain, "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, pp. 1–17, 2013.

[12] S. Janardhanan and C. Mas-Machuca, "Modeling and evaluation of a data center sovereignty," in 2022 18th International Conference on the Design of Reliable Communication Networks (DRCN), pp. 1–8, 2022.

[13] R. Singh, M. Mukhtar, A. Krishna, A. Parkhi, J. Padhye, and D. Maltz, "Surviving switch failures in cloud datacenters," ACM SIGCOMM Computer Communication Review, vol. 51, no. 2, pp. 2–9, 2021.

[14] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Y. Zomaya, "On the characterization of the structural robustness of data center networks," IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 1–1, 2013.

[15] M. Manzano, K. Bilal, E. Calle, and S. U. Khan, "On the connectivity of data center networks," IEEE Communications Letters, vol. 17, no. 11, pp. 2172–2175, 2013.

[16] R. D. S. Couto, S. Secci, M. E. M. Campista, and L. H. M. K. Costa, "Reliability and survivability analysis of data center network topologies," Journal of Network and Systems Management, vol. 24, no. 2, pp. 346–392, 2016.

[17] R. S. Couto, M. E. M. Campista, and L. H. M. K. Costa, "A reliability analysis of datacenter topologies," in 2012 IEEE Global Communications Conference (GLOBECOM), pp. 1890–1895, 2012.

[18] R. Alshahrani and H. Peyravi, "Modeling and simulation of data center networks," in Proceedings of the 2nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, pp. 75–82, 2014.

[19] T. A. Nguyen, D. Min, E. Choi, and T. D. Tran, "Reliability and availability evaluation for cloud data center networks using hierarchical models," IEEE Access, vol. 7, pp. 9273–9313, 2019.

[20] P. Vizarreta, P. Heegaard, B. Helvik, W. Kellerer, and C. M. Machuca, "Characterization of failure dynamics in sdn controllers," in 2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM), pp. 1–7, IEEE, 2017.

[21] Cisco, "Cisco catalyst 9600 series switches data sheet," Cisco, San Jose, CA, USA, Sep 2022.

[22] A. Sivaraman, C. Kim, R. Krishnamoorthy, A. Dixit, and M. Budiu, "Dc. p4: Programming the forwarding plane of a data-center switch," in Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, pp. 1–8, 2015.

[23] F. Hauser, M. Häberle, D. Merling, S. Lindner, V. Gurevich, F. Zeiger, R. Frank, and M. Menth, "A survey on data plane programming with p4: Fundamentals, advances, and applied research," arXiv preprint arXiv:2101.10632, 2021.

[24] H. Harkous, M. Jarschel, M. He, R. Pries, and W. Kellerer, "P8: P4 with predictable packet processing performance," IEEE Transactions on Network and Service Management, vol. 18, no. 3, pp. 2846–2859, 2020.

[25] S. Janardhanan and C. Mas-Machuca, "Modeling and evaluation of a data center sovereignty with software failures," in 2022 6th International Conference on System Reliability and Safety (ICSRS), pp. 233–242, 2022.

[26] D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. Doyle, W. Sanders, and P. Webster, "The mobius framework and its implementation," IEEE Transactions on Software Engineering, vol. 28, no. 10, pp. 956–969, 2002.

[27] Cisco, "Cisco catalyst 9600 series supervisor engine data sheet," Cisco, San Jose, CA, USA, Sep 2022.

[28] R. Govindan, I. Minei, M. Kallahalla, B. Koley, and A. Vahdat, "Evolve or die: High-availability design principles drawn from googles network infrastructure," in Proceedings of the 2016 ACM SIGCOMM Conference, pp. 58–72, 2016.

[29] G. Wang, L. Zhang, and W. Xu, "What can we learn from four years of data center hardware failures?," in 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 25–36, IEEE, 2017.

[30] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica, "Surviving failures in bandwidth-constrained datacenters," ACM SIGCOMM Computer Communication Review, vol. 42, no. 4, pp. 431–442, 2012.

[31] R. Potharaju and N. Jain, "Demystifying the dark side of the middle: A field study of middlebox failures in datacenters," in Proceedings of the 2013 conference on Internet measurement conference, pp. 9–22, 2013.

[32] Compass Mining, "How long do ASICs last?," Apr 2021.

[33] T. A. Nguyen, D. S. Kim, and J. S. Park, "A comprehensive availability modeling and analysis of a virtualized servers system using stochastic reward nets," The Scientific World Journal, vol. 2014, 2014.

[34] Intel Enterprise Platform and Services Division, "Intel server board s1200v3rp, intel server system p4308rplshdr, r1208rpmshor, r1208rposhor, r1304rpssfbn, r1304rposhbn, r1304rpmshor," Calculated MTBF Estimates, May 2013.

[35] Intel, "Intel server board s1200v3rp data sheet," Technical Product Specification, Jun 2015.

[36] Cisco, "Cisco catalyst 9600 series line cards data sheet," Cisco, San Jose, CA, USA, Oct 2022.

[37] B. Schroeder, E. Pinheiro, and W.-D. Weber, "Dram errors in the wild: a large-scale field study," Communications of the ACM, vol. 54, no. 2, pp. 100–107, 2011.

[38] C. Oggerino, High availability network fundamentals. Cisco Press, 2001.

[39] W. Hou, "Integrated reliability and availability aanalysis of networks with software failures and hardware failures," 2003.

[40] S. K. Uzaman, J. Shuja, T. Maqsood, F. Rehman, S. Mustafa, et al., "A systems overview of commercial data centers: initial energy and cost analysis," International Journal of Information Technology and Web Engineering (IJITWE), vol. 14, no. 1, pp. 42–65, 2019.

[41] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," ACM SIGCOMM computer communication review, vol. 38, no. 4, pp. 63–74, 2008.

[42] N. Farrington and A. Andreyev, "Facebook's data center network architecture," in 2013 Optical Interconnects Conference, pp. 49–50, 2013.

[43] R. R. Reyes and T. Bauschert, "Infrastructure cost comparison of intra-data centre network architectures," in 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–7, IEEE, 2018.

[44] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: A scalable and flexible data center network," in Proceedings of the ACM SIGCOMM 2009 conference on Data communication, pp. 51–62, 2009.

[45] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, pp. 267–280, 2010.

[46] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," ACM SIGCOMM Computer Communication Review, vol. 40, no. 1, pp. 92–99, 2010.

[47] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, pp. 123–137, 2015.

[48] T. Benson, A. Anand, A. Akella, and M. Zhang, "Microte: Fine grained traffic engineering for data centers," in Proceedings of the seventh conference on emerging networking experiments and technologies, pp. 1–12, 2011.

[49] Z. Hu, Y. Qiao, and J. Luo, "Atme: Accurate traffic matrix estimation in both public and private datacenter networks," IEEE Transactions on Cloud Computing, vol. 6, no. 1, pp. 60–73, 2015.

[50] W. P. Turner IV, J. PE, P. Seader, and K. Brill, "Tier classification define site infrastructure performance," Uptime Institute, vol. 17, 2006.