ANALYSIS AND CHARACTERIZATION OF CLOUD BASED DATA CENTER ARCHITECTURES FOR PERFORMANCE, ROBUSTNESS, ENERGY EFFICIENCY, AND THERMAL UNIFORMITY

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Kashif Bilal

In Partial Fulfillment
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Electrical and Computer Engineering

April 2014

Fargo, North Dakota

North Dakota State University
Graduate School

Analysis and Characterization of Cloud Based Data Center Architectures for Performance, Robustness, Energy Efficiency, and Thermal Uniformity

Kashif Bilal

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Samee U. Khan
Chair

Jacob S. Glower

Sudarshan K. Srinivasan

Ying Huang

Approved:

06/09/2014
Date

Scott Smith
Department Chair

ABSTRACT

Cloud computing is anticipated to revolutionize the Information and Communication Technology (ICT) sector and has been a mainstream of research over the last decade. Today, the contemporary society relies more than ever on the Internet and cloud computing. However, the advent and enormous adoption of cloud computing paradigm in various domains of human life also brings numerous challenges to cloud providers and research community. Data Centers (DCs) constitute the structural and operational foundations of cloud computing platforms. The legacy DC architectures are inadequate to accommodate the enormous adoption and increasing resource demands of cloud computing. The scalability, high cross-section bandwidth, Quality of Service (QoS) guarantees, privacy, and Service Level Agreement (SLA) assurance are some of the major challenges faced by today's cloud DC architectures. Similarly, reliability and robustness are among the mandatory features of cloud paradigm to handle the workload perturbations, hardware failures, and intentional attacks. The concerns about the environmental impacts, energy demands, and electricity costs of cloud DCs are intensifying. Energy efficiency is one of mandatory features of today's DCs.

Considering the paramount importance of characterization and performance analysis of the cloud based DCs, we analyze the robustness and performance of the state-of-the-art DC architectures and highlight the advantages and drawbacks of such DC architecture. Moreover, we highlight the potentials and techniques that can be used to achieve energy efficiency and propose an energy efficient DC scheduling strategy based on a real DC workload analysis. Thermal uniformity within the DC also brings energy savings. Therefore, we propose thermal-aware scheduling policies to deliver the thermal uniformity within the DC to ensure the hardware

reliability, elimination of hot spots, and reduction in power consumed by cooling infrastructure. One of the salient contributions of our work is to deliver the handy and adaptable experimentation tools and simulators for the research community. We develop two discrete event simulators for the DC research community: (a) for the detailed DC network analysis under various configurations, network loads, and traffic patterns, and (b) a cloud scheduler to analyze and compare various scheduling strategies and their thermal impact.

ACKNOWLEDGEMENTS

All praises and thanks to Allah almighty, my Creator, my Sustainer, for giving me courage and strength to pursue my PhD and fulfill the requirements of this disquisition.

My heartiest and sincere appreciation and gratitude to my mentor and adviser Dr. Samee U. Khan, who always encouraged me, and persistently conveyed the spirit and guidance required for the research. Without his kind guidance and continuous efforts, this disquisition would not have been possible.

Special thanks to my committee members, Dr. Jacob S. Glower, Dr. Sudarshan K. Srinivasan, and Dr. Ying Huang for their support, guidance and helpful recommendations. Thanks to the Electrical and Computer Engineering staff members Jeffrey Erickson, Laura D. Dallman, and Priscilla Schlenker for all the unconditional help and favor. I owe my heartiest thanks to all my friends and colleagues here in the US and Pakistan, who always helped me in the time of need.

Finally, I would like to thank my family. Their continuous support is always a source of motivation and encouragement for me. I especially like to thank my mother, who is the only and every reason for whatever I am today and whatever I achieved in my life. I also would like to thank my loving wife and my kids, Ibrahim and Eishal, for their patience, time, and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Cloud Computing

Cloud computing is foreseen to be the next major paradigm shift in the Information and Communication Technology (ICT). Today, the contemporary society relies more than ever on the Internet and cloud computing. Cloud investments have delivered around $4B of return on the investment in the last five years [1.1]. According to a report published in January 2013 by Gartner, the overall Public Cloud services are anticipated to grow by 18.4% in 2014 to $155B market [1.2]. Moreover, the total market is expected to grow from $93B in 2011 to $210B in 2016. Cloud computing offers promising incentives to the ICT sector. Among the foremost incentives offered by cloud computing are: (a) ease and pervasive (anytime, anywhere) access to the data and applications, and (b) cost effectiveness. Significant savings in the initial Capital Expenditure (CapEx) and Operational Expenses (OpEx) inspire enterprises and businesses to adopt cloud services for their computing demands. Enormous budget for the deployment of computing infrastructure is no more a pressing concern for the enterprises by utilizing the cloud. Moreover, cloud also helps in reducing the running (OpEx) costs by (a) minimizing the required Information Technology (IT) staff, (b) relieving the data security and backup concerns, and (c) reducing the utility (energy) bills. The employees can access cloud-based services anywhere, anytime using smart devices. Pervasive and convenient access to the enterprise data and applications augment the productivity. Moreover, cloud computing also offers to procure the computing and storage resources when required on the fly. The enterprises can procure and release the cloud resource for their short-term needs based on “pay per use” policy.

Many enterprises and small businesses are adopting cloud for their ICT needs. In the "Market Trends" report by Gartner, it is estimated that the cloud-based business services and Software-as-a-Service (SaaS) market will increase from $13.4 to $32.2 billion from 2011 to 2016. Similarly, Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) market is estimated to grow from $7.6 billion to $35.5 billion from 2011 to 2016. Besides augmenting various dimensions of the business and enterprise, cloud computing is also transforming various aspects of the social and personal life. For instance, social networking has minimized the communication gap, and the users are connected through cloud. Cloud facilitates the downloading and updating of various apps (mobile applications). Pictures, videos, files, and reviews are shared via cloud. Moreover, cloud gaming enables the users to play the state-of-the-art games online at low performance endpoints, such as smart phones. Besides offering a rich set of online players to play with, all of the game processing and rendering is performed at cloud for real-time gaming.

The benefits offered by cloud computing, such as unlimited resources at nominal prices are attracting the research organization to utilize cloud for their computation and data storage requirements. Eli Lilly, a medicine company executed a complex bioinformatics job on 64-machines cluster at cloud that cost $6.4 only. Various research domains, such as scientific applications, agriculture, nuclear science, healthcare, smart grids, and e-Commerce are increasingly employing cloud computing for their research needs.

1.2. Data Centers

The data center is a pool of computational and storage resources clustered together using the communication infrastructure. Data centers constitute the building blocks and underlying

foundations of cloud computing. Cloud computing relies on data centers to deliver the required resources and services. The growing adoption of cloud services mandates the growth in computational and storage resources. Cloud service providers already have hundreds of thousands of servers in their data centers. Google is estimated to have around 0.9 million servers in their data centers. The number of servers in the Microsoft data centers double every fourteen months. The number of servers in Facebook data center doubled within six months from 2009 to 2010. Amazon cloud services are supported by a data center having around 454,000 servers. Growing the number of servers in the data center is not a problem. However, interconnecting these servers to deliver high inter-server communication bandwidth and required QoS is the major challenge. Today's data centers are not constrained by the computational power; they are limited by the interconnection network [1.3].

The major Information and Communication Technology (ICT) components within the data centers are: (a) servers, (b) storage, and (c) interconnection network. Data Center Network (DCN) being the communication backbone of the data center is one of the foremost design concerns in the data center [1.4]. The DCN infrastructure plays a vital role to ascertain the performance aspects and initial capital investment in the data center.

1.3. Motivation and Objectives

The DC architecture holds a pivotal role to ascertain the performance boundaries and initial capital investment of the cloud infrastructure. The DC architecture must ensure the Quality of Service (QoS) and reliability of the cloud paradigm. Therefore, appropriate analysis and characterization of the DC architectures is mandatory. Moreover, the concerns about the environmental impacts, energy needs, and electricity cost of DCs are rising. DCs are required to

be energy efficient and energy proportional. Therefore, energy efficient and workload consolidation based scheduling techniques are needed. Furthermore, enormous power consumption by the DC resources releases excessive heat. The DC environment needs to be cooled down to ensure the reliable functioning of hardware. Substantial amount of energy is consumed by the air conditioning and cooling infrastructure to control the thermal signature of the DC. Hotspots, excessive heat, and uneven thermal signature within a DC lead to energy wastage by the cooling unit and malfunctioning of the hardware in the hotspot areas. Therefore, thermal aware scheduling and dynamic workload migrations are required to maintain the thermal uniformity within a DC to eliminate hotspots and minimize energy consumed by the cooling infrastructure. Finally, detailed simulation frameworks are required to rigorously analyze and test the solutions proposed by the research community, as the realistic testing is economically unviable. Therefore, new simulation frameworks and toolkits are required to help the cloud research community. In our work, we aim to address the aforementioned challenges faced by today's data centers and cloud paradigm. Following are some of our major objectives:

- We analyze and characterize various data center architectures to highlight their advantages and drawbacks to select the most appropriate data center architecture to ensure the: (a) performance and (b) reliability of the cloud paradigm. We develop a simulation framework for performance analysis, and proposed two novel metrics to quantify the robustness and connectivity of the DC architectures.
- We focus on energy efficiency of the cloud paradigm that is one of the foremost requirements of the cloud computing. We highlight the potentials and techniques that can be used to achieve energy efficiency. We also propose energy efficient

and thermal efficient DC scheduling strategies based on a real DC workload analysis.

- We aim to deliver the thermal uniformity within the data center to ensure the hardware reliability, elimination of hot spots, and reduction in power consumed by cooling infrastructure.
- One of the salient contributions of our work is to deliver the handy and adaptable experimentation tools and simulators for the research community. Today, negligible simulation and experimentation toolkits are available for the cloud research community for testing and analysis. Therefore, such simulation toolkits and frameworks are one of the major requirements of cloud computing research community.

1.4. Contributions

1.4.1. Analysis and Characterization of the State-of-the-Art Data Center Architectures

The legacy DC architectures, such as the ThreeTier architecture are unable to handle the growth and adoption trend of the cloud paradigm. Consequently, various new DC architectures have been proposed in the literature to overcome the challenges faced by the legacy cloud based DCs. However, DC architecture being the architectural and functional foundation of the cloud paradigm must ensure: (a) performance, (b) high cross-section bandwidth, (c) robustness and reliability, and (d) connectivity. Therefore, we analyzed three state-of-the-art DC architectures, namely: (a) ThreeTier, (b) FatTree, and (c) DCell, for performance, connectivity, and robustness. Fundamental aim of the analysis is to ensure the suitability, performance, and reliability for the

cloud paradigm, and to highlight the salient features and drawbacks of the considered DC architectures.

We focused the robustness and connectivity of the DC architectures in case of failures and perturbations. We analyzed the three DC architectures using the classical robustness metrics. Our analysis revealed that the classical robustness metrics are inadequate to quantify the robustness of the DC architectures. Therefore, we proposed a novel robustness measures for the robustness quantification of the DC architectures named, deterioration. Our robustness analysis work is published in the *IEEE Transactions on Cloud Computing* [1.5]. Moreover, we performed the connectivity analysis of the DC architectures in case of: (a) random and (b) targeted failures. We proposed a new connectivity metric named, μ-A2TR. Our connectivity analysis of the DC architectures is published in the *IEEE Communications Letters* [1.6].

Besides robustness analysis, we develop a simulation framework to implement the DC architectures and performed rigorous simulations in various scenarios and traffic patterns. We tested 222 different configurations for the aforementioned analysis to quantify the network throughput and delay. Our work is published in the *Concurrency and Computation: Practice and Experience journal* [1.4].

1.4.2. Energy Efficiency in the Cloud based DCs

Data center consume around 30% – 80% more energy per square meter as compared to a traditional office space. Energy cost of a data center dominates the total Operating Expenses (OpEx), for instance around 45% of total OpEx in the IBM. Similarly, the concerns related to environmental aspects and Green House Gases (GHG) footprints of data centers are also intensifying. Therefore, energy efficiency is one of the most required and crucial features of

today's data centers. The cloud resources are overprovisioned to handle the peak load. Therefore, a plenty of resources remain idle within DCs most of the time. As reported by the IBM, around 85% of the computing capacity of the distributed systems remains idle. We reviewed various potentials in the state-of-the-art DC architectures that can be exploited to save energy. We published thorough reviews on energy efficiency potentials and challenges in DC architectures and DC networks in *Future Generation Computer Systems* [1.7], *Cluster Computing* [1.8], and *IEEE FIT'13* [1.9].

We analyzed a real data center workload and observed that most of the time the servers remained idle. Consolidating the workload on fewer servers and placing the idle servers into sleep mode depicted potential to save substantial amount of energy. Therefore, based on our real workload analysis, we proposed a Data Center-wide Energy Efficient Resource Scheduling framework (DCEERS) that schedules data center resources according to the current workload demands of the DC. Our work has been published in the *Cluster Computing* journal [1.10].

1.4.3. DC Thermal Analysis and Thermal Aware Resource Scheduling

We analyzed a real data center workload and thermal signatures to find the thermal impact of the resource utilization within a DC and its ambient effects. We aim to employ, Statistical models, such as mean procedure, correlation, and VARMAX model to find thermal effects of resource utilization. The facts discovered from the thermal analysis of the real workload were employed to define a thermal aware resource allocation and migration strategy to ensure the uniform thermal signature within the DC. The said strategy depicted thermal uniformity across the DC resources reducing the excessive energy used by the cooling

infrastructure in case of uneven thermal signature, and increase the reliability and life of the hardware resources.

1.4.4. DC Experimentation and Simulation Toolkits

The cloud based DCs face various challenges today. These challenges and their proposed solutions require detailed analysis and quantification. In this particular case, simulation is an appropriate solution for the detailed analysis and quantification of various problems faced by the DCs, because experimentation comprised of realistic DC environments are economically unviable. Unfortunately, network models, simulators, and schedulers to quantify the data center network, varying traffic patterns, and thermal impacts at a detailed level are scarce, currently. Therefore, we developed two discrete event simulators for the DC research community: (a) for the detailed DC network analysis under various configurations, network loads, and traffic patterns, and (b) a cloud scheduler to analyze and compare various scheduling strategies and their thermal impact. The aforementioned simulators will aid the cloud research community to implement and analyze the impact of various network protocols and scheduling policies.

1.5. References

[1.1] IBM, *Get more out of cloud with a structured workload analysis*, Oct. 2011.

[1.2] Gartner, *Forecast overview: Public cloud services, worldwide, 2011-2016, 4Q12 Update*, Gartner Inc., 2013.

[1.3] K.Bergman, "Optically Interconnected High Performance Data Centers," *Optical Communication (ECOC), 2010 36th European Conference and Exhibition on*, Sep. 2010, pp. 1-3

[1.4] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C.-Z. Xu, and A. Y. Zomaya, "Quantitative Comparisons of the State of the Art Data Center Architectures," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 12, pp. 1771-1783, 2013.

[1.5] 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. 64-77, 2013.

[1.6] 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.

[1.7] K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, U. S. Khan, A. Abbas, N. Jalil, and S. U. Khan, "A Taxonomy and Survey on Green Data Center Networks," *Future Generation Computer Systems*. (DOI: 10.1016/j.future.2013.07.006.)

[1.8] K. Bilal, S. U. Khan, S. A. Madani, K. Hayat, M. I. Khan, N. Min-Allah, J. Kolodziej, L. Wang, S. Zeadally, and D. Chen, "A Survey on Green Communications using Adaptive Link Rate," *Cluster Computing*, vol. 16, no. 3, pp. 575-589, 2013.

[1.9] K. Bilal, S. U. Khan, and A. Y. Zomaya, "Green Data Center Networks: Challenges and Opportunities," In *11th IEEE International Conference on Frontiers of Information Technology*, Pakistan, 2013, pp. 229-234.

[1.10] J. Shuja, K. Bilal, S. A. Madani, and S. U. Khan, "Data Center Energy Efficient Resource Scheduling," *Cluster Computing*. (DOI 10.1007/s10586-014-0365-0.)

2. BACKGROUND AND RELATED WORK

2.1. Data Center Network Architectures

Data center network architecture is one of the most significant components of large-scale data centers, which wields a great impact on the general data center performance and throughput. Numerous empirical and simulation analysis show that almost 70% of network communication takes place within a data center [2.1]. The cost of the implementation of the conventional two-tier and three-tier-like DCN architectures is usually too high and makes the models virtually ineffective in the large-scale dynamic environments [2.2]. Over the last few years, the fat-tree based and the recursively defined architectures are presented as the most promising core structures of the modern scalable data centers. On the basis of the different types of the traffic routing models, the DCN architectures can be classified into the following three basic categories: (a) switch-centric models [2.3], (b) hybrid models (using server and switch for packet forwarding [2.4]), and (c) server-centric models [2.5].

The switch-centric DCN architectures rely on the network switches to perform routing and communication in the network, such as three-tier architecture and the fat-tree based architecture [2.3]. Hybrid architectures use a combination of switches and servers that usually are con figured as routers within the network to accomplish routing and communication, such as DCell [2.4]. The server-centric architectures do not use switches or routers. The basic components of such models are servers that are configured as computational devices and data and message processing devices.

2.1.1. ThreeTier Architecture

The legacy three-tier architecture is by far the most extensively used DCN architecture [2.6]. In the three-tier architecture, the switches are primarily arranged in three layers: (a) access, (b) aggregate, and (c) core (Fig. 2.1). The pool of servers is thereby connected to access layer switches. The core layer makes the foundation of the network tree, and each core layer switch is connected successively to all of the aggregate layer switches. High-end enterprise switches are usually used at aggregation and core layers, rendering three-tier DCN an excessively expensive and power hungry architecture [2.3]. Different layers of three-tier architecture are oversubscribed at different threshold values. Variation in the oversubscription ratio at the various network layers is based on the physical infrastructure. The oversubscription is de fined for optimizing the cost of the system design. Oversubscription can be calculated as a ratio of worst-case aggregated bandwidth available to end hosts and the overall bisection bandwidth of the network topology [2.3]. For instance, the oversubscription 4:1 means that the communication pattern may use only 25% of the available bandwidth. The typical oversubscription values are between 2.5:1 and 8:1, and 1:80 to 1:240 for the paths near the root at highest level of system hierarchy [2.3].

2.1.2. FatTree Architecture

The basic model of the fat-tree DCN architecture has been proposed by Al-Fares et al. [2.3]. The fat-tree model is promoted by the authors as an effective DCN architecture by using a set of commodity switches to provide more end-to-end bandwidth at a considerably lower monetary cost and energy consumption as compared with the high-end network switches. The proposed solution is backward compatible, and only requires modification in the switch

forwarding functions. The fat-tree based DCN architecture aims to provide 1:1 oversubscription ratio.

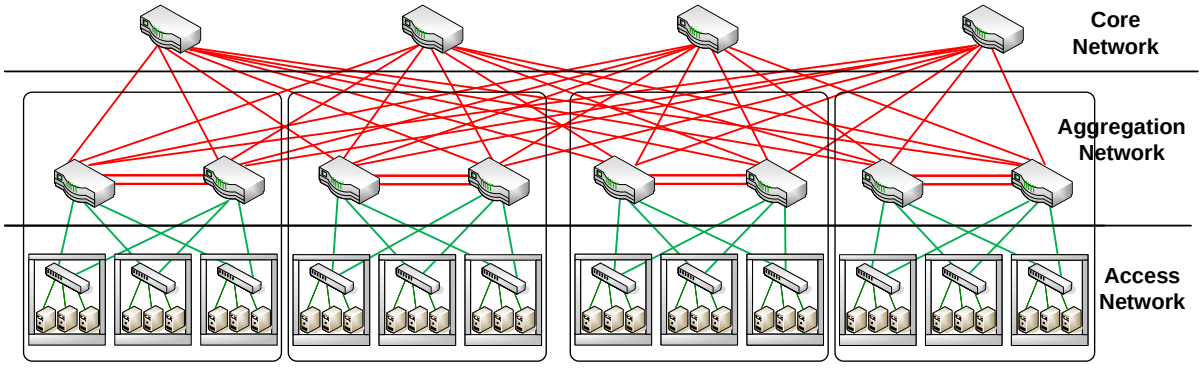


Fig. 2.1. ThreeTier Architecture.

Al-Fares et al. [2.3] adopted a special topology called fat-tree topology. The network structure is composed of n pods. Each pod contains n servers and n switches organized in two successive layers of n/2 switches. Every lower layer switch is connected to n / 2 hosts in the pod and n / 2 upper layer switches (making the aggregation layer) of the pod. There are (n / 2) 2 core level switches, each connecting to one aggregation layer switch in each of n pods.

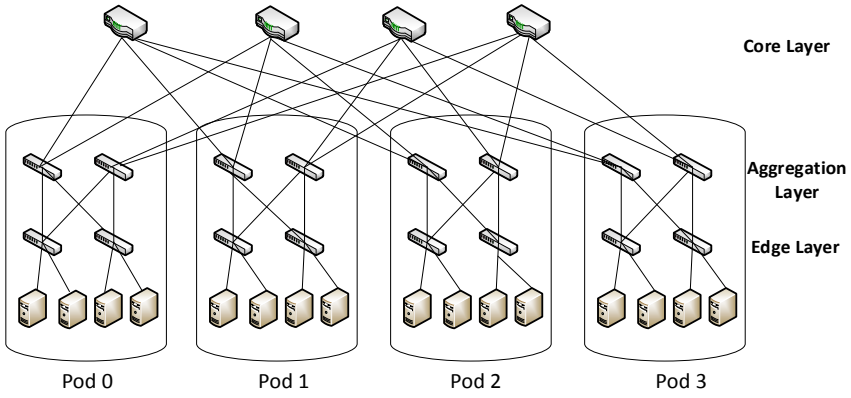


Fig. 2.2. FatTree Architecture.

The interconnection of servers and switches for n =4 pods is presented in Fig. 2.2. The fat-tree based DCN architecture [2.3] uses a customized routing protocol, which is based on primary prefix and secondary suffix lookup for next hop. Routing table is divided into two levels. For each incoming packet, destination address pre fi x entries are matched in primary table. If longest pre fi x match is found, then the packet is destined to the specified port. If there is no match, then the secondary level table is used, and the port entry with longest suffix match is used to forward the packets.

2.1.3. DCell Architecture

A recursively defined DCN architecture, referred to as the DCell model was reported in [2.4]. In this model, the whole system is organized in the cells or pods with n servers and a commodity switch. A 0 level cell $DCell_0$ serves as the building block of the whole system. A $DCell_0$ comprises of *n* commodity servers and a mini network switch. Higher levels of cells are built by connecting multiple lower level ($level_{l-1}$) DCells. Each $DCell_{l-1}$ is connected to all of the other $DCell_{l-1}$ within the same $DCell_l$. The DCell provides an extremely scalable architecture. A 4-level DCell, having six servers in $DCell_0$ can accommodate around 3.26 million servers. Fig. 2.3 shows a level-2 DCell having two servers within each $DCell_0$. The figure shows the connection of only $DCell_{1[0]}$ to all other $DCell_1$. Unlike the conventional switch-based routing used in the hierarchical and fat-tree based DCN architectures, the DCell uses a hybrid routing and data processing protocol. Switches are used to communicate among the servers within the same $DCell_0$. The communication with servers in other DCells is performed by servers acting as routers. In fact, computational servers are also considered as the routers in the system. The DCell routing scheme is used in the DCell architecture to compute the path from the source to destination node exploiting divide and conquer approach. Source node (s) computes the path

from s to destination (d). The link that interconnects the DCells that contain the s and d at the same level is calculated first, and then the sub-paths from s to link and from link to d are calculated. Combination of both of the sub-paths results in the complete routing path between s and d. The DCell routing is not a minimum hop routing scheme. Therefore, the calculated route possesses more hops than the shortest path routing.

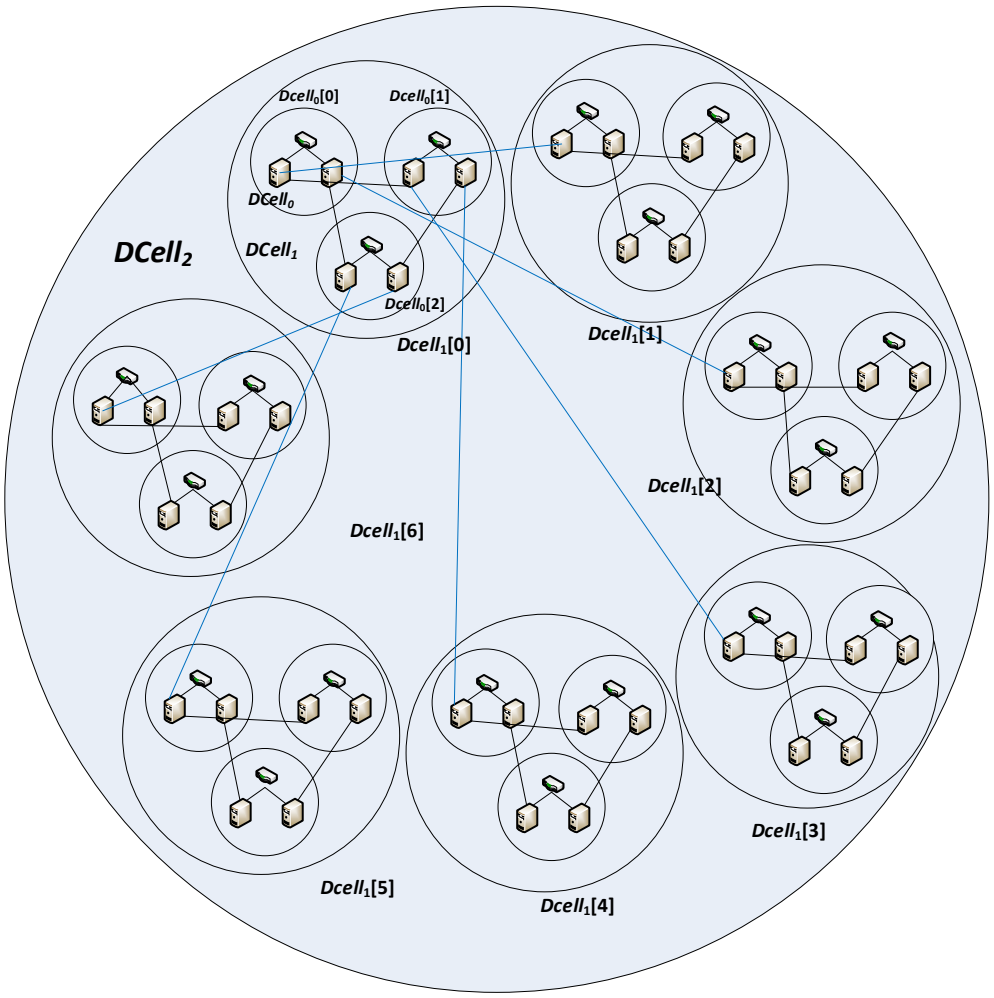


Fig. 2.3. DCell Architecture.

2.1.4. DCN Architectures: Issues, Solutions, and Potentials

Servers are interconnected to each other in the recursively defined server-centric DCN architectures. Such DCN architectures do not use multiple layers of network switches. Therefore, the network links interconnecting the cells experience high oversubscription ratio (up to 256:1 for 4096 nodes) [2.7]. The recursively defined DCNs exhibit strong reliance on the network size. The results of our simulation analysis show that the network throughput and inter-node bandwidth are inversely proportional to the network size in the DCell. Moreover, the servers also perform the additional task of traffic processing and routing within the server-centric routing model, which usually requires a dedicated processor/core. Furthermore, the routing schemes used in the recursive DCN architectures are usually not based on the shortest path routing. The path between the source and destination may possess additional intermediate hops, which results in higher packet delays and increased link utilization. We have simulated the DCell architecture with the DCell's customized routing and shortest path routing schemes. Our simulation results, demonstrate that the shortest path routing outperforms the DCell's routing in terms of network throughput and average packet delay. Because servers are used to inter-connect cells, the idle servers cannot be placed into sleep state. Therefore, Dynamic Power Management (DPM) techniques for energy savings are not feasible for the recursive DCNs, resulting in continuously full energy consumption despite being idle. Details of the simulation analysis can be seen in [2.7].

The server-centric DCNs inherently save energy that is used in the switch-centric DCNs by network switches. The network traffic flows may be managed by applying load-balancing techniques to overcome the network congestion problem. The DCell architecture also exhibits path diversity from source to destination. Network flow based adaptive routing protocols may

exploit the path diversity to select the best and most conducive path from the source to destination.

The switch-centered DCNs exhibit better candidature for energy efficiency, because of high path overprovisioning. One of the major drawbacks of the contemporary switch-centric networks is the use of a large number of network switches to ensure 1:1 oversubscription ratio. For example, the 8-pod (128 nodes) fat-tree network requires 80 network switches [2.7]. There is a great deal of overprovisioning and path diversity in the switch-centric networks. Moreover, the average link utilization of network links is reported around 5% - 25% [2.8]. Therefore, underutilization of the links and path diversity may be exploited for energy efficiency. For instance, ALR based techniques can be very conveniently applied to save energy. Moreover, end-to-end path diversity offers an opportunity for network traffic consolidation and re-routing on a subset of links and devices. Therefore, remaining idle devices may be transitioned to the low power sleep mode. Furthermore, IEEE 802.3az EEE may be employed to increase energy savings in amalgamation with other energy efficiency techniques.

Hybrid (electrical + optical/wireless) DCNs offer solutions to various DCN problems. Wireless connectivity offers a feasible solution to extend existing DCN infrastructure eliminating cabling cost, complexity, and installation. Wireless links can be created among server and racks on the fly, reducing the network load on the core network. Energy efficiency is one of the foremost design requirements for 60 GHz technology, resulting in energy efficient 60 GHz devices and technology. Wireless interconnects can be exploited to migrate the traffic from underutilized network devices to wireless links to place the idle devices in sleep mode for energy saving. Optical interconnects offer higher port density and bandwidth at considerably low energy

consumption. Estimated energy consumption of an optical transmitter is around 0.5nJ/bit, whereas the energy consumed by a high-end electrical router is around 20nJ/bit. Consequently, overall energy consumption can be reduced significantly. A complete optical DCN is estimated to deliver around 75% energy savings. Hybrid DCNs can be used to relieve the hotspots (congested links) in the DCN. Elephant flows and high fan in/out traffic are considered the chief reasons for the network congestion and performance deprivation. 60 GHz wireless flyways and optical circuit switched paths offer load balancing opportunities by offloading elephant flows from electrical switches to reduce network load and congestion.

2.1.5. Green DCN Challenges

New DCN architectures are required to cater to the increasing GHG emissions produced by the ICT sector. DCNs typically experience an average load of not more than 25% of the peak load. Moreover, around 70% of the time, a considerable number of the links remain idle within data centers [2.9]. However, the links do not remain idle constantly for long periods of time. Benson et al. analyzed data center traffic over a period of ten days and observed that the set of idle links continuously varied for the entire time period. Moreover, it also was observed that 80% of the links remained idle only for 0.002% of the time [2.9]. Therefore, it is important to consider the traffic characteristic within a data center prior to applying ALR or other energy saving techniques.

Overprovisioning and underutilization of links and devices enable opportunity for energy efficiency techniques. However, due consideration is required to be given to the QoS and performance constraints. Performance degradation and increased latency may result in substantial revenue loss. Google reported 20% revenue loss because of an experiment that added

an extra delay of 500ms in displaying the search results. Amazon experienced 1% sales decrease because of 100ms additional delay [2.10]. Therefore, green networking initiatives must be reliable and ensure required performance and QoS constraints.

Hybrid DCNs offer promising opportunities to DCNs. However, hybrids DCNs are in their infancy and are facing numerous challenges. 60 GHz wireless technology is limited by line of sight, short range, propagation loss, and signal attenuation. 60 GHz technology poses serious challenges in transceiver positioning, beam forming, interference due to power leaks, and signal reflection in densely populated data centers. Similarly, optical interconnects also experience numerous challenges, such as cost, scalability, link setup, switching time, and insertion loss. The wavelength switching time for commercially available optical switches is around 10 to 25ms. Moreover, hybrid networks lack sufficient efficacy for DCNs multi-tenant based mixed and heterogeneous workloads. Various hybrid DCN architectures make stringent overlay assumptions, such as that (a) flows are independent, (b) flows do not have priority, and (c) random hashing for flow distribution is effective. However, in practice such assumptions do not hold true for the DCN traffic patterns. Hybrid interconnects promise significant network upgrades. Aforementioned are some of the numerous unresolved challenges that pose a barrier in adopting hybrid technologies in data centers.

Network protocols may be optimized or re-designed for enhanced performance and energy efficiency. Network-aware and energy-aware adaptive routing protocols are needed for better performance, high link utilization, and traffic consolidation and redirection. Moreover, many network services remain active to ratify their availability to periodic heartbeat messages or network chatter. The “interface proxying” techniques may be used to transparently transition

such services to sleep without affecting the network operations [2.11]. The EEE is a promising energy efficient technology but is still in its infancy. Efficient and reliable ALR policies are required to be designed for the IEEE 802.3az EEE.

There are very little details available on characteristics of the data center traffic [2.10]. There is presumably no network workload generator at hand, which may generate data center traffic for various scenarios, such as one-to-one, all-to-all, and one-to-all, and for data intensive, computational intensive, and mixed workloads. A realistic data center traffic generator will substantially help the research community to analyze DCN under various scenarios, and tune the DCN for energy efficiency and reliability.

Energy efficiency of network equipment has not increased following the Dennard's law and current network equipment is not energy-proportional. Energy consumed by network devices in idle state is around 80% - 90% of energy consumed in peak load [2.12]. Energy proportional network devices are required to be designed, and can save enormous amount of wasted energy.

The DCN is one of the most significant data center components wielding a marked impact on initial capital investment and performance parameters. The state of the art DCN are implemented at a very small scale and tested under non-realistic data center traffic [2.7]. There are very less comparative studies for DCNs [2.7], and presumably no comparative study of different DCN architectures under realistic traffic conditions. Different DCN comparative studies under realistic workloads are required to highlight the DCN drawbacks and future research for enhancement.

2.2. References

[2.1] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "Energy Aware Network Operations," *Proceedings of the IEEE Global Internet Symposium*, Apr. 2009.

[2.2] D. Kliazovich, P. Bouvry, and S.U. Khan, "GreenCloud: A Packet-level Simulator for Energy-aware Cloud Computing Data Centers," *Journal of Supercomputing*. (Forthcoming.)

[2.3] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM 2008 conference on Data communication*, Seattle, WA, 2008, pp. 63-74.

[2.4] C.Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A Scalable and Fault-tolerant Network Structure for Data Centers." *ACM SIGCOMM Computer Communication Review*, Vol. 38, No. 4, 2008, pp. 75-86.

[2.5] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O'Shea, and A. Donnelly, "Symbiotic Routing in Future Data Centers," *ACM SIGCOMM 2010 conference*, New Delhi, India, 2010, pp. 51-62.

[2.6] Cisco, *Cisco Data Center Infrastructure 2.5 Design Guide*, Cisco press, 2010.

[2.7] K. Bilal, S.U. Khan, L. Zhang, H. Li, K. Hayat, S.A. Madani, N. Min-Allah, L. Wang, and D. Chen, "Quantitative Comparisons of the State of the Art Data Center Architectures," *Concurrency and Computation: Practice and Experience*, (DOI: 10.1002/cpe.2963).

[2.8] A. Carrega, S. Singh, R. Bruschi, and R. Bolla, "Traffic Merging for Energy-Efficient Datacenter Networks," *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '12)*, Genoa, Italy, July 8–11, 2012.

[2.9] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding Data Center Traffic Characteristics," *SIGCOMM Computer Communication Review*, Vol. 40, No. 1, 2010, pp. 92–99.

[2.10] A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM SIGCOMM Computer Communication Review,* Vol. 39, No. 1, Jan. 2009, pp. 68–79.

[2.11] K. Bilal, S.U. Khan, J. Kolodziej, L. Zhang, K. Hayat, S.A. Madani, N. Min-Allah, L. Wang, and D. Chen, "A Survey on Green Communications using Adaptive Link Rate," *Cluster Computing*, (DOI 10.1007/s10586-012-0225-8). 5100101937325.

[2.12] L. Niccolini, G. Iannaccone, S. Ratnasamy, J. Chandrashekar, and L. Rizzo, "Building a Power Proportional Router," *Usenix ATC '12*, Boston, June 2012.

3. ON THE CHARACTERIZATION OF THE STRUCTURAL ROBUSTNESS OF DATA CENTER NETWORKS

This paper is published in IEEE Transactions on Cloud Computing (TCC), vol. 1, no. 1, pp. 64-77, 2013. The authors of the paper are Kashif Bilal, Marc Manzano, Samee U. Khan, Eusebi Calle, Keqin Li, and Albert Y. Zomaya.

3.1. Introduction

Cloud computing has emerged as a promising paradigm in various domains of the information and communication technology (ICT). Recently, cloud computing has increasingly been employed to a wide range of applications in various research domains, such as agriculture, smart grids, e-commerce, scientific applications, healthcare, and nuclear science [3.1]. Data centers being an architectural and operational foundation of cloud, play a vital role in the economic and operational success of cloud computing. Cloud providers need to adhere and comply with the service-level agreement (SLA) and Quality of Service (QoS) for success. Any violation to the SLA may result in huge revenue and reputation loss. Cloud environment is dynamic and virtualized, with a shared pool of resources [3.2]. Therefore, the resources in the data center are prone to perturbations, faults, and failures. Cloud environment and data center networks (DCNs) need to function properly to deliver required QoS in presence of perturbations and failures [3.3].

DCNs constitute the communicational backbone of a cloud, and hold a pivotal role to ascertain the data center performance and integrity [3.4]. A minor network performance degradation may result in enormous losses. Google reported 20 percent revenue loss, when an experiment caused an additional delay of 500 ms in the response time [3.5]. Moreover, Amazon

reported 1 percent sales decrease for an additional delay of 100 ms in search results [3.5]. Currently, the network robustness quantification of the widely used DCN architectures is unavailable. Therefore, there is an immense need to carry out such a study to quantify the network behavior in the presence of perturbations. A minor failure in the O2 (leading cellular service provider in UK) network affected around seven million customers for three days [3.6]. Similarly, a core switch failure in the BlackBerry's network left millions of customers without Internet access for three days [3.6]. The significance of the interconnection networks is obvious from the aforementioned discussion, providing adequate evidences for the robustness requirement of the network. It can be inferred from the discussion that the network robustness holds a key role to ensure desired level of performance and QoS in cloud computing. In the said perspective, measuring the robustness of the DCN is crucial to identify the behavior and level of performance that a network can attain under perturbations and failure-prone circumstances. Therefore, DCN's robustness is a vital measure for proven performance and fault tolerance in cloud computing.

Network (or also referred to as topology) robustness is the ability of the network to deliver the expected level of performance when one or more components of the network fail [3.7]. Sydney et al. [3.8] defined robustness as the "ability of a network to maintain its total throughput under node and link removal". Ali et al. [3.3] consider a system robust, when the system is able to operate as expected in presence of uncertainties and perturbations. System robustness, and network robustness in particular, has been widely discussed in the literature [3.7], [3.8], [3.9], [3.10], [3.11], [3.12], [3.13], [3.14], [3.15]. Network robustness metrics generally consider the graph theory-based topological features of the network [3.7]. Several metrics, such as the node connectivity, symmetry ratio, shortest path length, diameter, and

assortativity coefficient are used to measure network robustness. However, DCNs exhibit various divergences from the conventional random networks and graph models, such as heterogeneity, multi-layered graph model, and connectivity pattern. DCNs follow a predefined complex architectural and topological pattern, and are generally composed of various layers, such as ThreeTier and FatTree DCNs. Therefore, proper modeling of DCNs is required to measure the robustness.

In this paper, we evaluate various topological features and robustness of the state-of-the-art DCNs namely: 1) ThreeTier [3.16], 2) FatTree [3.17], and 3) DCell [3.18]. Our major contributions include:

- modeling DCN topologies using multilayered graphs;
- developing a DCN graph topology generation tool;
- measuring several robustness metrics under various failure scenarios;
- comparative robustness analysis of the DCN topologies and indicating the inadequacy of the classical robustness metrics to evaluate DCNs;
- proposing new robustness metric for the DCN topologies.

The robustness analysis of the DCN topologies unveiled noteworthy observations. The results revealed that the classical robustness metrics, such as average nodal degree, algebraic connectivity, and spectral radius are unable to evaluate DCNs appropriately. Most of the metrics only consider the largest connected component for robustness evaluation. Consequently, the metrics are unable to depict the factual measurements and robustness of the network. Moreover, none of the DCNs can be declared as more robust based on the measurements taken: 1) without

failure and 2) under various failure scenarios. Therefore, we present a new metric named *deterioration* to quantify the DCN robustness.

3.2. Graph Definitions for DCN Architectures

3.2.1. Previous Definitions

Kurant and Thiran proposed a general multi-layered graph model in [3.24]. The authors elaborated that although networks are deliberated as distinctive objects, these objects are usually fragments of complex network, where various topologies interdependently interact with each other. The authors defined two layers of the network: 1) physical and 2) logical. The physical graph represents the lower layer topology, and the logical graph represents the upper layer topology. Each logical edge exhibits mapping on the physical graph as a path. Because the number of layers is fixed, the proposed model is inapplicable to the DCN architectures. Moreover, none of the layers in DCNs are logical. Therefore, the idea of mapping one layer to the other is incapable to characterize DCNs.

Dong et al. in [3.25] defined a multilayered graph G composed of M layers. Each layer represents an undirected weighted graph, composed of a set of common vertices υ and edges ε. having associated weights. As the number of nodes (vertices) in each layer needs to be same, the proposal is inapplicable to DCNs. Moreover, the definition lacks the interconnection information between different layers in the proposal. Because none of the previously proposed graph models matches the DCN-based graph definition, we present a formal definition for each of the DCN architectures.

Table 3.1. Definition of the Variables Used in the DCN Models.

ν	set of vertices
ε	Set of edges
P_i	a *pod* in the topology that is composed of servers and middle-layer switches
k	number of pods/modules in the topology
n	number of nodes connected to a single access layer switch
m	number of access layer switches in each pod P_i
q	number of aggregate layer switches in each pod P_i
r	number of core switches
δ	servers
α	access layer switch
γ	aggregate layer switch
C	core layer switch

3.2.2. ThreeTier DCN Architecture

We define the ThreeTier architecture according to the definitions in Table 3.1 as

$$DCN_{TT} = (\nu, \varepsilon\,). \tag{3.1}$$

Here ν are the vertices and ε represents the edges. Vertices are arranged in k pods P_i^k (servers, access switches, and aggregate switches), and a single layer of r core C_i^r switches:

$$\nu = \{P_i^k \cup \mathsf{C}_i^r\}, \tag{3.2}$$

where C_i^r is a set composed of all of the core switches:

$$\mathsf{C}_i^r = \{c_{1\,}, c_{2\,}, \ldots, c_{r\,}\}. \tag{3.3}$$

Each P_i is composed of three layers of nodes, namely: 1) servers layer (l^s), 2) access layer (l^a), and 3) aggregate layer (l^g). Nodes in each of the pods can be represented as

$$P_i = \left\{ l^1_{m.\alpha \times n.\delta} \, , \, l^2_{m.\alpha} \, , \, l^3_{q.\gamma} \right\}, \tag{3.4}$$

where α represents access layer switches, γ represents aggregate layer switch, and δ represents servers. Total number of nodes in each of the pods can be calculated as

$$|P_i| = \left(\sum_{1}^{m} n \; + m + q \right), \tag{3.5}$$

where $|P_i|$ stands for the cardinality of the set of nodes in each pod. The total number of vertices of a topology can be calculated as

$$|\nu| = \left\{ \sum_{i=1}^{k} |P_i| + |\mathrm{C}| \right\}. \tag{3.6}$$

There are generally three layers of edges ε = {§ ,ά ,₡ }, where 1) § are the edges that connect servers to the access layer, 2) ά edges connect the access layer to the aggregate layer, and 3) ₡ edges connect the aggregate layer to the core layer. Beside the aforementioned, the ThreeTier topology also has a set of edges connecting the aggregate layer switches to each other within the pod, represented by

$$\boldsymbol{\varepsilon} = \left\{ \S_{(\forall \boldsymbol{\delta}, \boldsymbol{\alpha})} \, , \acute{\alpha}_{(\forall \boldsymbol{\alpha}, \forall \boldsymbol{\gamma})}, \curlyvee_{(\forall \boldsymbol{\gamma}, \forall \boldsymbol{\gamma})}, ₡_{(\forall \boldsymbol{\gamma} \, \forall \mathrm{C})} \right\}, \tag{3.7}$$

Therefore, the set of edge of the ThreeTier DCN can be represented by

$$\boldsymbol{\varepsilon} = \left\{ \S_{(\forall \boldsymbol{\delta}, \boldsymbol{\alpha})} \, , \acute{\alpha}_{(\forall \boldsymbol{\alpha}, \forall \boldsymbol{\gamma})}, \curlyvee_{(\forall \boldsymbol{\gamma}, \forall \boldsymbol{\gamma})}, ₡_{(\forall \boldsymbol{\gamma} \, \forall \mathrm{C})} \right\}, \tag{3.8}$$

where $\S_{(\forall \boldsymbol{\delta}, \boldsymbol{\alpha})}$, are the edges connecting each server to a single- access layer switch, , $\acute{\alpha}_{(\forall \boldsymbol{\alpha}, \forall \boldsymbol{\gamma})}$ connecting each access layer switch to all aggregate layer switches, $\curlyvee_{(\forall \boldsymbol{\gamma}, \forall \boldsymbol{\gamma})}$, connecting each aggregate layer switch to all other aggregate layer switches within the pod, and

, ¢$_{(\forall \gamma \, \forall C)}$connecting each aggregate layer switch to all of the core layer switches. The total edges in the topology can be calculated as

$$|\varepsilon| = k\left(mn + mq + \frac{q\,(q-1)}{2} + qr\right). \tag{3.9}$$

3.2.3. FatTree DCN Architecture

Similar to the ThreeTier DCN, the FatTree architecture is also composed of a single layer of computational servers and three layers of network switches arranged in k pods. However, the FatTree architecture follows a Clos-based topology [3.26], and the number of networking elements and interconnecting edges in the FatTree architecture are much higher than the ThreeTier architecture. We will use the conventions defined in Table 3.1 for the graph modeling of the FatTree architecture. The number of elements in each layer within each P_i is fixed based on the k. The number of vertices v in n, m, q, and r can be calculated as

$$n = m = q = \left(k/2\right), \tag{3.10}$$

$$r = \left(k/2\right)^2. \tag{3.11}$$

The FatTree DCN can be modeled similar to that of the ThreeTier architecture as

$$DCN_{FT} = (v, \varepsilon), \tag{3.12}$$

where as $v, C_i^r, P_i, |P_i|$, and $|v|$ can be modeled by using (3.2) to (3.6), respectively. Contrary to the ThreeTier architecture, the aggregate layer switches in the FatTree architecture are not connected to each other. Moreover, every core layer switch C_i^r is connected only to a single-aggregate layer switch, which allow us to state:

$$\varepsilon = \{\S_{(\forall\delta,\alpha)} \cup \acute{\alpha}_{(\forall\alpha,\forall\gamma)} \cup \text{¢}_{(\forall\text{C},\gamma_i)}\}, \tag{3.13}$$

and the total number of edges in the FatTree topology can be calculated as

$$|\varepsilon| = k\,(mn + mq + +qr) + kr. \tag{3.14}$$

3.2.4. DCell Architecture

In contrast with the ThreeTier and FatTree DCN architectures, the DCell uses server-based routing architecture. Every $dcell_0$within the DCell holds a switch to connect all of the computational servers within the $dcell_0$. The DCell uses a recursively built hierarchy, and $dcell_l$ is built of x_i $dcells_{l-1}$. The algorithm for the interconnections among the servers in various $dcells$ can be seen in [3.18]. The graph model of the DCell architecture can be represented as:

$$DCN_{DC} = (\nu, \varepsilon), \tag{3.15}$$

$$\nu = \{\partial_i, \partial_{i+1}, \dots, \partial_n\}, \tag{3.16}$$

where $0 \leq i \leq l$, and ∂_0 represents the $dcell_0$:

$$\partial_0 = \delta \cup \alpha, \tag{3.17}$$

where δ represents the set of servers within $dcell_0$, s is the number of servers within $dcell_0$, and α is the network switch connecting s servers within $dcell_0$.

$$\partial_l = \{x_l.\,\partial_{l-1}\}, \tag{3.18}$$

where x_l is total number of ∂_{l-1} in ∂_l.

$$\partial_1 = \{x_1\partial_0\}, \tag{3.19}$$

$$\partial_1 = \{x_1\partial_0\}, \tag{3.20}$$

$$x_1 = s + 1. \tag{3.21}$$

Similarly, for $l \geq 2$:

$$x_l = \left(\prod_{i=1}^{l-1} x_i \times s \right) + 1. \tag{3.22}$$

The DCell DCN is a highly scalable architecture and supports any level of $dcells$. However, a 3-level DCell is sufficient to accommodate millions of servers. The total number of nodes in a 3-level DCell can be computed as

$$|v_0^3| = \sum_{1}^{x_3} \sum_{1}^{x_2} \sum_{1}^{x_1} (s+1), \tag{3.23}$$

and the total number of edges in a 3-level DCell are:

$$|\varepsilon_0^3| = \sum_{1}^{x_3} \left(\sum_{1}^{x_2} \left(\left(\sum_{1}^{x_1} \mathrm{s} \right) + (x_1(x_1-1)/2) \right) + (x_2(x_2-1)/2) \right) + (x_3(x_3 - 1)/2) . \tag{3.24}$$

The total number of nodes in the $l - level$ DCell, ∂_l , can be computed as

$$|v| = \left(\prod_{i=1}^{l} \left(\sum_{1}^{x_i} (s+1) \right) \Big/ (s+1)^{(l-1)} \right), \tag{3.25}$$

and he total number of edges in the $l - level$ DCell, ∂_l , can be computed as

$$|\varepsilon| = \left(\prod_{i=1}^{l} \left(\sum_{1}^{x_i} (s) \right) \Big/ (s)^{(l-1)} \right) + \frac{1}{2} \left(\sum_{j=1}^{l} \left(\prod_{k=j}^{l} x_j - 1 \right) \right). \tag{3.26}$$

3.3. Robustness Metrics

3.3.1. Background

This section briefly presents some of the well-known graph robustness metrics. Some of the metrics classified here (see Table 3.2) as classical are based on the concepts of the graph theory, while the contemporary metrics consider the services supported by the networks. In this paper, we consider the classical robustness metrics, leaving the dynamic aspects of the DCN robustness as future work. A brief description of the robustness metrics is presented in the following section.

3.3.2. Robustness Metrics Glossary

Assortativity coefficient (r): presents the tendency of a node to connect to other nodes having dissimilar degrees [3.13]. The value of r lies within the range $-1 \leq r \leq 1$. The value of $r < 0$ represents dissassortative network, having excess of links among nodes of dissimilar degrees.

Average neighbor connectivity ($\frac{k_{nn}}{|v|-1}$): delivers information about one-hop neighborhood of a node [3.9]. The value of $\frac{k_{nn}}{|v|-1}$ delivers joint degree distribution statistics, and is calculated as average neighbor degree of the average k-degree nodes.

Average nodal degree ($\langle k \rangle$): is one of the coarse robustness measures [3.13]. Networks having high $\langle k \rangle$ values are considered more robust and "better-connected" on average.

Average shortest path length ($\langle l \rangle$): is the average of all of the shortest paths among all of the node-pairs of the network [3.12]. Small $\langle l \rangle$ values exhibit better robustness, because such

networks are likely to lose fewer connections in response to different types of failures (random or targeted).

Table 3.2. Classical and Contemporary Robustness Metrics.

	Characteristic	**Reference**
Classical	Average nodal degree ($\langle k \rangle$)	[3.13]
	Node connectivity (k)	[3.9]
	Link connectivity (ρ)	[3.9]
	Heterogeneity($\sqrt{\sigma_k^2}/\langle k \rangle$)	[3.10]
	Symmetry ratio ($\epsilon/(D+1)$)	[3.11]
	Diameter (D)	[3.32]
	Average shortest-path length ($\langle l \rangle$)	[3.12]
	Assortativity coefficient (r)	[3.13]
	Average neighbor connectivity ($\frac{k_{nn}}{\lvert v \rvert - 1}$)	[3.13]
	Clustering coefficient ($\langle C \rangle$)	[3.13], [3.29]
	Betweenness centrality ($\langle b \rangle$)	[3.28]
	Largest eigenvalue or spectral radius (λ_i)	[3.13], [3.14]
	Second smallest Laplacian eigenvalue or algebraic connectivity ($\mu_{\lvert v \rvert - 1}$)	[3.15]
	Average two-Terminal Reliability (A2TR)	[3.27]
Contemporary	Elasticity (E)	[3.8]
	Quantitative Robustness Metric (QNRM)	[3.7]
	Qualitative Robustness Metric (QLRM)	[3.7]
	R-value (R)	[3.30]
	Viral Conductance (VC)	[3.31]

Average two-terminal reliability ($A2TR$*)*: delivers the probability of the connectivity between a randomly chosen node pair [3.27]. In a fully connected network, $A2TR$ value is one. Otherwise, $A2TR$ is the sum of total number of node pairs in each connected cluster divided by all of the node pairs in the network.

Betweenness centrality ($\langle b \rangle$): measures the number of shortest paths among nodes that pass through a node or link. Betweenness centrality is used to estimate the prestige of node/link [3.28].

Clustering coefficient ($\langle C \rangle$): is the percentage of 3-cycles among all of the connected node triplets within the network [3.13], [3.29]. If two neighbors of a node are connected, then a triangle (3-cycle) is formed by these three nodes.

Diameter (D): is the longest path among all of the shortest paths of the network. Generally, low D represents higher robustness.

Elasticity (E): relates to the total throughput in response to the node removal [3.8]. The fundamental idea is to successively remove a certain fixed number of nodes r (in the original definition, $r < 1\%$) and measure the consequent throughput degradation. The more pronounced and abrupt is the throughput drop experienced by a given topology, the lower is the robustness.

Heterogeneity: is the standard deviation of the average node degree divided by the average node degree [3.10]. The lower heterogeneity value translates to higher network robustness.

Largest eigenvalue or spectral radius (λ_1): is the largest eigenvalue of the adjacency matrix of a network [3.13], [3.14]. Generally, the networks with the higher eigenvalues have small diameter and higher node distinct paths.

Node connectivity (k): represents the smallest number of nodes whose removal results in a disconnected graph [3.9]. The node connectivity is the least number of node-disjoint paths between any two nodes within the network, which provides a rough indication of network robustness in response to any kind of failures or attacks (random or targeted). The same definition can be applied to link connectivity, when considering links instead of nodes.

Quantitative Robustness Metric ($QNRM$): analyzes how multiple failures affect the number of connections established in a network [3.7]. The $QNRM$ delivers the number of the blocked connections (that cannot be established because of failure).

Qualitative Robustness Metric ($QLRM$): analyzes the variation in the quality of service of a network under various types of failures [3.7]. The $QLRM$ measures the variation of the average shortest path length of the established connections.

R-value (R): computes the robustness of a topology under one or more topological features [3.30]. The obtained value is normalized to [3.0, 1].

Second smallest Laplacian eigenvalue or algebraic connectivity (μ_{v-1}): depicts how difficult it is to break the network into islands or individual components [3.15]. The higher the value of μ_{v-1}, the better the robustness.

Symmetry ratio ($\frac{\epsilon}{D-1}$): is the quotient between the distinct eigenvalues of the network adjacency matrix and the network diameter [3.11]. The networks with low symmetry ratio are considered more robust to random failures or targeted attacks.

Viral Conductance (VC): measures the network robustness in case of epidemic scenarios (propagation/spreading of failures) [3.31]. The VC is measured by considering the area under the curve that provides the fraction of infected nodes in steady-state for a range of epidemic intensities.

3.4. Simulation Scenarios and Methodologies

This section details the simulation scenarios and methodologies used in this work. To generalize the robustness analysis of the state-of-the-art DCNs, we performed extensive simulations considering four node failure scenarios to measure the various robustness metrics, namely:

1. random failures,
2. targeted failures,
3. network-only (failures introduced only in the net- work devices), and
4. real DCN failures (using real DCN failure data collected over a period of one year).

To do so, we consider six DCN networks, which are presented in Section 6. For the first three failure scenarios, we analyzed the robustness of each DCN by introducing the failures within a range from 0.1 to 10 percent of the network size. With the purpose of providing a detailed robustness evaluation, we analyzed the robustness metrics by introducing 0.1 to 2.5

percent of failures with an increment of 0.1, whereas from 3 to 10 percent the increment was equal to 1.

In the real DCN failures case, we used the observations reported in [3.33]. Gill et al. analyzed the network failure logs collected over a period of around one year from tens of data centers. The authors derived the failure probability for various network components by dividing the number of failures observed in a specific network device type, such as access layer or aggregate layer switches, with the total population of the devices in the given device type. We used the frequentist probability to derive the number of failures in three DCN architectures. We analyzed the various robustness metrics under real failure scenario by instigating the derived failures at each layer. As the number of network elements in the FatTree is much higher than the ThreeTier architecture, the number of failed nodes is around five times in the FatTree as compared to the ThreeTier architecture.

We introduced random failures in data center nodes (including the computational servers) within a range of 0.1 to 10 percent of the network size, as discussed in the various studies, such as [3.7], [3.34], [3.35]. The node failures are distributed among the nodes at each layer and dcell level within a range of 31-3,266 nodes. Besides instigating failures randomly in the whole network, we also considered the scenario of the network-only node failure, as discussed in [3.33]. Another significant scenario to measure the system robustness is by introducing the targeted attacks [3.35], [3.36], [3.37]. In the targeted failures case, we considered the betweenness centrality of the nodes to introduce the node failures.

3.5. Network Topologies

In this section, we present six representative topologies of the DCN architectures. Moreover, robustness is discussed according to the characteristics of each of the DCN architectures. The selected topologies represent connected and symmetric DCN networks.

DCN architectures follow a complex interconnection topology that entails a detailed understanding of the architecture to generate the DCN topology. Therefore, generating the representative DCN synthetic topologies is a difficult task. There is presumably no publically available DCN topology generation tool. We developed a DCN topology generator for custom and flexible creation of various DCN topologies. Based on various input para- meters, such as number of pods for the FatTree, dcell levels, and number of nodes in dcell0 for the DCell, and number of nodes and switches in various layers in the ThreeTier DCN architecture, the DCN topology generator engenders the network topology in various popular graph formats. We generated two representative network topologies for each of the DCN architectures:

- three large networks (DCell30K, FatTree30K, and ThreeTier30K),
- three smaller networks (DCell2K, FatTree2K, and ThreeTier2K).

Increasing a single server in the DCell topology exponentially expands the network. A 3-level DCell with two servers in $dcell_0$ constitute a network of 2,709 nodes. An increase in the number of servers to three in $dcell_0$ results in a network of 32,656 nodes. Therefore, the considered topologies are 2K and 30K networks.

Table 3.3 depicts some of the features of the three large networks. As observed, all of the topologies have more than 30,000 nodes. The FatTree30K has the largest number of edges among the considered set of large networks. The density of the FatTree30K is around three times

higher than the ThreeTier30K. The higher number of edges and density exhibit better resilience to failures. The value of the average shortest path length ⟨l⟩ for the FatTree30K and ThreeTier30K is less than six, whereas the DCell30K has a higher path length (11). A higher ⟨l⟩ means that the communication between the end hosts in the DCell30K is more susceptible to be affected by a failure than in the FatTree30K or ThreeTier30K. This is due to the fact that such a communication is going to be routed (in average) through a longer path. The higher the number of links and nodes involved in a path, the higher is the probability to be affected by failures. Similarly, the DCell30K diameter D presents a value four times higher than the FatTree30K and ThreeTier30K. However, the DCell30K possesses high-average nodal degree that depicts strong resilience against failures. Moreover, all of the three networks exhibit dissassortativity and have negative value of the assortativity coefficient. It means that all of the three networks have an excess of links among nodes with dissimilar degrees.

Tables 3.4 and Table 3.5 present features of the DCell2K, FatTree2K, and ThreeTier2K topologies. Each topology is composed of around 2,500 to 2,700 nodes. As observed previously in the 30K networks, the FatTree DCN architecture has the largest number of edges. Regarding the spectral radius and algebraic connectivity μ_{v-1}, the FatTree2K proves to be the most robust network. The higher the value of λ_1 and μ_{v-1}, the higher the robustness. Although the ThreeTier2K also indicates better robustness when considering λ_1, the ThreeTier2K possess the highest maximum nodal degree k_{max}. High k_{max} is an indicator of vulnerability, depicting that removal of such a node could seriously damage the network. Moreover, the minimum values of the node and link connectivity for all of the networks are $= 1$ and $\rho = 1$, respectively. Such values indicate that a single node or link failure may cause the network fragmentation. Because of having the lowest symmetry ratio value, the DCell2K exhibits a higher robustness.

Table 3.3. 30K DCN Topology Features.

Topology	$\|v\|$	$\|\varepsilon\|$	$\langle k \rangle$	$\langle l \rangle$	$\langle d \rangle$	r	$\frac{2 \cdot \|\epsilon\|}{(\|v\| \cdot (\|v\| - 1}$
DCell30K	32656	61230	3.7500	11.1521	23	-0.25	0.00011
FatTree30K	30528	82994	5.4336	5.6200	6	-0.20	0.00017
Three Tier30K	30676	31632	2.0620	5.9000	6	-0.95	0.00006

Table 3.4. 2K DCN Topology Features.

Topology	$\|v\|$	$\|\varepsilon\|$	$\langle k \rangle \mp$StDev	λ_1	k_{ma}	$\mu_{\|v\|-1}$	$\frac{k_{nn}}{\|v\|-1} \mp$StDev	k	ρ	$\frac{\epsilon}{D+1}$
DCell2K	2709	4515	3.3333∓0.94	3.56155	4	0.12439	0.00066	1	1	169.3125
FatTree2K	2500	6000	4.8000∓7.60	17.4186	20	0.31528	0.00337∓0.003	1	1	318.2857
ThreeTier2K	256	274	2.1389∓4.64	10.2504	40	0.0230	0.00119∓0.001	1	1	318.714

It can also be observed that the FatTree2K and ThreeTier2K have a low average shortest path length $\langle l \rangle$ than the DCell2K, and consequently can be considered more robust with respect to $\langle l \rangle$. The average node betweenness centrality $\langle b \rangle$ depicts that although the DCell2K has the highest value of $\langle b \rangle$, the DCell2K exhibits least standard deviation in the individual node's $\langle b \rangle$ value. Therefore, it can be inferred that all of the nodes of the DCell2K have nearly similar value of the betweenness centrality. Alternatively, the value of $\langle b \rangle$ for the FatTree2K and ThreeTier2K is lower than the DCell2K, but they have higher standard deviation, which means that the

FatTree2K and ThreeTier2K networks have an excess of centrality measures for some nodes, indicating the vulnerability of networks under targeted failures. The node betweenness centrality distribution for the 600 highest values in the three networks is shown in Fig. 3.1. The DCell2K curve illustrates uniformly distributed values of $\langle b \rangle$ for all of the nodes.

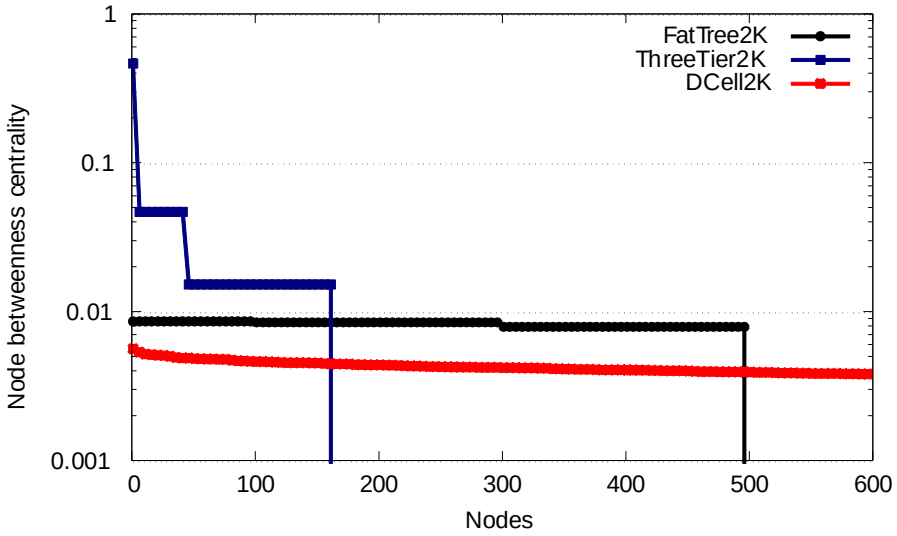


Fig. 3.1. Node Betweenness Centrality Distribution in Logarithmic Scale of 600 Nodes with the Highest Value of DCell2K, FatTree2K, and ThreeTier2K.

Table 3.5. 2K DCN Topology Features.

Topology	$\langle l \rangle \mp$StDev	$\langle b \rangle \mp$StDev	$\langle C \rangle \mp$StDev	r	D	$\frac{2 \cdot \lvert\varepsilon\rvert}{(\lvert v\rvert \cdot (\lvert v\rvert - 1))}$	$\frac{\sqrt{\sigma_k^2}}{\langle k \rangle} \mp$StDev
DCell2K	8.51062 ± 1.93990	0.00277 ± 0.00133	0±0	-0.25	15	0.00123	0.28284
FatTree2K	5.21063 ± 1.12342	0.00169 ± 0.00337	0.8000± 0.4000	-0.2	6	0.00192	1.58333
ThreeTier2K	5.72473 ± 0.70278	0.00185 ± 0.01482	0.9404 ± 0.2303	-0.8961	6	0.00083	2.17007

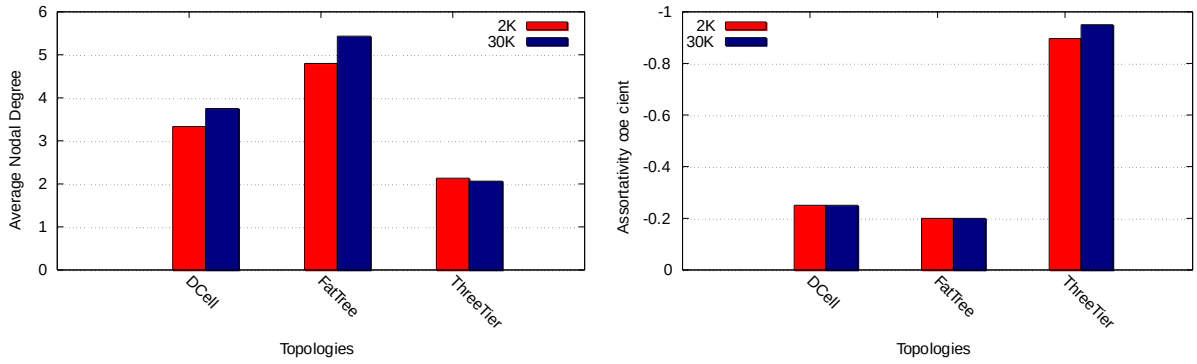


Fig. 3.2. Average Nodal Degree (Left) and Assortativity Coefficient (Right) Comparison of the 30K and 2K Networks.

Table 3.6. Robustness Classification of the Three DCN Architectures.

Metrics	FatTree	DCell	ThreeTier
$\lvert\varepsilon\rvert$	Highest	Average	Least
$\langle k \rangle$	Highest	Average	Least
$\langle l \rangle$	Highest	Least	Average
D	Highest	Least	Average
r	Highest	Average	Least
$\frac{2 \cdot \lvert\varepsilon\rvert}{(\lvert v\rvert \cdot (\lvert v\rvert - 1))}$	Highest	Average	Least
λ_1	Highest	Least	Average
k_{max}	Average	Highest	Least
$\mu_{\lvert v\rvert} - 1$	Highest	Average	Least
$\frac{\epsilon}{D+1}$	Average	Highest	Least
$\langle b \rangle$	Highest	Least	Average
$\langle C \rangle$	Average	Least	Highest
$\frac{\sqrt{\sigma_k^2}}{\langle k \rangle}$	Average	Highest	Least

The absence of 3-cycles in the clustering coefficient $\langle C \rangle$ measurements reveal that the DCell2K lacks two-hop paths to re-route the traffic in case of failure of one of its neighbors. On the contrary, the FatTree2K and Three- Tier2K exhibit better robustness by having high values of $\langle C \rangle$, which illustrate the existence of multiple alternative two-hop paths. Moreover, all of the three networks are dissassortative, $r < 0$. The density measurements show that the FatTree2K is the most dense and henceforth the most robust network. The low heterogeneity value shows that the DCell2K can be considered as the most robust network when considering the heterogeneity.

The initial network analysis (for the whole network without failures) of the considered DCN topologies reveals that none of the three networks can be considered as the most robust architecture for all of the metrics. The robustness classification of the DCN networks for various metrics is reported in Table 3.6. The highest, average, and least values in the Table 3.6 depict the robustness level of the network. It can be observed that the FatTree architecture exhibits highest robustness for most of the metrics. Therefore, based on the initial network analysis without failures, it can be stated that the FatTree DCN exhibits better robustness than the DCell and ThreeTier architectures.

3.6. Results

This section presents a detailed analysis of the structural robustness of the DCN networks presented in Section 6. Initially, a comparison of the 30K with the 2K networks (6.1) is presented. Thereafter, the robustness analysis of the: 1) 30K networks (6.2) and 2) 2K networks (6.3) considering various failure scenarios is discussed. Although the study has been carried out within the range of 0.1 to 10 percent of the nodes affected by the failures, the results present a

maximum of 6 percent of the affected nodes. This is due to the fact that the higher percentages in the targeted and network-only failures completely disconnect some of the networks. Therefore, the considered graph metrics do not deliver any useful information for higher failure percentages.

Several graph metrics are computational intensive and require a large amount of CPU time. Therefore, the large (30K) networks are analyzed by their: 1. largest connected component, 2. average nodal degree, 3. node connectivity, and 4. number of clusters. Whereas, the small networks are studied considering their: 1. algebraic connectivity, and 2. spectral radius or largest eigenvalue. It is noteworthy to consider that some of the metrics are applicable only to the largest connected component, as they require connected graph. Therefore, the result values of μ_{v-1} , $\langle b \rangle$, and λ_1 are dependent on the largest connected component of the network.

3.6.1. Network Size Comparison

The degree distribution of nodes in various DCNs exhibit homogeneous pattern, and the degree of each node is one among the few values in the degree set. For example, there are only two types of nodes (switches and servers) in the DCell. Therefore, the degree distribution follows two values: 1) having a similar value for all of the switches and 2) for all of the servers. Similarly, in case of the FatTree architecture, each node's degree is either one (for servers) or the k (for switches). In the ThreeTier architecture, the nodal degree falls within one of the four values, one each for the servers, access layer switches, aggregate layer switches, and core layer switches. The average nodal degree $\langle k \rangle$ of the DCNs does not strictly depend on the network size. A comparison of $\langle k \rangle$ for the 2K and 30K networks is illustrated in Fig. 3.2a. It can be observed that there is no significant difference in the values of $\langle k \rangle$ of the large and small DCN networks. Similarly, regarding the assortativity coefficient values for the large and small

networks (see Fig. 3.2b), it can be observed that there is no remarkable difference between the assortativity coefficients of the large and small DCNs, and all of them remain dissassortative.

In essence, increasing the DCN size does not imply obtaining very different network topology characteristics. Therefore, we divide the robustness metrics analysis into two parts: 1) the low CPU time consuming metrics are analyzed for the 30K network set and 2) the high CPU time consuming metrics are studied for the 2K network set.

3.6.2. 30K Networks

Component structure of a network is one of the most important properties to be considered. Therefore, largest connected component is considered significant to measure the effectiveness of the network [3.38], [3.39], [3.40], [3.41], [3.42]. In case of random failures, all of the considered DCN architectures exhibit a robust behavior. As observed in Fig. 3.3a, the largest component size remains above 85 percent, even when 6 percent of the nodes fail randomly. However, in case of the targeted attack, the ThreeTier and FatTree topology behave contrary (see Fig. 3.3b). Removal of a very small fraction (< 0.1 percent) of the nodes in the ThreeTier architecture results in segregation of the network. The network completely disconnects when around 2.5 percent of the nodes fail. This is due to the fact that the ThreeTier architecture have certain nodes (core and aggregate layer switches) with very high betweenness centrality values. Therefore, failure of such nodes segregate the network. However, the FatTree shows an altered behavior. Around 98 percent of the nodes reside in the connected component until a targeted failure of 1.8 percent of the nodes. An abrupt change is observed when the failure rate reaches 1.9 percent, resulting in the decline of largest component size from 98 to 2 percent, depicting a phase change. Therefore, the point at which 1.9 percent of the nodes fail can be

considered as the critical point [3.43] for the FatTree architecture. Alternatively, the DCell confirms the resilience to the targeted attack. A smooth linear decline is observed in the largest component size decay. Around 94 percent of the network nodes reside in the largest component when 6 percent of the nodes fail in the DCell. The detailed values of the largest cluster size for the three networks can be found in Table 3.7.

Table 3.7. Largest Connected Component Size of the 30K Networks.

Percentage	Random			Targeted		
	FT30K	TT30K	DC30K	FT30K	TT30K	DC30K
0	1	1	1	1	1	1
0.1	0.997491	0.997268	0.998989	0.998985	0.020831	0.998928
0.5	0.989269	0.990452	0.994978	0.994988	0.001597	0.994488
0.9	0.983035	0.984007	0.990997	0.990992	0.001597	0.990323
1.1	0.978403	0.979955	0.988976	0.988994	0.001597	0.98818
1.5	0.971452	0.971903	0.984995	0.984997	0.001597	0.984015
1.9	0.966457	0.96304	0.980981	0.02044	0.001597	0.979912
2.3	0.956073	0.953742	0.976969	0.02044	0.001597	0.975778
3	0.942934	0.938871	0.969984	0.018802	0.000033	0.96803
4	0.927083	0.917294	0.959952	0.013889	0.000033	0.957343
5	0.906394	0.906516	0.949954	0.006519	0.000033	0.946564
6	0.891349	0.881696	0.939928	0.000033	0.000033	0.935755

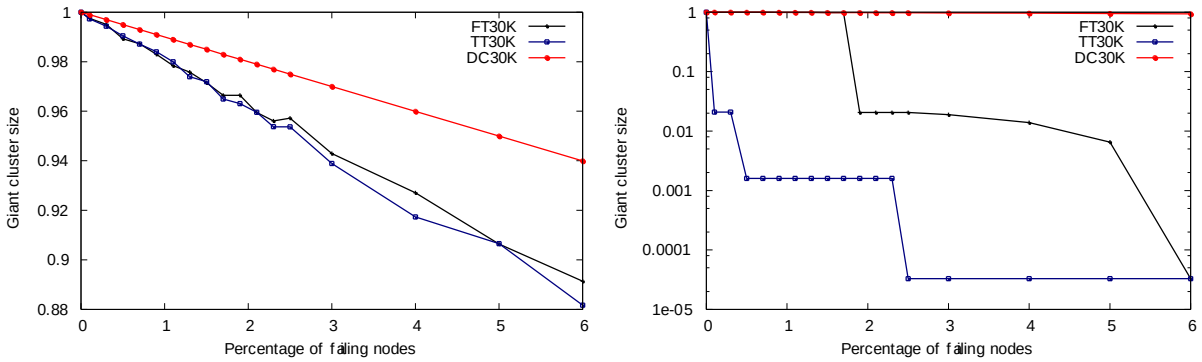


Fig. 3.3. Largest Connected Component Size Analysis under Random and Targeted Failures of the 30K Networks.

Table 3.8. Average Nodal Degree ($\langle k \rangle$) of the 30K Networks.

Percentage		Random			Targeted		Network-only failure		
	FT30K	*TT30K*	*DC30K*	*FT30K*	*TT30K*	*DC30K*	*FT30K*	*TT30K*	*DC30K*
0	5.433	2.062	3.75	5.433	2.062	3.75	5.4336	2.062	3.75
0.1	5.4257	2.05887	3.7462	5.3419	2.014815	3.74643	5.34226	1.971878	3.74628
0.5	5.4015	2.05268	3.7312	4.9777	1.792805	3.73318	4.98835	1.612142	3.73128
0.9	5.3808	2.04681	3.7165	4.6107	1.411625	3.71969	4.64439	1.258456	3.71658
1.3	5.3590	2.03783	3.7012	4.2406	1.030485	3.70683	4.30927	0.911840	3.70126
1.7	5.3394	2.02696	3.6864	3.8676	0.643098	3.693558	3.98853	0.562512	3.686467
2.1	5.31762	2.02049	3.67106	3.48845	0.252539	3.681867	3.67092	0.219879	3.671067
2.5	5.31431	2.01617	3.6563	3.10926	0	3.670571	3.37251	0	3.656359
3	5.27029	1.99950	3.6372	2.63244	0	3.65664	3.01409 6	0	3.637258
4	5.22503	1.97237	3.5999	1.65754	0	3.627364	2.33342	0	3.59993

The average nodal degrees of the largest connected component are presented in Table 3.8 and Fig. 3.4 for the random, targeted, and network-only failures. It can be observed that in case of random failures, where the failure is introduced both in the network and computational nodes,

the three DCNs behave similarly. However, when the failures are introduced in the network portion or in case of the targeted attack, the FatTree and ThreeTier exhibit a rapid decline in the nodal degree. The reason for such a rapid decay is the fact that the failure of a single access layer switch disconnects n nodes. Therefore, the average nodal degree decays rapidly. The failure analysis depicts that the DCell30K exhibits robustness in terms of the average nodal degree under all of the failures.

The node connectivity is an important measure to illustrate how many nodes need to fail to disconnect the network. The node connectivity can be measured by calculating the minimum node distinct paths between any two nodes within the network. As there is only a single edge that connects the node to the access switch, the node disjoint paths for the ThreeTier and FatTree DCNs are always one. However, from every access layer switch to every other switch, there are always, $k/2$ node distinct paths in the FatTree. Similarly, in the ThreeTier network, the maximum node distinct paths between any two access layer and the aggregate layer devices are equal to q and r, respectively. Because there is only a single edge between the network switch and servers within $dcell_0$, the minimum node disjoint paths of the network is one. However, the network switch only performs the packet forwarding within $dcell_0$ and the actual communication always occurs among the servers in the DCell architecture. The node distinct paths between any two servers in the DCell are equal to $l + 1$, where l is the DCell level.

The number of the segregated clusters are depicted in Fig. 3.5 and the detailed values are presented in Table 3.9. It can be observed that the FatTree30K and ThreeTier30K behave similarly in random failures. The failure of a single access layer switch disconnects n servers, resulting in n segregated clusters. Therefore, the FatTree30K and ThreeTier30K networks

disconnect into more than 45 clusters when 0.1 percent of the nodes fail. However, for the network-only failure case, the ThreeTier30K disconnects into more clusters than the FatTree30K. In the targeted failures case, the robustness difference is even more, where the FatTree30K remains connected until 1.9 percent of the nodes fail. This is due to the fact that the FatTree network has a considerable portion of the nodes ($(k/2)^2$ core switches) with similar high betweenness centrality values (see the betweenness centrality distribution in Section 6). Therefore, the topology remains fully connected until ($(k/2)^2$ - 1) nodes fail in the FatTree30K. It is noteworthy to mention that because of nearly similar betweenness centrality distribution among the nodes, the DCell30K outperforms the other two 30K networks. The DCell portrays high robustness in random or network-only failures, and remains connected until 4 percent of the nodes are affected. However, the network disconnects with only 0.1 percent of the nodes failure in case of the targeted attack. Nonetheless, the number of segregated cluster remains much less than the counterparts. In any of the failure cases, the DCell30K can be considered as the most robust network in terms of number of isolated clusters among all of the DCN architectures.

Table 3.9. Number of Clusters of the 30K Networks.

Percentage		Random			Targeted		Network-only failure		
	FT30K	TT30K	DC30K	FT30K	TT30K	DC30K	FT30K	TT30K	DC30K
0	1	1	1	1	1	1	1	1	1
0.1	46.6	53.8	1	1	72	3	308.2	1282.6	1
0.5	175.6	139.9	1	1	3162	19	1491.4	6405.7	1
0.9	243.9	214.6	1	1	8943	26	2540.2	11510.6	1
1.3	344	338.7	1.1	1	14677	32	3874.6	16580.6	1.1
1.7	507.9	492.1	1.1	1	20458	36	4865.8	21692.7	1.1
2.1	596.3	595	1.1	1632	26239	44	6269.8	26730.3	1.1
2.5	543.6	655.7	1.2	4560	29909	52	7165	30676	1.2
3	827.1	898.7	1.2	8208	29755	70	8626.6	30676	1.2
4	1005	1255.2	1.8	15552	29448	93	11792.2	30676	1.8
5	1331.6	1334.7	2.3	22872	29142	120	14504.2	30676	2.3

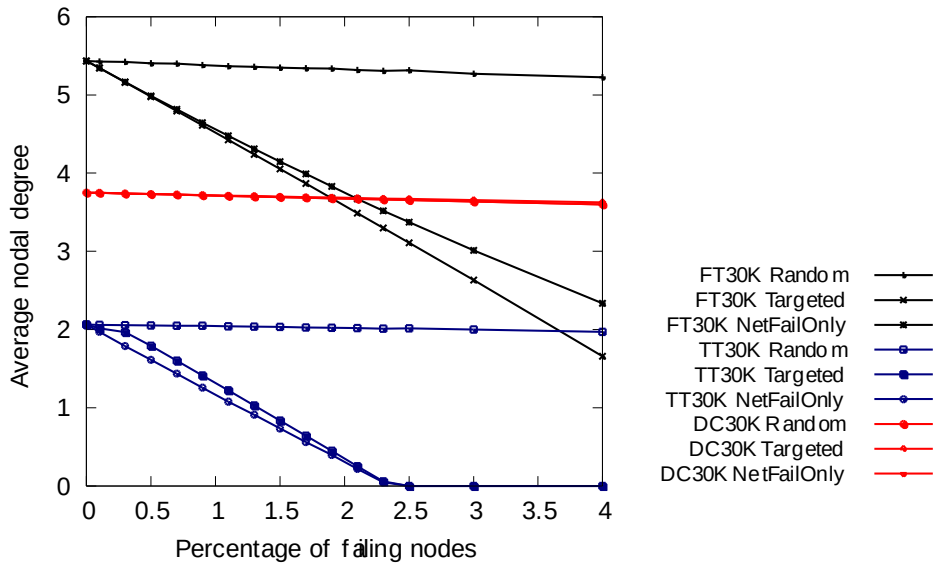


Fig. 3.4. Average Nodal Degree Analysis for 30K Networks.

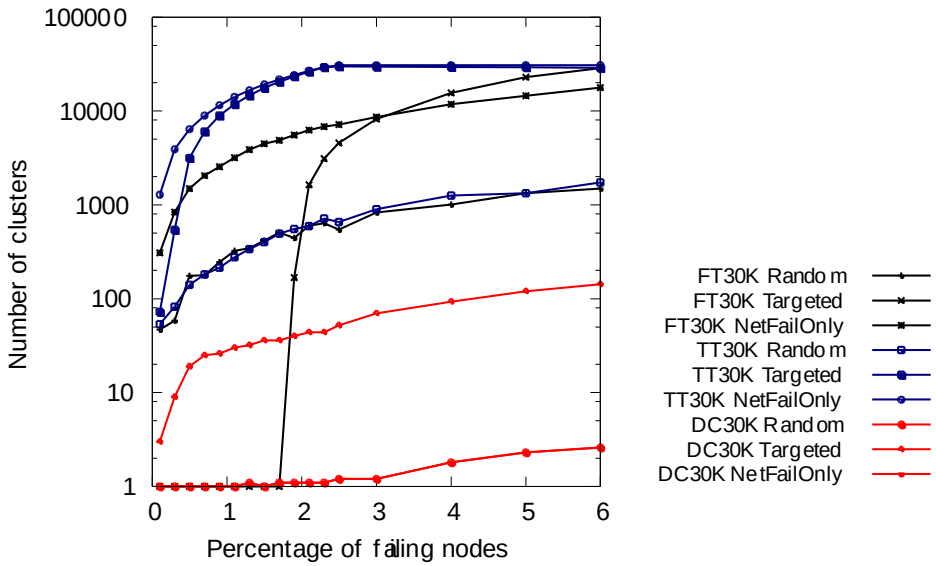


Fig. 3.5. Number of Clusters Analysis for the 30K Networks.

3.6.3. 2K Networks

The robustness evaluation of the large (30k) networks is infeasible when considering the computational intensive metrics, such as μ_{v-1}, ⟨b⟩, and λ_1. Therefore, we evaluate small (2k) networks for the computational intensive metrics. One of the most significant considerations in the evaluation of the 2k networks for the computational intensive metrics is that such metrics only consider the largest connected component of the network. In case of the targeted and network-only failures, the size of the largest connected component is typically very small, and it constitute very little portion of the network. Therefore, the resulting values are abrupt and unrealistic, and are unable to depict the factual robustness of the network. Fig. 3.6 illustrates the process of the propagation of targeted failures within a ThreeTier2k network. (Fig. 3.6a) depicts the initial network, and (Fig. 3.6b) shows the disconnected network with 1 percent targeted failures. The nodes at different layers of the network are shown in different colors and sizes.

The algebraic connectivity μ_{v-1} is an important measure to evaluate that how difficult it is to break the network into islands or individual components. The algebraic connectivity for the 2K networks is presented in Fig. 3.7 and the details can be observed in Table 3.10. It is noteworthy to consider that although the DCell2K does not possess the highest value of μ_{v-1}, it exhibits a smooth and slow decline in the value of μ_{v-1} in random and network-only failures. However, in case of the targeted attack, the value of μ_{v-1} for the DCell2K drops significantly when 3 percent of the nodes fail. Such an abrupt decrease portrays that the DCell2K is vulnerable to the targeted failures when the percentage of node failure is increased. For the FatTree2K network, it can be observed that despite showing a clearly descending curve under random and network-only failures, the value of μ_{v-1} increases in case of the targeted failure

when more than 3 percent of the nodes fail. The ThreeTier2k also depicts a similar behavior as the FatTree2k network when percentage of the node failure increases in the targeted and network-only failures. However, such abrupt increase in the values of μ_{v-1} is due to the fact that μ_{v-1} is analyzed for a very small sized largest connected component. Therefore, the value of μ_{v-1} increases. It is noteworthy to consider that an increase in the value of μ_{v-1} for high percentages of node failures (4.5 or 6 percent) does not mean that the network is more robust. Because the value of μ_{v-1} is calculated only for the largest connected component, it cannot be inferred that the networks become more robust after failures.

The spectral radius or largest eigenvalue λ_1 analysis is presented in Fig. 3.8 and the detailed values are provided in Table 3.11. As observed in Fig. 3.8, the DCell2K has a smaller value of λ_1, but the value of decrease slightly for all of the considered percentages and types of failures. The FatTree2K also exhibit slight decrease in the value of λ_1 in random failures case. However, the value of λ_1 decreases almost linearly under network-only and targeted failures in the FatTree2K. The ThreeTier2K is significantly affected by the targeted failure, and the value of λ_1 divides almost to half with only 0.1 percent of the nodes failure.

The robustness analysis of the DCN architectures considering various failure types and percentages reveals the vulnerability of the ThreeTier and FatTree DCN architectures to the targeted and network-only failures. However, the DCell architecture exhibits graceful and little variation of the metric values in response to all of the failure types and percentages. Therefore, it can be inferred from the failure analysis that the DCell exhibits better robustness than the ThreeTier and FatTree architectures. Moreover, the results drawn from the initial robustness analysis of the DCN networks without failure (see Table 3.6) proves invalid. In contrary to the

values reported in Table 3.6, the failure analysis reveals that the DCell architecture exhibits better robustness. Therefore, it is evident that the classical robustness metrics are inadequate to evaluate the DCN robustness.

Table 3.10. Algebraic Connectivity ($\mu_{|v|-1}$) of the 2K Networks.

Percentage		Random			Targeted		Network-only failure		
	DC2K	*FT2K*	*TT2K*	*DC2K*	*FT2K*	*TT2K*	*DC2K*	*FT2K*	*TT2K*
0	0.1243	0.3152	0.0230	0.1243	0.3152	0.023	0.1243	0.3152	0.0230
0.1	0.1243	0.3152	0.0230	0.1229	0.3116	0.087	0.1243	0.2899	0.0120
0.5	0.1225	0.3152	0.0230	0.1205	0.2984	0.087	0.1225	0.2853	0.0120
0.9	0.1222	0.2921	0.0230	0.1164	0.2822	0.087	0.1222	0.2623	0.0117
1.3	0.1187	0.2894	0.0230	0.1126	0.2646	0.087	0.1187	0.2355	0.0117
1.7	0.11881	0.2781	0.02308	0.11041	0.24262	1	0.11881	0.20661	0.00672
2.1	0.11745	0.29799	0.01205	0.10776	0.22144	1	0.11745	0.2291	0.01337
2.5	0.11621	0.28985	0.01182	0.10536	0.18455	1	0.11621	0.18142	0.00693
3	0.11305	0.26777	0.01242	0.02845	0.14618	1	0.11305	0.22228	0.01337
4	0.1107	0.26647	0.01208	0.05186	0.4875	1	0.1107	0.16066	0.04555
5	0.10723	0.24866	0.01457	0.03693	0.48751	1	0.10723	0.19194	0.01167
6	0.09943	0.27956	0.01267	0.00552	0.4875	1	0.09943	0.15423	1

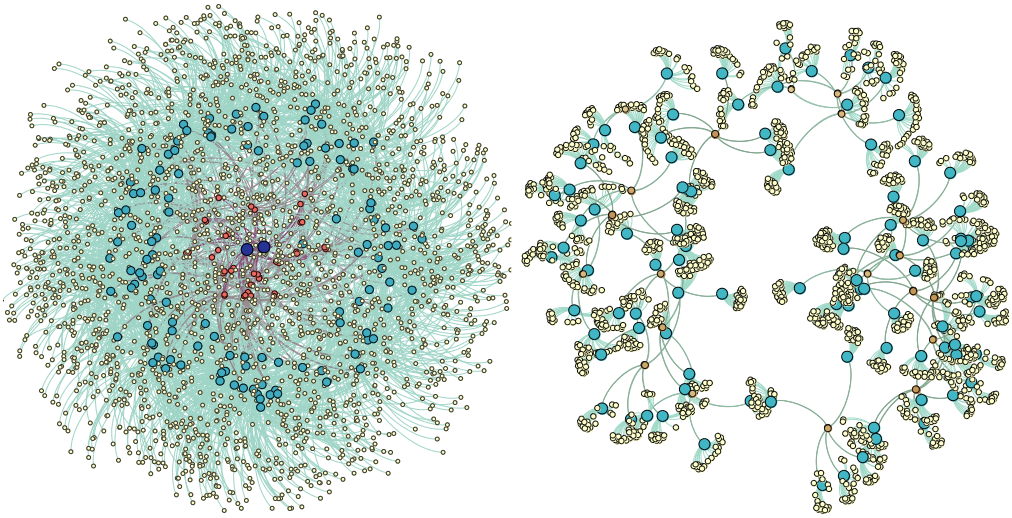


Fig. 3.6. ThreeTier DCN Before and After (1%) Targeted Failure.

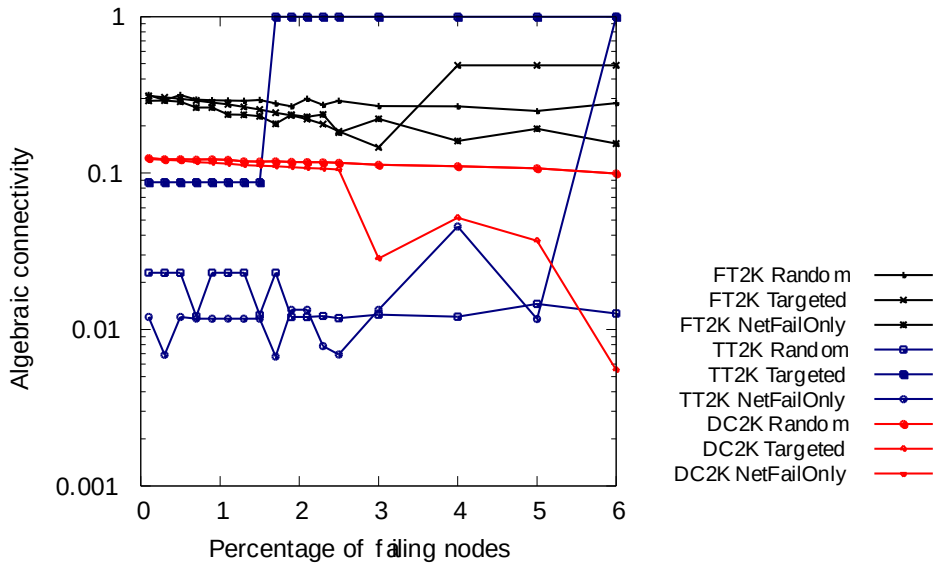


Fig. 3.7. Algebraic Connectivity Analysis of the 2K Networks.

Table 3.11. Spectral Radius (λ_1) of the 2K Networks.

Percentage		Random			Targeted		Network-only failure		
	DC2K	*FT2K*	*TT2K*	*DC2K*	*FT2K*	*TT2K*	*DC2K*	*FT2K*	*TT2K*
0	3.56155	17.4186	10.25044	3.56155	17.4186	10.25044	3.56155	17.4186	10.25044
0.1	3.55928	17.41844	10.25023	3.55873	17.269	5.87059	3.55928	17.29531	10.11993
0.5	3.54998	17.37591	10.23774	3.55031	16.75502	5.87059	3.54998	17.01578	9.94468
0.9	3.54272	17.31246	10.24875	3.54182	16.23312	5.87059	3.54272	16.7982	9.29244
1.3	3.53132	17.12761	10.24804	3.5342	15.69121	5.87059	3.53132	16.26041	8.87328
1.7	3.52701	17.05352	10.24152	3.52819	15.19576	4.47214	3.52701	15.95128	6.86486
2.1	3.51929	17.17246	10.10359	3.52321	14.21964	4.47214	3.51929	15.8202	8.35723
2.5	3.50736	16.99966	9.87806	3.5172	13.8575	4.47214	3.50736	15.6113	6.15025
3	3.49963	17.08234	9.86205	3.5033	12.63656	4.47214	3.49963	15.24117	7.35323
4	3.48352	16.69618	9.96564	3.48346	10.48809	4.47214	3.48352	13.95714	5
5	3.46261	16.53128	10.08541	3.46717	10.48809	4.47214	3.46261	13.77363	4.83971
6	3.43394	16.87975	9.94185	3.46481	10.48809	4.47214	3.43394	12.86074	4.47214

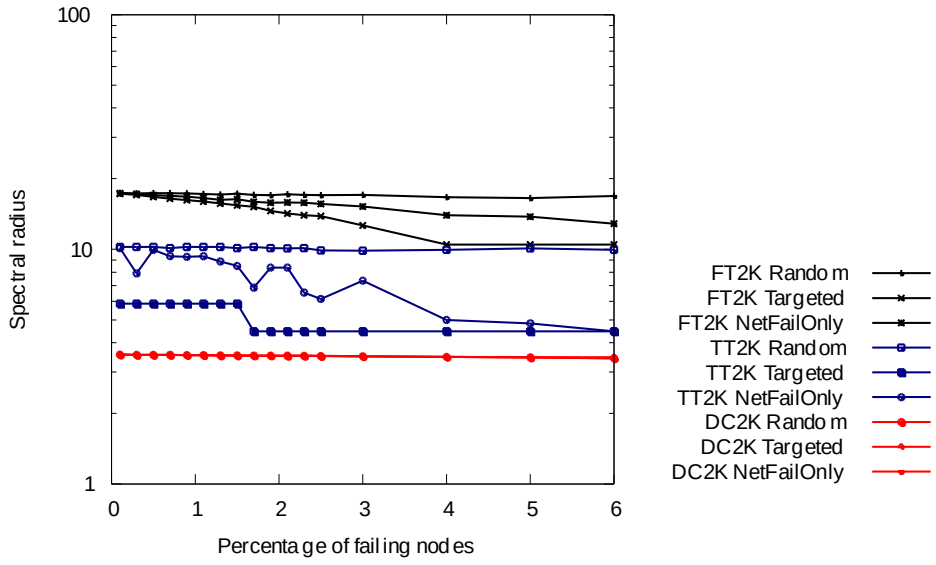


Fig. 3.8. Spectral Radius Analysis of the 2K Networks.

3.6.4. Real Failures in DCNs

This section presents the robustness measurements obtained from the largest connected components of the six networks (three 30K and three 2K), when the real failures within the DCNs are produced. As defined in Section 5, a specific number of nodes from each layer of network topology (based on the failure logs of various data centers) have been selected to fail, and the graph metrics have been computed for the resulting largest connected component.

The Table 3.12 presents the results of the real failures. All of the networks possess more than 90 percent of the nodes in the largest connected component in response to the real failures, as indicated by the value of max (υ). The results obtained from the real failures illustrate that the average nodal degree decreases slightly in all of the networks. The average shortest path length and diameter exhibit minor increase in all of the three networks. The assortativity coefficient value also depicts minor change for all of the DCN architecture. The value of ⟨b⟩ increases for all of the networks, indicating the increased vulnerability of the networks. The value of algebraic

connectivity exhibits comparatively higher decrease for the FatTree2K and ThreeTier2K than the DCell2K network. Similarly, the value of λ_1 also decreases significantly for the FatTree2K and ThreeTier2K as compared to the DCell2K. Therefore, the DCell2K network can be considered more robust network in case of the real failures while considering μ_{v-1} and λ_1.

All of the considered networks exhibit robust behavior in response to the real failures. However, the DCell architecture depicts graceful and minor variations in all of the observed metrics as compared to the ThreeTier and FatTree architectures. Therefore, the DCell DCN can be considered as the most robust architecture in case of the real failures.

3.6.5. Deterioration of DCNs

It has been observed that depending on the: 1) DCN architecture, 2) type of failure (whether it is random, targeted, network-only, or real), and (3) specific percentage of the nodes failed, the level of robustness according to a specific graph metric, computed from the largest connected component might be different. Moreover, the results for the various metrics exhibit strong dependence on the largest connected component, as observed in Section 7. Furthermore, the failure analysis depicts that the initial metric measurements are unable to quantify the DCN robustness appropriately (see Table 3.6 and Section 7.3). Therefore, we propose deterioration metric, a procedure for the quantification of the DCN robustness based on the percentage change in various graph metrics.

Deterioration σ_M, for any metric M can be calculated as the difference between the metric value for the whole network M_0, and the average of the metric values at various failure percentages M_i, divided by M_0.

$$\sigma_M = \left| \frac{\mathbf{1}}{M_{\mathbf{0}}} \left(\frac{\sum_{i=1}^{n} M_i}{n} - M_0 \right) \right| \tag{3.27}$$

Where M_i is measurement of the metric M at i percent of the nodes failure, and M_0 is the metric value for the whole of the network (without failure). To demonstrate that our proposed metric is able to quantify network robustness, we compute σ_M for:

1. six graph metrics namely:
    a. cluster size,
    b. average shortest-path length,
    c. nodal degree,
    d. algebraic connectivity,
    e. symmetry ratio, and
    f. spectral radius,
2. for the random, targeted, and real failures, taking into the account 1 to 6 percent of the nodes failure.

The results for the random, targeted, and real failures are presented in Figs. 9a, 9b, and 9c, respectively. The results depict that for almost all of the failure types, the σ_M or the DCell is much less as compared to the ThreeTier and FatTree architectures. The ThreeTier DCN exhibits the highest deterioration in random and network-only failures. However, for the real failures, the FatTree DCN exhibits more deterioration than the ThreeTier network. It is noteworthy to consider that for the real failures, the number of the failed nodes for the FatTree is around five times higher than the ThreeTier architecture (see Section 5). Our proposed σ_M evaluates the network robustness, and also allows to compare the results among various DCN architectures. The lower the value of deterioration, the higher is the robustness of the network.

It can be observed that the robustness of the considered networks evaluated by the deterioration metric complies with the robustness of the networks observed in Sections 7.2 and 7.3. Therefore, it can be stated that the deterioration metric can be employed to evaluate the robustness of the networks where the classical robustness metrics are inapplicable, such as the DCNs.

Table 3.12. DCN Features in Case of Real Failure.

Feature	*FatTree30K*	*ThreeTier30K*	*DCell30K*	*FatTree2K*	*ThreeTier2K*	*DCell2K*
$max(v)$	0.91	0.94	0.99	0.98	0.95	0.9
$\langle k \rangle$	4.7453	1.9523	3.7155	4.4825 ±7.07	2.0921 ±4.51	3.2899 ±0.95
$\langle l \rangle$	5.6417	5.901332	11.21	5.2863 ±1.09	5.7204 ±0.71	8.5909 ±1.95
D	7	6	23	7	6	15
r	-0.288845	-0.974917	-0.215366	-0.21106	-0.95623	-0.23834
$\mu_{\lvert v \rvert - 1})$	-	-	-	0.24157	0.00672	0.12017
$\langle b \rangle$	-	-	-	0.00181 ±0.00372	0.00205 ±0.02116	0.00284 ±0.0014
λ_1	-	-	-	16.34671	7.68104	3.53358

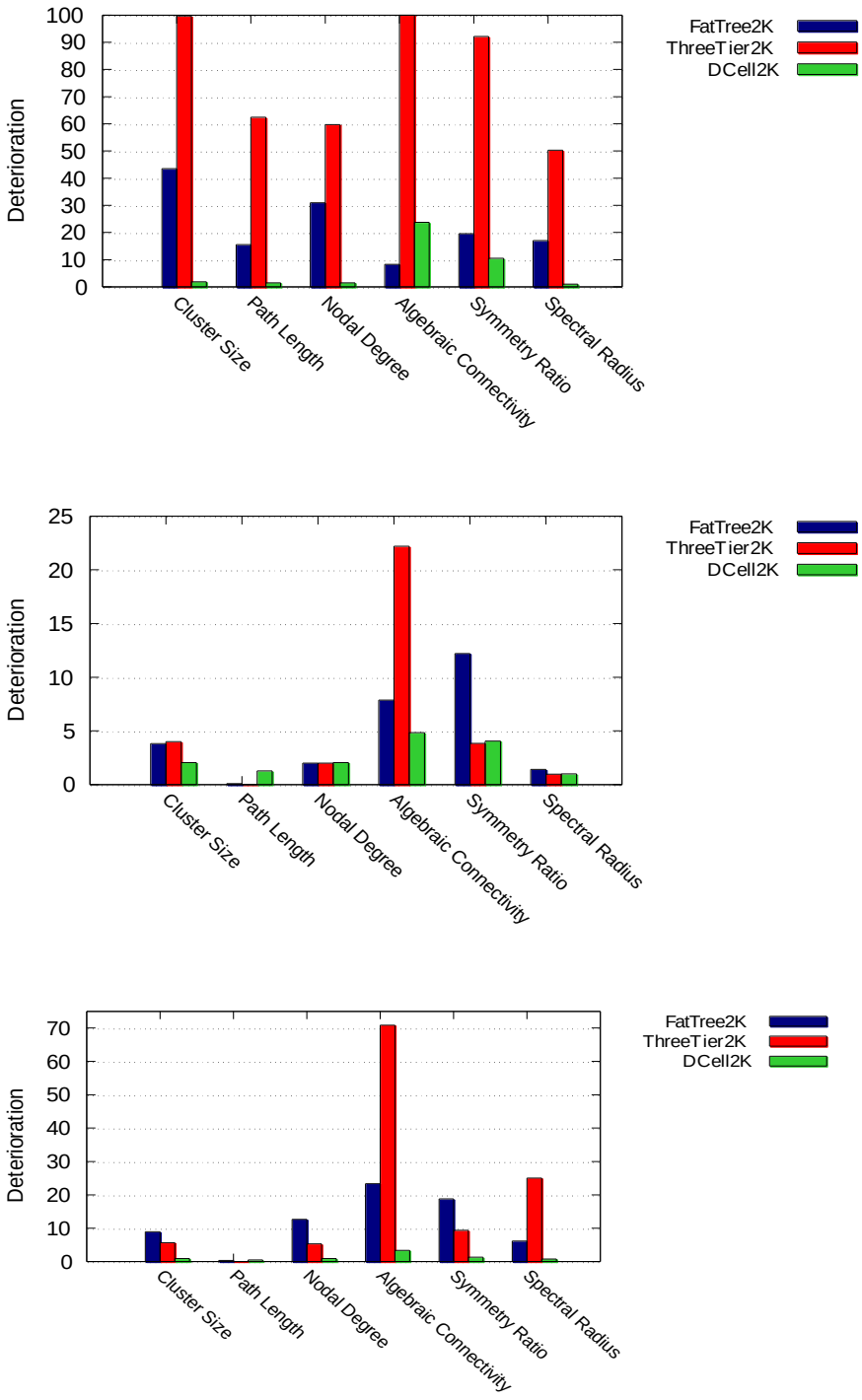


Fig. 3.9. Deterioration of the 2K Networks in Case of Random, Targeted, and Real Failures.

3.7. References

[3.1] K. Bilal, S.U.R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, U.S. Khan, A. Abbas, N. Jalil, and S.U. Khan, "A Taxonomy and Survey on Green Data Center Networks," *Future Generation Computer Systems*, 2013.

[3.2] P. Mel and T. Grance, "The Nist Definition of Cloud Computing," http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145. pdf, 2011.

[3.3] S. Ali, A.A. Maciejewski, H.J. Siegel, and J. Kim, "Definition of a Robustness Metric for Resource Allocation," *Proc. Int'l Parallel and Distributed Processing Symp*., p. 10, 2003.

[3.4] K. Bilal, S.U. Khan, L. Zhang, H. Li, K. Hayat, S.A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Xu, and A.Y. Zomaya, "Quantitative Comparisons of the State-of-the-Art Data Center Architectures," *Concurrency and Computation: Practice and Experience*, vol. 25, pp. 1771-1783, 2012.

[3.5] A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM SIGCOMM Computer Comm. Rev.*, vol. 39, no. 1, pp. 68-79, 2009.

[3.6] ITProPortal, http://www.itproportal.com/2012/07/12/o2- outage-latest-string-major-it-infrastructure-failures/, 2012.

[3.7] M. Manzano, E. Calle, and D. Harle, "Quantitative and Qualitative Network Robustness Analysis under Different Multiple Failure Scenarios," *Proc. Third Int'l Workshop Reliable Networks Design and Modeling (RNDM '11)*, pp. 1-7, 2011.

[3.8] A. Sydney, C. Scoglio, P. Schumm, and R.E. Kooij, "Elasticity: Topological Characterization of Robustness in Complex Networks," Proc. *Third Int'l Conf. Bio-Inspired Models of Network*, pp. 19:1-19:8, 2008.

[3.9] A.H. Dekker and B.D. Colbert, "Network Robustness and Graph Topology," *Proc. 27th Australasian Conf. Computer Science*, pp. 359-368, 2004.

[3.10] J. Dong and S. Horvath, "Understanding Network Concepts in Modules," *BMC Systems Biology*, vol. 1, no. 1, pp. 1-24, 2007.

[3.11] A.H. Dekker and B.D. Colbert, "The Symmetry Ratio of a Network," Proc. *Australasian Symp. Theory of Computing*, pp. 13- 20, 2005.

[3.12] C. Shannon and D. Moore, "The Spread of the Witty Worm," *IEEE Security and Privacy*, vol. 2, no. 4, pp. 46-50, July 2004.

[3.13] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, K.C. Claffy, and A. Vahdat, "The Internet AS-Level Topology: Three Data Sources and One Definitive Metric," *SIGCOMM Computer Comm. Rev.*, vol. 36, pp. 17-26, Jan. 2006.

[3.14] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, "Epidemic Thresholds in Real Networks," *ACM Trans. Information and System Security*, vol. 10, no. 4, pp. 1-26, 2008. [3.15] A. Jamakovic and S. Uhlig, "Influence of the Network Structure on Robustness," *Proc. 15th IEEE Int'l Conf. Networks (ICON '07*), pp. 278-283, 2007.

[3.16] Cisco Data Center Infrastructure 2.5 Design Guide, Cisco, 2010.

[3.17] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," *ACM SIGCOMM Computer Comm. Rev.*, vol. 38, no. 4, pp. 63-74, 2008.

[3.18] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers," *SIGCOMM Computer Comm. Rev.*, vol. 38, no. 4, pp. 75-86, Aug. 2008.

[3.19] L. Gyarmati and T.A. Trinh, "Scafida: A Scale-Free Network Inspired Data Center Architecture," *ACM SIGCOMM Computer Comm. Rev.*, vol. 40, no. 5, pp. 4-12, 2010.

[3.20] L. Gyarmati, A. Gulya´s, B. Sonkoly, T.A. Trinh, and G. Biczo´ k, "Free-Scaling Your Data Center," *Computer Networks*, vol. 57, pp. 1758-1773, 2013.

[3.21] J. Kim, W.J. Dally, and D. Abts, "Flattened Butterfly: A Cost- Efficient Topology for High-Radix Networks," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 126-137, 2007.

[3.22] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu, "FiConn: Using Backup Port for Server Interconnection in Data Centers," *Proc. IEEE INFOCOM*, pp. 2276-2285, 2009.

[3.23] K. Bilal, S.U. Khan, J. Kolodziej, L. Zhang, K. Hayat, S. Madani, N. Min-Allah, L. Wang, and D. Chen, "A Comparative Study of Data Center Network Architectures," *Proc. 26th European Conf. Modeling and Simulation*, pp. 526-532, May 2012.

[3.24] M. Kurant and P. Thiran, "Layered Complex Networks," *Physical Rev. Letters*, vol. 96, p. 138701, 2006.

[3.25] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with Multi-Layer Graphs: A Spectral Perspective," *CoRR*, vol. abs/1106.2233, 2011.

[3.26] C. Clos, “A Study of Non-Blocking Switching Networks,” *Bell System Technical J.*, vol. 32, no. 2, pp. 406-424, 1953.

[3.27] S. Neumayer and E. Modiano, “Network Reliability with Geographically Correlated Failures,” *Proc. INFOCOM*, pp. 1658- 1666, 2010.

[3.28] L.C. Freeman, “A Set of Measures of Centrality Based Upon Betweenness,” *Sociometry*, vol. 40, no. 1, pp. 35-41, 1977.

[3.29] B. Bolloba´s, Random Graphs. vol. 73, Cambridge Univ. Press, 2001.

[3.30] P.V. Mieghem, C. Doerr, H. Wang, J.M. Hernandez, D. Hutchison, M. Karaliopoulos, and R.E. Kooij, “A Framework for Computing Topological Network Robustness,” 2010.

[3.31] M. Youssef, R. Kooij, and C. Scoglio, “Viral Conductance: Quantifying the Robustness of Networks with Respect to Spread of Epidemics,” *J. Computer Science*, vol. 2, no. 3, pp. 286-298, 2011.

[3.32] E. Weisstein, http://mathworld.wolfram.com/GraphDiameter. html, 2013.

[3.33] P. Gill, N. Jain, and N. Nagappan, “Understanding Network Failures in Data Centers: Measurement Analysis, and Implications,” *Proc. ACM SIGCOMM*, 2011.

[3.34] R. Albert, H. Jeong, and A. Barabasi, “Error and Attack Tolerance of Complex Networks,” `, vol. 406, pp. 378-382, 2000.

[3.35] J. Guillaume, M. Latapy, and C. Magnien, "Comparison of Failures and Attacks on Random and Scale-free Networks," *Proc. Eight Int'l Conf. Principles of Distributed Systems*, 2005.

[3.36] P. Holme, B. Kim, C. Yoon, and S. Han, "Attack Vulnerability of Complex Networks," *Physical Rev. E*, vol. 65, no. 5, p. 056109, 2002.

[3.37] M. Manzano, V. Torres-Padrosa, and E. Calle, "Vulnerability of Core Networks under Different Epidemic Attacks," *Proc. Fourth Int'l Workshop Reliable Networks Design and Modeling*, 2012.

[3.38] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph Structure in the Web," *Computer Networks*, vol. 33, nos. 1-6, pp. 309-320, 2000.

[3.39] D. Callaway, M. Newman, S. Strogatz, and D. Watts, "Network Robustness and Fragility: Percolation on Random Graphs," *Physical Rev. Letters*, vol. 85, no. 25, 2000.

[3.40] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, "Resilience of the Internet to Random Breakdowns," *Physical Rev. Letters*, vol. 85, no. 21, p. 3, 2000.

[3.41] S.N. Dorogovtsev, J.F.F. Mendes, and A.N. Samukhin, "Giant Strongly Connected Component of Directed Networks," *Physical Rev. E*, vol. 64, no. 2 Pt 2, p. 4, 2001.

[3.42] M. Newman, S. Strogatz, and D. Watts, "Random Graphs with Arbitrary Degree Distributions and their Applications," *Physical Rev. E*, vol. 64, no. 2, p. 19, 2000.

[3.43] B. Luque and R. Sole´, “Phase Transitions in Random Networks: Simple Analytic Determination of Critical Points,” *Physical Rev. E*, vol. 55, no. 1, pp. 257-260, 1997.

4. ON THE CONNECTIVITY OF DATA CENTER NETWORKS

This paper is published in IEEE Communications Letters, vol. 17, no. 11, pp. 2172-2175, 2013. The authors of the paper are Marc Manzano, Kashif Bilal, Eusebi Calle, and Samee U. Khan.

4.1. Introduction

Cloud computing is an emerging paradigm that in the forthcoming years is expected to play a pivotal role in the Information and Communication Technology (ICT) sector. Data centers are the foundations of the cloud computing paradigm and are crucial for its operational and economic success. Data centers are composed of tens of thousands of hosts that are organized in clusters. Services are sourced from multiple clusters within the data center, and each cluster may host multiple services to increase system utilization. Most of the network communication, such as indexing, search, or other Map-Reduce tasks [4.1], take place within the data center [4.2]. For example, to process a single search query, thousands of servers within the data center are contacted in parallel [4.2]. The expected response time to the user is generally in tens of milliseconds [4.1], and a minor performance degradation or network congestion may result in a Quality of Service (QoS) violation.

Data Center Networks (DCNs) that constitute the communicational backbone of the cloud computing paradigm are of paramount importance to guarantee the system integrity [4.3]. The DCNs can be broadly classified into: (a) switch-centric and (b) server-centric or hybrid models [4.3]. The ThreeTier DCN is the most commonly used switch-centric architecture [4.4]. Al-Fares et al. used commodity network switches to design the FatTree switch-centric DCN architecture [4.5]. Guo et al. proposed the DCell; a hybrid DCN architecture [4.6] composed of

recursive building units called dcells. The aforementioned are the three most common DCNs [4.3].

Various network robustness and connectivity metrics have been proposed, such as the Average Two-Terminal Reliability ($A2TR$) [4.7], which take into consideration the physical topology and node interconnection of the network. To operate successfully, the DCNs are expected to possess high tolerance to network failures [4.3]. However, the networks may behave diversely when exposed to various types of node or link failures.

The A2TR is used to evaluate network connectivity in response to random failures [4.8], [4.9]. In this work we extend and customize the $A2TR$ procedure to evaluate targeted failures. Our analysis reveals that the DCNs exhibit diverse connectivity features and robustness in response to the targeted and random failures. As a consequence, we propose a new connectivity metric called $\mu - A2TR$, which evaluates how difficult it is to break a network into components according to a specific type of failure. We believe that our proposal will aid network engineers and the research community in designing more robust and better-connected DCNs.

Our major contributions include: (a) comparing the traditional network features of the state of the art DCNs namely: ThreeTier, FatTree, and DCell; (b) studying the DCNs architectural network connectivity in response to random and targeted node removals; and (c) proposing μ-A2TR, a metric to characterize the underlying connectivity of the DCNs.

4.2. Connectivity Analysis

According to the characteristics of the DCNs, it can been inferred that the FatTree is the least vulnerable network, followed by the DCell and ThreeTier architectures, respectively. To

examine the connectivity of the DCNs in detail, we evaluate the A2TR [4.7] value of each network in the case of three different types of node removals. The nodes to be removed are selected: (a) randomly, as discussed in various studies, such as [4.8], [4.9], [4.12]; (b) by their nodal degree; and (c) by betweenness centrality. The nodes with high betweenness centrality and nodal degree are selected for removal to demonstrate the system connectivity under targeted attacks [4.12], [4.13], [4.14].

The $A2T\ R(p)$ is the probability that a randomly chosen pair of the nodes is connected when p nodes are removed from the network. If the network is fully connected, the value of $A2TR$ is equal to 1. Otherwise, when p nodes are removed, the $A2TR$ value is calculated as the sum of the number of the node pairs in every strongly connected component (SCC) divided by the total number of node pairs in the network:

$$A2TR\ (p)\ =\ \frac{\sum_{i=1}^{|SCC|}|C_i|\ .\ (|C_i|-1)}{|N'|\ .\ (|N'|-1)}, \tag{4.1}$$

where $|C_i|$ is the number of nodes of the SCC number i, and $|N'|$ is the vertex size of the residual graph $|N|-p$. This ratio indicates the fraction of node pairs that are connected to each other. Therefore, the higher the $A2TR$ value (for a given number of removed nodes), the more connected the DCN is.

We compute the $A2TR$ value from p = 0 to $p = |N| - 2$, where $|N|$ is the total number of nodes in a DCN. In the procedure described in this section, the most expensive computation is in obtaining, for each p, the strongly connected components of the network. For that step, we use Tarjan's algorithm [4.15], whose running time complexity is in $O(|N\ |\ +\ |E|)$. The simulation

was performed on a Linux system with an 8-core 64-bit Intel Xeon processor of 2GHz and 16 GB of RAM. We employed a discrete-event simulation tool called PHISON [4.16].

4.3. Results

The results of the connectivity analysis are presented in Fig. 4.1, which depicts the $A2TR$ evolution according to the three types of node removals. The depicted values are the average of 1,000 runs with different random seeds, this being a widely used value in the bootstrap literature to carry out replications because it guarantees low variance [4.17].

In Fig. 4.1 it can be observed that for a lower percentage of randomly removed nodes (up to 40%), the DCell exhibits highly connected network, because of the high $A2TR$ values as compared to the ThreeTier and FatTree architectures. The ThreeTier network is more affected by the random removal of the nodes than the ThreeTier network. However, it is interesting to note that the connectivity of the DCell decreases extremely rapidly within the interval of 40% to 60% of removed nodes. Nevertheless, FatTree maintains a smooth linear decline for any percentage of removed nodes. Consequently, as discussed in Section II, of all three architectures considered and in response to high percentages of random node failures, the FatTree proves to be the most connected network.

The connectivity analysis of the DCNs observed in the case of high nodal degree and betweenness centrality based node removal differs significantly from the random nodes removal. The results presented in Fig. 4.2 and Fig. 4.3 depict the targeted removal of the nodes. It can be observed that the ThreeTier architecture is the most vulnerable network. Less than 10% of the node pairs remain connected to each other when removing only four nodes (core layer nodes) in the ThreeTier architecture. Contrary to the random nodes removal case, the FatTree network is

significantly affected by the targeted failures. However, the A2TR value curves of the FatTree exhibit a smoother decline than the ThreeTier A2TR value curves. Finally, of the three architectures considered, the DCell is the most connected network for targeted nodes removal cases.

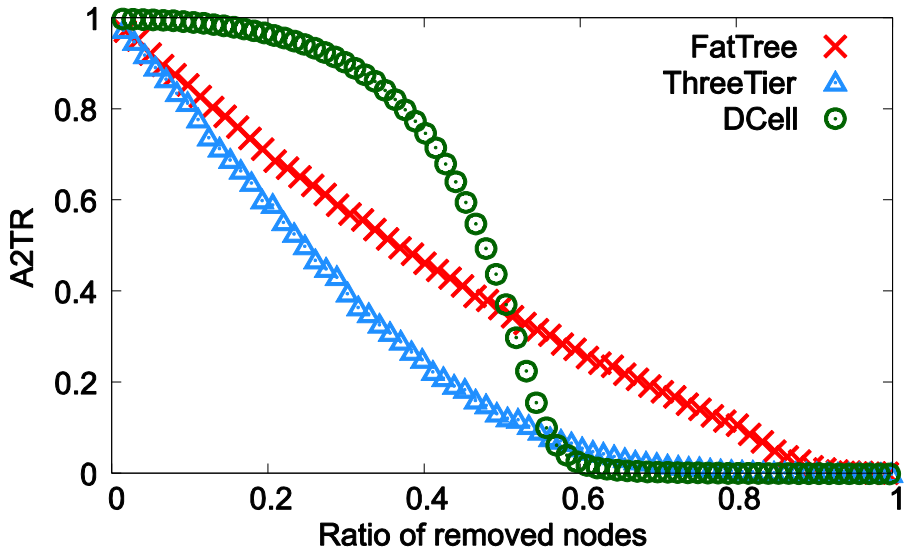


Fig. 4.1. A2TR of the DCNs for Random Failures.

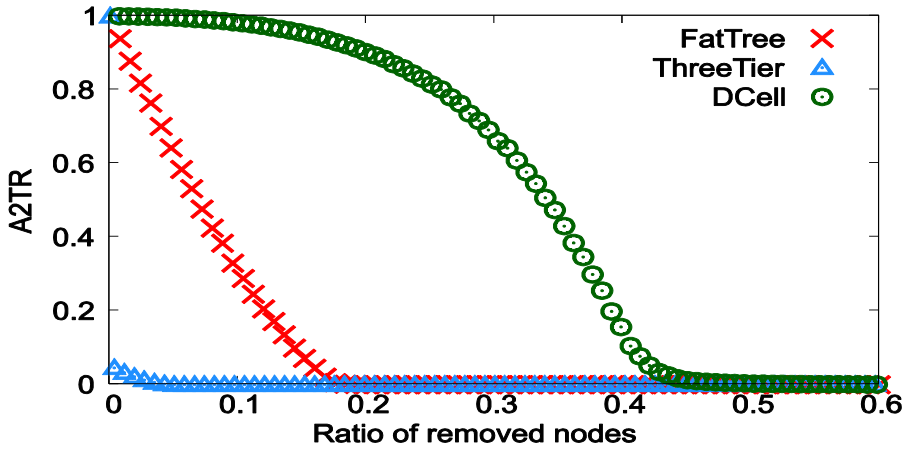


Fig. 4.2. A2TR of the DCNs for the Nodal Degree based Failures.

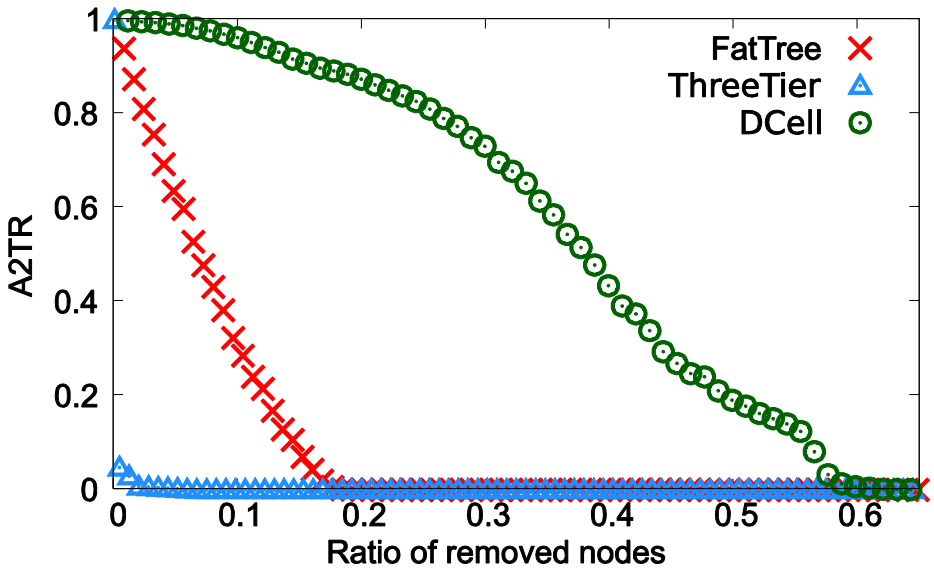


Fig. 4.3. A2TR of the DCNs for the Betweenness Centrality based Failures.

To conclude, it is worth noting that the network features for the DCN architectures do not accurately translate when evaluating the connectivity of the networks in various failure/node removal scenarios. Despite the fact that the FatTree exhibits better robustness features than the DCell architecture, the connectivity analysis demonstrates that the DCell architecture exhibits less vulnerability than the FatTree architecture. Therefore, it necessitates defining a new metric, which can accurately evaluate the connectivity of the DCNs.

4.4. μ-A2TR

In this section we present $\mu - A2TR$ as our third contribution to this letter, a novel metric to evaluate the connectivity of DCNs. We compute $\mu - A2TR$ for a given network and a given type of failure, from the $A2TR$ values which are obtained by conducting the analysis defined previously in this letter. The idea of considering the performance curve for increasing network damage was initially proposed in [4.18]. As a result, our proposal characterizes how difficult it is

to break a network into components when considering an incremental node failure scenario. Therefore, $A2TR$ can be defined as:

$$\mu - A2TR = \frac{\sum_{p=0}^{|N|-2} A2TR\ (p)}{(|N|-1)}, \tag{4.2}$$

where p is the number of nodes that have been removed from the network, and $A2T\ R(p)$ is the $A2TR$ value of the network for p removed nodes. $\mu - A2TR$ takes values over the interval [0, 1]. The higher the value of μ-A2TR, the more robust the DCN, in terms of connectivity, and more difficult to segregate the DCN into smaller clusters is.

Fig. 4.4 presents the $\mu - A2TR$ values for the three DCNs, and for the three types of node removals. As can be observed, the DCell architecture exhibits the highest $\mu - A2TR$: 0.45, 0.32, and 0.37 for the random, nodal degree, and betweenness centrality, respectively. The DCell architecture follows a recursively built topology, where each $dcell$ connects to l other $dcells$ (l is the level of the DCell architecture [4.3]). Moreover, the nodes within the DCell architecture exhibit low standard deviation in the nodal degree and betweenness centrality. Therefore, the DCell architecture exhibits high resilience to the node failures. On the contrary, the FatTree and ThreeTier architectures follow a hierarchical topology, where some of the nodes (core layer nodes) possess high nodal degree and betweenness centrality. In case of the targeted failures, these nodes are chosen for removal, resulting in network segregation and low connectivity. However, the number of nodes in the core layer of the FatTree are higher than in the ThreeTier architecture, the latter only having 4 nodes in our case. Therefore, the FatTree architecture exhibits better connectivity in terms of $\mu - A2TR$ (0.41, 0.07 and 0.07 for the three types of node removals) than the ThreeTier (0.27 in the case of random removals and close to 0 in both of the targeted cases).

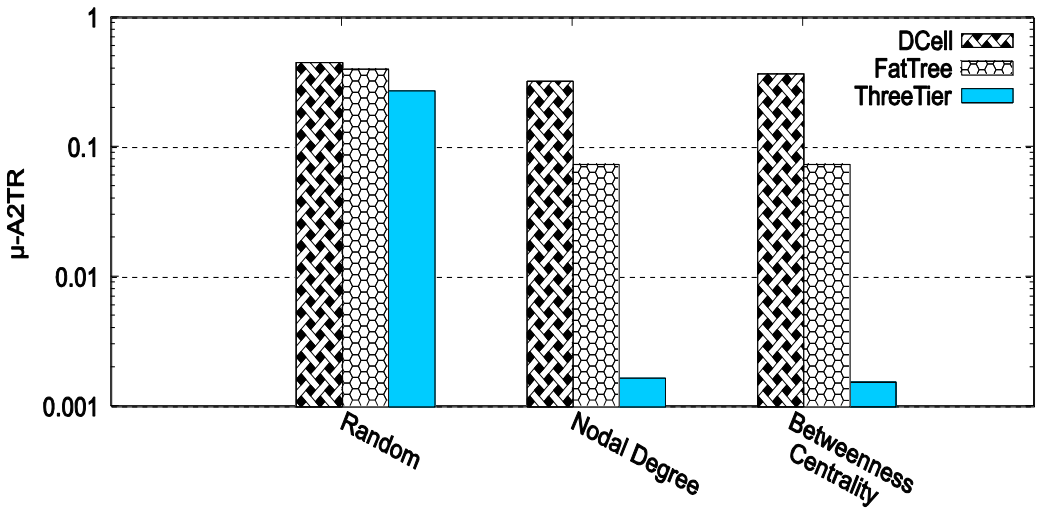


Fig. 4.4. μ − A2TR for the DCNs.

Unlike traditional graph features, our proposal describes how robust a network is in the case of specific failure scenarios. The $\mu - A2TR$ metric is able to denote significant differences between the three DCN topologies considered in this work. For instance, according to the algebraic connectivity or the spectral radius, the FatTree is the most robust network. However, $\mu - A2TR$ demonstrates that, although the DCell and the FatTree perform similarly in the case of random node removals, the former is more robust under targeted failures. In conclusion, $\mu - A2TR$ provides further insight into the connectivity of the DCNs, this being highly beneficial for the network research community given the critical role played by such networks currently.

4.5. References

[4.1] D. Abts and B. Felderman, "A guided tour of datacenter networking," *Commun. ACM – ACM Queue*, vol. 55, no. 6, pp. 44–51, 2012.

[4.2] A. Vahdat, H. Liu, X. Zhao, and C. Johnson, "The emerging optical data center," in *Proc. 2011 Optical Fiber Communication Conference*, pp. 8–10.

[4.3] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N.Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Xu, and A. Y. Zomaya, "Quantitative comparisons of the state-of-the-art data center architectures," *Concurrency and Computation: Practice and Experience*, 2012.

[4.4] Cisco, *Cisco Data Center Infrastructure 2.5 Design Guide*, 2010.

[4.5] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.

[4.6] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: a scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 75–86, 2008.

[4.7] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures," in *Proc. 2010 Conference on Information Communications*, pp. 1658–1666.

[4.8] M. Manzano, E. Calle, and D. Harle, "Quantitative and qualitative network robustness analysis under different multiple failure scenarios," in *Proc. 2011 International Workshop on Reliable Networks Design an Modeling*, pp. 1–7.

[4.9] M. Manzano, J.-L. Marzo, E. Calle, and A. Manolova, "Robustness analysis of real network topologies under multiple failure scenarios," in *Proc. 2012 European Conference on Networks and Optical Communications*.

[4.10] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, K. C. Claffy, and A. Vahdat, "The Internet AS-level topology: three data sources and one definitive metric," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 17–26, 2006.

[4.11] A. H. Dekker and B. D. Colbert, “Network robustness and graph topology,” in *Proc. 2004 Australasian Conference on Computer Science*, pp. 359–368.

[4.12] J. Guillaume, M. Latapy, and C. Magnien, “Comparison of failures and attacks on random and scale-free networks,” in *Proc. 2005 International Conference on Principles of Distributed Systems*, pp. 186–196.

[4.13] P. Holme, B. Kim, C. Yoon, and S. Han, “Attack vulnerability of complex networks,” *Physical Review E*, vol. 65, no. 5, p. 056109, 2002.

[4.14] M. Manzano, V. Torres-Padrosa, and E. Calle, “Vulnerability of core networks under different epidemic attacks,” in *Proc. 2012 International Workshop on Reliable Networks Design and Modeling*.

[4.15] R. E. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM J. Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[4.16] M. Manzano, J. Segovia, E. Calle, and J.-L. Marzo, “Phison: playground for high-level simulations on networks,” in *Proc. 2012 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*.

[4.17] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[4.18] A. Sydney, C. Scoglio, M. Youssef, and P. Schumm, “Characterising the robustness of complex networks,” *International J. Internet Technol. And Secured Transactions*, vol. 2, no. 3/4, pp. 291–320, 2010.

5. ROBUSTNESS QUANTIFICATION OF HIERARCHICAL COMPLEX NETWORKS UNDER TARGETED ATTACKS

This paper is submitted to Physica A and is in the second round of review. The authors of the paper are Kashif Bilal, Marc Manzano, Eusebi Calle, Catrina Scoglio, and Samee U. Khan.

5.1. Introduction

The society today is more dependent than ever on complex networks, such as the transportation and power networks, Internet, and Data Center Networks (DCNs). A complex network is generally denoted by the structural complexity, network evolution characteristics, connection diversity, dynamical complexity, and node diversity [5.1]. It has been argued that most of the complex networks inherently follow a hierarchical organization [5.2, 5.3]. However, there is no widely accepted definition of hierarchical networks, mainly because the definition of hierarchy involves several descriptors, such as *order*, *levels*, *inclusion*, or *control* [5.4]. Mones *et al*. discussed three types of network hierarchies within the complex systems, namely: flow, nested, and order hierarchy [5.2]. In a flow hierarchy, the nodes are organized and connected as a layered graph having multiple layers. The nodes in the lower layer are influenced by the nodes in the upper layer. Most of the complex networks within the domain of computer science and engineering, such as DCNs that form the backbone of the cloud computing [5.5], exhibit flow hierarchy. Conversely, a nested hierarchy is composed of high level and lower level elements, where high level elements contain lower level elements [5.6]. The ordered hierarchy is based on an ordered set, where ordering is made up of the values of the variables in the set of elements [5.7].

Recently, a 3-D morphological framework has been proposed to characterize quantitatively the concept of *hierarchy* [5.4]. However, the framework cannot be applied to undirected networks. Several quantitative hierarchy measures can be found in the literature, such as the hierarchical path and Global Reaching Centrality (GRC) [5.8]. Among all of the aforesaid, we consider the GRC hierarchical measure, because the GRC is applicable to any type of complex network, such as directed, undirected, weighted, or unweighted [5.2]. The GRC is based on the *Local Reaching Centrality* (LRC) that denotes the portion of nodes that can be reached via outgoing edges of a node. The GRC measure is computed as the difference between the maximum and average values of the LRCs within the network. The GRC values lie within the interval [0-1]. A directed tree network has the GRC value close to one; whereas, the GRC value of a homogeneous network, such as a lattice is near to zero (see Fig. 5.1).

Robustness is the ability of a network to deliver an anticipated level of performance, while sustaining component failures and system parametric perturbations [5.9]. For instance, the DCNs need to be robust against failures and system parametric perturbations for the successful and timely delivery of cloud services [5.10]. Minor performance degradation may result in enormous financial and reputation loss, as reported by Google and Amazon [5.11]. Therefore, the robustness analysis of the complex networks that represent the foundations of our modern society is extremely crucial. In general, the network robustness is evaluated by using classical graph metrics [5.9]. Some of the well-known network robustness metrics are discussed in [5.9, 5.12]. Various studies have been conducted for the robustness analysis of complex networks, such as biological, technological, and social networks [5.13]. However, to the best of our knowledge, there is no previous work that studies the impact of network hierarchy on the robustness of a network in case of intentional (or targeted) attacks.

In this work, we aim to emphasize the relationship between the network hierarchy and robustness. We use ten different networks for the said analysis. We calculate the network hierarchy using the GRC measure. The networks are then categorized into two classes based on the GRC values: **(a)** highly hierarchical (high GRC values networks) and **(b)** low hierarchical (low GRC valued networks). We employ various classical robustness metrics to measure the network robustness prior to and after the targeted failures. To imitate targeted attacks and node failures within a network, we choose the failing nodes based on the highest nodal degree and betweenness centrality of the nodes. We choose targeted failures by the nodal degree and betweenness centrality to represent the worst-case scenarios [5.14]. The failures are performed from 1% to 5% of the nodes within a network. To quantify the robustness of the networks after failures, we use the *deterioration* procedure to find the percentage change in the value of the metrics for the network. Our analysis reveals a strong relationship between the hierarchy and robustness of a network. The highly hierarchical networks are more vulnerable to the targeted attacks than the low hierarchical networks. Our major contributions can be summarized as:

- evaluate hierarchy of the networks using the GRC measure,
- measure network robustness using various classical metrics under targeted attacks,
- employ *deterioration* procedure to quantify the network robustness after targeted failures,
- analyze the relationship between the GRC value and classical graph metric results,

- investigate the correlation between the hierarchy and the robustness of a network under targeted failures.

5.2. Preliminaries

The robustness of complex networks has been extensively studied in the past decade. The studies were aimed to understand the physical connectivity of a network and the effects of the random and targeted failures within a network using classical graph metrics [5.15, 5.13, 5.16]. Various new metrics to capture the advanced robustness characteristics were also proposed in [5.17, 5.18]. Moreover, the correlations of traditional graph metrics were also considered and various interesting observation were made, such as the average shortest path length and clustering coefficient were strongly correlated [5.19]. Recently, several metrics have been proposed to consider the performance of a network under failure scenarios [5.20, 5.21, 5.22]. However, the relationship between the robustness of a network under targeted attacks and the underlying hierarchical structure of the network has not been considered in detail. Therefore, we aim to analyze the effect of the hierarchy on the robustness of a network in response to targeted attacks. To do so, we define the following case study for a set of ten different complex networks:

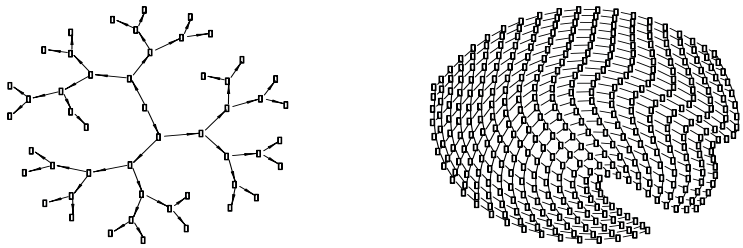


Fig. 5.1. Layout of a Directed Tree (Left) and a Lattice of 20 × 20 Nodes (Right).

- We consider the GRC measure to quantify the hierarchy of complex networks.
- We define two scenarios of node targeted attacks using the nodal degree and node betweenness centrality.
- Failures are instigated from 1% to 5% of the nodes within the networks.
- We calculate the GRC value for each of the considered networks, and classify the networks into highly hierarchical and low hierarchical network.
- We use six classical graph metrics to evaluate the robustness of each of the networks in each of the failure scenarios.
- We study the relationship between the GRC metric and the robustness of networks using the deterioration procedure when incremental and irreversible attacks occur.

The rest of this section is dedicated to present the networks and define the different measures that are used in our posterior analysis.

5.2.1. Networks

In this study we consider ten networks. The details of the networks are discussed below.

- PowerlawN400: a power-law network that has been generated using the Barabási Albert (BA, preferential attachment mechanism) model [5.23].
- EuroPGN1400: an approximated model of the European power grid network [5.24].

- ASN25K: an Autonomous System (AS) network of the year 2012 [5.25].
- ASN26K: the largest AS connected graph from the network set available in November 2007 in the Cooperative Association for Internet Data Analysis (CAIDA) repository [5.26].
- USmotorway: the US interstate highway topology [5.27].
- ATTN383: a physical fiber network of AT&T [5.28].
- SprintN300: a physical fiber network of Sprint [5.28].
- ThreeTierN2K: commonly used network topology within data centers [5.29, 5.30]
- FatTreeN2K: the Clos based network topology proposed for data centers [5.31]
- DCellN2K: hierarchical server-centric topology proposed for data centers [5.32]

Table 5.1 depicts the main features of the networks. As observed, our set of networks is heterogeneous with respect to size, number of edges, average nodal degree, and maximum nodal degree.

5.2.2. Global Reaching Centrality (GRC)

Several measures have been proposed to measure the network hierarchy [5.33, 5.34, 5.35, 5.4]. Some of the hierarchical measures require the definition of free parameters that are unknown for many networks [5.33, 5.34]. Some of the proposals only quantify the deviation from the pure tree structure to measure the hierarchy of the network [5.35], or are only applicable to fully directed graphs [5.4]. We consider the GRC measure as it is applicable to any type of

complex network, such as undirected and unweighted [5.2]. Given a graph *G*, the GRC can be defined as follows:

$$GRC = \frac{\sum_{i \in V}[C_R^{max} - C_R(i)]}{N-1}, \tag{5.1}$$

where *V* denotes the set of nodes, *N* is the number of nodes within *G*, $C_R(i)$ is the *Local Reaching Centrality* (LRC) of the node *i*, and C_R^{max} is the highest LRC value among all of the nodes within *G*. The GRC values lie within the interval [0,1]. A higher GRC value depicts a higher hierarchy of the network. The LRC value depicts the portion of the nodes of a network that can be reached from a node *i*. For unweighted and undirected graphs, the LRC is denoted as:

$$C_R(i) = \frac{1}{N-1} \sum_{j:0<d(i,j)<\infty} \frac{1}{d(i,j)}, \tag{5.2}$$

where $d(i, j)$ is length of the shortest path from the node *i* to *j*. It must be noted that this value for undirected and unweighted graphs is very similar to the *local closeness centrality* [5.36].

Fig. 5.2 illustrates the GRC measures for the set of ten networks considered in this work. As can be observed, there are five networks that have the value of GRC greater than 0.14, while the other five present the GRC values below 0.05. For the ease of the analysis and comparison, we classify the considered networks into two categories. The network with the GRC value greater than 0.14, such as the ASN25K, ASN26K, Powerlaw400, ThreeTierN2K, and FatTreeN2K, fall under the former category (referred as *high GRC valued* networks in the paper). Alternatively, the network with the GRC value less than 0.05, such as the ATTN383, EuroPGN1400, Sprint200, USmotorway, and DCellN2K, fall under the latter category (referred as *low GRC valued* networks in the paper). The GRC values of the networks are reported in Table 5.2.

Table 5.1. Network Characteristics.

Network	Number of nodes	Number of edges	Average nodal degree	Maximum nodal degree
ASN25K	25,357	74,999	5.91	3,781
ASN26K	26,475	53,381	4.03	2,628
PowerlawN400	400	399	2	47
ThreeTierN2K	2,562	2,740	2.1389	40
FatTreeN2K	2,500	6,000	4.8	20
USmotorway	411	553	2.69	7
ATTN383	383	488	2.54	8
EuroPGN1400	1,494	2,154	2.88	13
SprintN300	264	313	2.37	6
DCellN2K	2,709	4,515	3.3333	4

Table 5.2. The GRC Values of the Networks.

High GRC valued Networks	ASN25K	ASN26K	PowerlawN400	ThreeTierN2K	FatTreeN2K
GRC Value	0.24336	0.19728	0.18713	0.17199	0.14343
Low GRC Valued Networks	USmotorway	ATTN383	EuroPGN1400	SprintN300	DCellN2K
GRC Value	0.05559	0.04579	0.04319	0.0413	0.00715

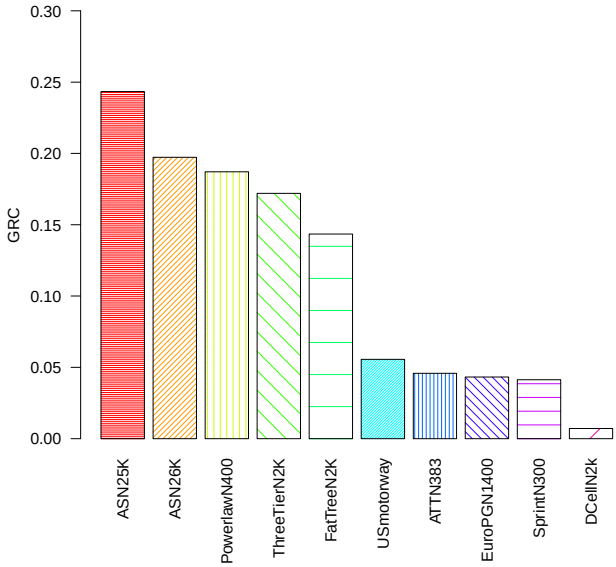


Fig. 5.2. The GRC Values of the Networks.

5.3. Robustness Analysis

5.3.1. Initial Network Analysis

We have analyzed the robustness of the ten complex networks considering the classical robustness metrics. Table 5.3 presents the values of the classical metrics for the networks without any failures. Table 5.3 is sorted based on the decreasing GRC values to depict the relation between hierarchy and robustness estimation of the network. As can be observed from the table, the networks with high GRC values (highly hierarchical networks) depict better robustness in most of the cases without failures. The FatTreeN2K and ThreeTierN2K networks have the least diameter value of six. The ASN25K network has a slightly higher diameter value than the FatTreeN2K and ThreeTierN2K networks, but a smaller diameter than most of the low GRC

valued networks, such as USmotorway and EuroPGN1400. The ASN26K network has the highest diameter value of seventeen within the highly GRC valued networks; whereas, the EuroPGN1400 network has the highest diameter of 44 within the low GRC valued networks. The average diameter value for the high and low GRC valued networks was eleven and 36.2, respectively. The average diameter is calculated to highlight the difference of the initial metric values for both of the network groups.

As discussed in Section 2.3, the networks with low average shortest path lengths (denoted as ASP) are considered more robust. It can be observed from the metrics results presented in Table 5.3 that in general, the high GRC valued networks have low average path lengths. The ASN25K and ASN26K networks have the average path lengths of 3.39 and 3.87, respectively that makes them the most robust network based on the path length metric. On the contrary, low GRC valued networks exhibit high average path length values. The average of the high GRC valued networks is 4.84; whereas, the low GRC valued networks have 13.97 as the average value for the path length metric.

The networks with high spectral radius are considered more robust. A similar trend of high GRC valued networks to exhibit better robustness can also be observed when considering the spectral radius metric. The ASN25K and ASN26K with the highest GRC value also exhibit the highest spectral ratio values of 103.36 and 69, respectively. Alternatively, the spectral radius values of the low GRC valued networks group are comparatively very low. The average spectral radius values for high and low GRC valued network groups are 41.53 and 3.87, respectively.

An analogous trend of robustness estimation based on the algebraic connectivity metric can also be observed in both of the network groups. The high GRC valued networks exhibit high

algebraic connectivity values than the low GRC valued networks. The only exception is the DCellN2K network that has a high algebraic connectivity value, almost equal to the ASN25K network. However, the algebraic connectivity value of the DCellN2K is three times less than the FatTreeN2K network. In general, high GRC valued networks exhibit high algebraic connectivity values with an average of 0.0942 than the low GRC valued networks having an average value of 0.0256. The DCellN2K and PowerlawN400 networks are to be considered as the exceptions in both of the network groups, while considering the algebraic connectivity values.

The networks with high average nodal degrees are considered more robust. The networks having a high GRC values in general, exhibit high average nodal degrees. The ASN25K and ASN26K networks exhibit the highest nodal degrees of 5.91 and 4.03, respectively. However, the PowerlawN400 network is an exception within the group having an average nodal degree of 1.995. The average value of the nodal degree for high and low GRC valued groups was 3.77 and 2.76, respectively.

In general, one may infer from the initial metric values that the higher the hierarchy, more robust is the network. However, we will see in Section 4 that the aforementioned assumption does not hold true in case of targeted failures where low GRC valued networks exhibit better hierarchy than the higher GRC valued networks, leading to the fact that the initial robustness estimation of the hierarchical networks using the classical robustness metrics may be misleading and erroneous.

5.3.2. Deterioration

We present a procedure for the quantification of the network robustness and comparison between various heterogeneous networks (in terms of size), named *deterioration,* based on the

percentage change in the metric values. The deterioration can be calculated as the difference between the initial value (without failure) of the robustness metric *M* and the average of the metric values at various failure percentages, divided by the initial value. The average of the metric values is considered to minimize the effect of a critical point and phase transition for various networks [5.37]. Some of the networks exhibit phase transition after a specific number of failures. For instance, the FatTree network exhibits a phase transition at 1.9% of targeted failures based on the betweenness centrality failures [5.10]. For targeted failures up to 1.8%, the FatTree holds around 98% of the nodes in the largest connected cluster. However, the total number of nodes within the largest cluster drops to 2% of the original network size, when targeted failures percentage reaches 1.9%. Therefore, we use the average of the metric values for the entire failure range to minimize the impact of phase change at a specific percentage of failure. Moreover, various metrics, such as diameter and shortest path length can only be calculated for connected component of the graph. Therefore, the resultant values of the aforementioned metrics may be misleading for a specific percentage of failure. For instance, the ThreeTierN2K network totally segregates after 5% of targeted failures, where the size of largest connected component is two. Therefore, the resultant diameter and path length is equal to one. Similarly, the size of the largest connected component after 5% of targeted failures in the ASN26K network was 35, compared to the initial network size of 26,475. Therefore, the resultant value for the diameter and average path length was 16 and 2.3, respectively. However, for 3% of the failures for the ASN26K network, the values for the largest connected component, diameter, and average path length was 718, 46, and 15.55, respectively. Consequently, considering only the final values after 5% of failure will be misleading. Therefore, we take the average for all of the percentages of failures

for calculating the percentage change within the metric value. The deterioration can be presented as:

$$\sigma_M = \left| \frac{\mathbf{1}}{M_0} \left(\frac{\sum_{i=1}^{n} M_i}{n} - M_0 \right) \right|, \tag{5.3}$$

where M_0 is the initial value of a metric M without failures, and M_i is the value of the metric when i percent of the nodes fail ($1 \leq i \leq n$). The lower the value of deterioration, the higher is the robustness of the network, indicating that the network has undergone the minimum number of changes. On the contrary, the larger the change in values of various metrics of a network, the higher the deterioration is, depicting less robustness of the network. To demonstrate the validity of the proposed procedure, we calculate σ_M for the: diameter, algebraic connectivity, spectral radius, nodal degree, largest connected cluster size, and average shortest path length.

Fig. 5.3 – Fig. 5.8 present the deterioration results for the considered metrics with respect to the nodal degree and betweenness centrality based targeted failures. For both of the failures scenarios (nodal degree and betweenness centrality), it can be observed that the low GRC valued networks exhibit lower deterioration in most of the cases depicting the higher robustness to targeted failures.

5.3.3. Robustness Analysis under Targeted Failures

To analyze the effect of the failures and the validity of the robustness metrics, we evaluate the considered networks by instigating targeted failures, and removing the nodes based on the: highest nodal degree and highest node betweenness centrality values. The aforementioned targeted failure criteria are used in various studies, such as [5.38, 5.14]. The nodal degree represents the number of outgoing physical connections of a node with its

neighbors. The betweenness centrality is used to estimate the prestige of a node, and is measured as the number of the shortest paths between all of the node pairs within the network that pass through that node [5.9]. Therefore, the two failure criteria consider the structural as well as the operational aspects of the network. The failures are introduced within the network from within a range of 1% – 5% of the total number of the nodes.

Table 5.3. Classical Metric Values for the Considered Network Topologies.

Network	GRC	Diameter	Nodal Degree	ASP	Cluster Size	Algebraic Connectivity	Spectral Radius
High GRC valued Networks							
ASN25K	0.243	10	5.9154	3.3984	25,357	0.10768	103.361
ASN26K	0.197	17	4.0325	3.8756	26,475	0.02043	69.642
PowerlawN400	0.187	16	1.995	6.0065	400	0.00463	7.01347
ThreeTierN2K	0.171	6	2.1389	5.7247	2,562	0.02307	10.25044
FatTreeN2K	0.143	6	4.8	5.2106	2,500	0.31526	17.4186
Average	0.188	11	3.776	4.8432	11,458.8	0.094214	41.5371
Low GRC valued Networks							
USmotorway	0.055	42	2.6909	13.654	411	0.00547	4.2156
ATTN383	0.045	39	2.5483	14.129	383	0.00554	3.7069
EuroPGN1400	0.043	48	2.8835	18.888	1494	0.00169	5.0272
SprintN300	0.041	37	2.3712	14.704	264	0.00535	2.9317
DCellN2K	0.007	15	3.3333	8.5106	2709	0.11021	3.475
Average	0.038	36.2	2.765	13.977	1052.2	0.0256	3.871

The values of the largest connected cluster size and deterioration under the targeted failures for the considered networks are reported in Table 5.4 and Table 5.5, for the nodal degree and betweenness centrality based failures, respectively. The reported values represent the cluster size and deterioration values for the metric results obtained after instigating 1% – 5% of the failures within the network. The values for the shortest path length, diameter, algebraic connectivity, and spectral radius, are calculated from the largest connected cluster of the resulting network after targeted failure. For an ease of comparison and observation, the initial size of the network without failure (labeled as Initial in the second column), the resultant largest connected cluster size obtained after 5% of the node failures (labeled as 5% F), and the deterioration value (labeled as D) are reported in the Table 5.4 and Table 5.5, respectively. The rows in the tables are sorted based on the decreasing GRC values to highlight the relation between the GRC value and deterioration. It can be observed from the tables that the networks with high GRC values exhibit more deterioration in contrary to the initial observations obtained from the robustness metric values without failure, as reported in Section 3.1. The detailed analysis of various metric results and deterioration is presented below.

5.3.4. Robustness Analysis Considering the Classical Metrics

5.3.4.1. Cluster Size

The largest connected cluster (giant cluster) is one of the simplest robustness measures. A network with high robustness must retain most of the nodes in the giant cluster depicting a minimum deterioration and segregation. Such a behavior of the network having high giant cluster size after failures is a natural and easy estimation of the network's robustness. We argue that the networks depicting higher hierarchical organization and high GRC values exhibit low robustness

in terms of the giant cluster size. It can be observed from the initial and after 5% failure size of the giant cluster (see Table 5.4 and Table 5.5) that the high GRC valued networks get segregated in very small sized clusters where the giant cluster has very little number of nodes as compared to the initial graph. For instance, the ASN25K, ASN26K, PowerlawN400, ThreeTierN2K, and FatTreeN2K, show 82%, 89%, 94%, 97%, and 18% deterioration in the giant cluster size for the nodal degree based targeted failures, respectively.

Table 5.4. Deterioration Values for 1% – 5% of Nodal Degree based Failures.

Network	Cluster Size			Diameter	Nodal Degree	ASP	Algebraic Connectivity	Spectral Radius
	Initial	5%F	D					
ASN25K	25,357	51	0.83	1.933	0.867	1.756	0.941	0.936
ASN26K	26,475	35	0.90	0.716	0.854	1.238	0.658	0.936
PowerlawN400	400	9	0.95	0.573	0.415	0.593	22.189	0.592
ThreeTierN2K	2,562	2	0.97	0.528	0.680	0.527	19.365	0.543
FatTreeN2K	2,500	1,901	0.18	0.194	0.243	0.017	0.315	0.127
USmotorway	411	376	0.07	0.122	0.088	0.266	0.403	0.152
ATTN383	383	328	0.12	0.208	0.102	0.219	0.370	0.037
EuroPGN1400	1,494	1,037	0.18	0.272	0.134	0.164	0.443	0.182
SprintN300	264	237	0.07	0.149	0.061	0.180	0.395	0.050
DCellN2K	2,709	2,572	0.04	0.052	0.036	0.027	0.085	0.017

Table 5.5. Deterioration Values for 1% – 5% of Betweenness Centrality based Failures.

Network	Cluster Size			Diameter	Nodal Degree	ASP	Algebraic Connectivity	Spectral Radius
	Initial	5%F	D					
ASN25K	25,357	51	0.80	2.32	0.85	2.17	0.91	0.78
ASN26K	26,475	35	0.88	0.87	0.84	1.61	0.89	0.88
PowerlawN400	400	9	0.94	0.55	0.39	0.58	22.19	0.59
ThreeTierN2K	2,562	2	0.97	0.43	0.62	0.53	19.37	0.55
FatTreeN2K	2,500	120	0.39	0.13	0.23	0.13	0.03	0.26
USmotorway	411	337	0.06	0.08	0.07	0.22	0.51	0.15
ATTN383	383	324	0.10	0.11	0.08	0.12	0.37	0.04
EuroPGN1400	1,494	1,036	0.15	0.29	0.12	0.18	0.51	0.19
SprintN300	264	232	0.08	0.33	0.06	0.31	0.47	0.03
DCellN2K	2,709	2,572	0.03	0.05	0.03	0.02	0.08	0.02

The FatTreeN2K network exhibits better connectivity in case of the nodal degree failures. However, for the betweenness centrality based failures, the FatTreeN2K networks exhibits 39% deterioration. This is due to the fact that all of the nodes in the upper three layers of the FatTreeN2K topology have the same nodal degree [5.10]. Therefore, the nodal degree based failures has little effect on network connectivity in case of the FatTree2K network.

Conversely, the networks with low hierarchy and GRC values, such as the USmotorway, ATTN383, SprintN300, EuroPGN1400, and DCellN2K depict 7%, 11%, 7%, 15%, and 3% deterioration in the giant cluster size, respectively. The deterioration results observed in the considered hierarchical networks depict that the networks with the low GRC values exhibit more tolerance to the targeted failures and show small variation in the giant cluster size. Therefore, a major portion and most of the nodes within the network stay connected even after 5% of the

failed/removed nodes. Conversely, the networks with high GRC values depict very high deterioration values, and the networks get fragmented and isolated in many small networks with very little number of the nodes in the giant cluster. A similar behavior and resultant values can be observed in Table 5.5 for the betweenness centrality based targeted failures as depicted by the nodal degree based failures. Such a behavior of the high GRC valued networks depicts weak tolerance against targeted attacks. The deterioration in size of the largest connected cluster illustrated in Fig. 5.3, also affirms that the high GRC valued networks are more prone to targeted failures and exhibit low robustness.

5.3.4.2. Average Shortest Path Length

The networks with small average shortest path length are considered more robust. The networks with higher GRC values, such as the ASN25K, ASN26K, and PowerlawN400 exhibit low average path length values of 3.39, 3.87, and 6.006, respectively. However, the deterioration values observed in case of the nodal degree based targeted attacks for the aforementioned networks are quite high with 175%, 123%, and 59%, respectively. Conversely, the networks with low hierarchical values, such as the USmotorway, ATTN383, SprintN300, and EuroPGN1400, have comparatively high average path values of 13.65, 14.12, 14.70, and 18.88, respectively. Whereas, the deterioration value of the aforementioned networks are 26%, 21%, 17%, and 16%, respectively for the nodal degree based failures. A similar trend can be observed in Table 5.5 for the betweenness centrality based failures. The deterioration values observed in the networks with low hierarchy and GRC values is quite small as compared to the high GRC valued networks. The aforementioned results depict that the average path length based robustness analysis of hierarchical networks may be misleading and inadequate for robustness quantification of hierarchical networks. Fig 4. depicts the results for the average path length deterioration for the

nodal degree and betweenness centrality based failures. The deterioration values in Fig. 5.4 are capped at value 1.0 to show a better comparison.

5.3.4.3. Diameter

The networks with low diameter are considered more robust. The diameter of a disconnected graph is infinite [5.39]. Therefore, the diameter of the largest connected cluster is calculated in case of disconnected networks. The high GRC valued networks have low diameter than the networks with low GRC values. On the contrary, higher deterioration and low giant cluster sizes are observed when a network with high GRC value is analyzed for the targeted attacks. The diameter of the high GRC valued networks, such as the ThreeTierN2K, FatTreeN2K, ASN25K, ASN26K, and PowerlawN400 are six, six, ten, seventeen, and sixteen, respectively. The diameter values for the aforementioned networks are quite small, compared to the networks with the low GRC values, such as USmotorway, EuroPGN1400, ATTN383, and Sprint, having diameter values of 42, 48, and 39, respectively. However, as can be observed in the Table 5.4 and Table 5.5, and as shown in Fig. 5.5, the deterioration values of the high GRC valued networks are very high as compared to the networks with low GRC values. Similar to the average shortest path length, it can be inferred that the diameter based robustness estimation of the hierarchical networks may be misleading and wrong as well.

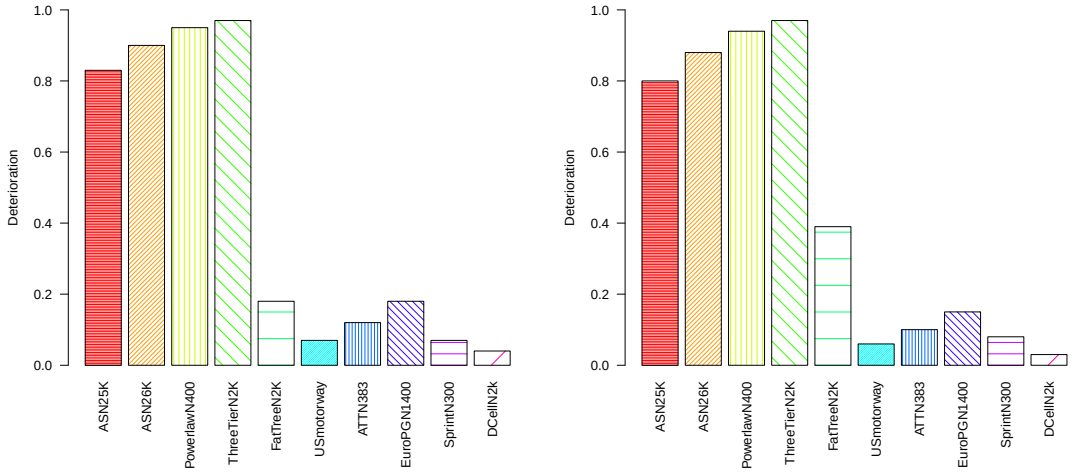


Fig. 5.3. Deterioration in Largest Connected Cluster Size based on: Nodal Degree (Left) and Betweenness Centrality (Right).

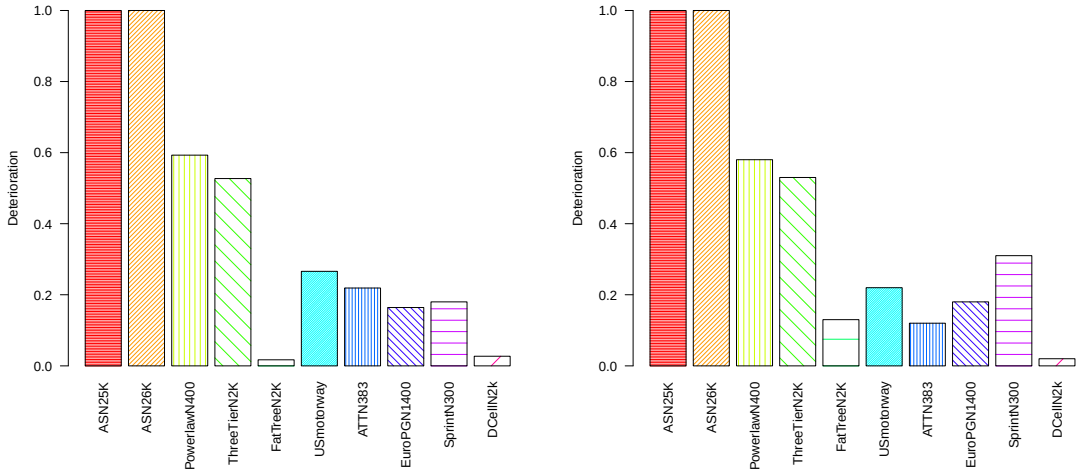


Fig. 5.4. Deterioration in Path Length based on: Nodal Degree and Betweenness Centrality.

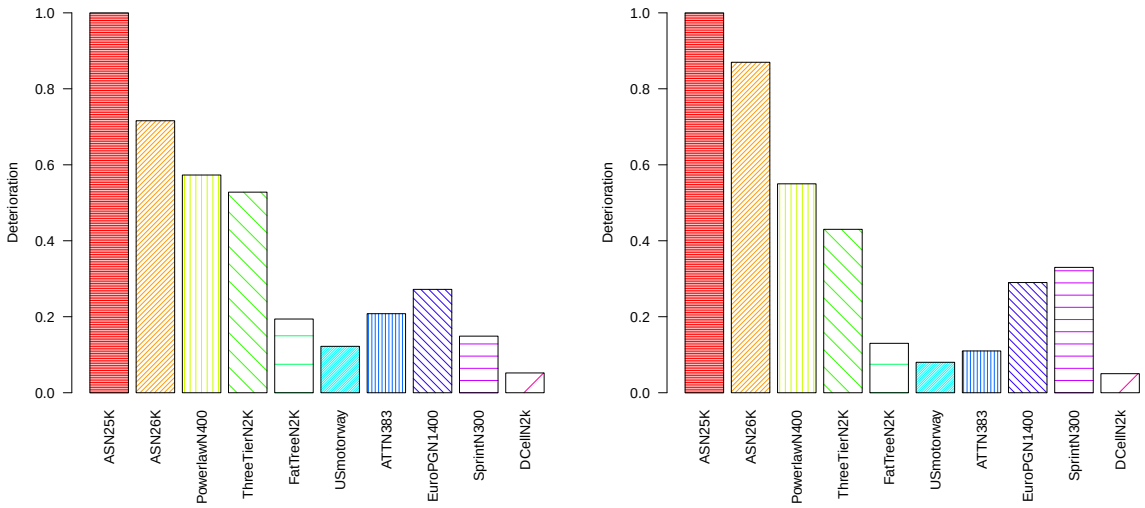


Fig. 5.5. Deterioration in Diameter based on: Nodal Degree and Betweenness Centrality.

5.3.4.4. Algebraic Connectivity

Network with high algebraic connectivity values are considered more robust. Most of the high GRC valued networks depict high algebraic connectivity values without failure, leading to the fact that such networks are more robust. The high GRC valued network, such as the ASN25K, ASN26K, ThreeTierN2K, and FatTreeN2K have 0.107, 0.020, 0.023, and 0.31 algebraic connectivity values, respectively. However, the deterioration values of the aforementioned networks are 94%, 65%, 193%, and 31%, respectively. In contrast, the algebraic connectivity values for low GRC values networks, such as USmotorway, ATTN383, EuroN1400, and Sprint300 are 0.005, 0.005, 0.001, and 0.005, respectively. The deterioration observed in these networks is 40%, 36%, 44%, and 39%, respectively, in case of nodal degree based failures. Moreover, the giant cluster size deterioration values for all of the aforementioned networks also depict conflicting behavior to the initial robustness estimation based on the algebraic connectivity values. Most of high algebraic connectivity valued based networks that

were expected to be more robust illustrated low resilience to the targeted attacks, and low algebraic connectivity value based network described better resilience and high robustness opposite to the initial robustness estimation. The algebraic connectivity based deterioration is illustrated in Fig. 5.6.

5.3.4.5. Spectral Radius

Spectral radius is another robustness measure where high spectral radius presents higher robustness of the network. However, in case of hierarchical networks, spectral radius based robustness estimation of hierarchical networks is misleading as well. The spectral radius value of high GRC valued networks, such as the ASN25K, ASN26K, PowerlawN400, ThreeTierN2K, and FatTreeN2K were 103.36, 69, 7.01, 10.25, and 17.41, respectively. The deterioration observed in spectral radius values after 5% of nodal degree based node failure was 93%, 93%, 59%, 52%, and 12%, respectively. Whereas, the spectral radius values of the low GRC valued networks, such as the USmotorway, ATTN383, EuroPGN1400, SprintN300, and DCellN2K were 4.21, 3.7, 5.02, 2.93, and 3.47, respectively. The deterioration values for the aforementioned low GRC valued networks are 15%, 3%, 18%, 5%, and 2%, respectively for the nodal degree based failures. Over again, the spectral radius based robustness estimation for hierarchical networks is misleading, where the networks with low spectral radius values show better robustness and high giant cluster size, and the network with high spectral radius values depict higher deterioration. The deterioration values for spectral radius are illustrated in Fig. 5.7.

5.3.4.6. Average Nodal Degree

The nodal degree depicts the average connectivity of the nodes within a network. Higher the average nodal degree, the more robust the network is. We observe that the networks with

high GRC values possess comparatively high average nodal degree values, such as 5.9, 4.03, 4.6, and, 2.67 for the ASN25K, ASN26K, FatTreeN2K, and ThreeTierN2K, respectively. However, the deterioration values of the aforementioned high GRC valued networks are 86%, 85%, 24%, and 68%, respectively. Alternatively, the low GRC valued networks like USmotorway, ATTN383, EuroPGN1400, and SprintN300 have 2.69, 2.54, 2.88, and 2.37 average nodal degree values, and 8%, 10%, 13%, and 6%, deterioration values, respectively. A similar trend can also be observed for deterioration values for betweenness centrality based targeted node failures depicted in Table 5.5. It can be observed from the aforementioned nodal degree and deterioration values after 5% of the node failures that the low valued GRC networks exhibit better robustness to the targeted failures as compared to the highly hierarchical networks. The deterioration trend illustrated in Fig. 5.8 also affirms the claim.

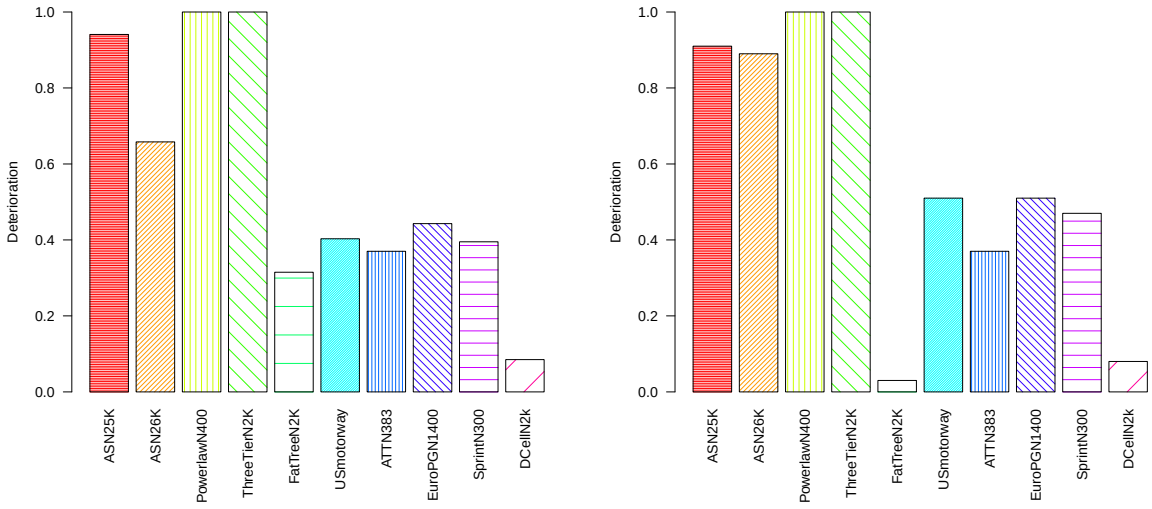


Fig. 5.6. Deterioration in Algebraic Connectivity based on: Nodal Degree (Left) and Betweenness Centrality (Right).

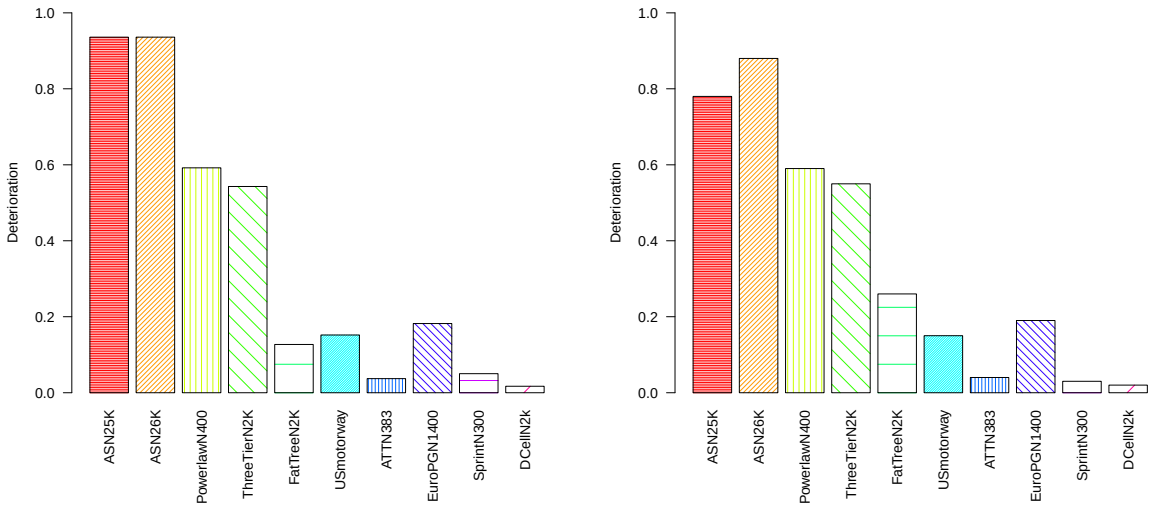


Fig. 5.7. Deterioration in Spectral Radius based on: Nodal Degree (Left) and Betweenness Centrality (Right).

Summarizing, the aforementioned results depict that the high GRC valued networks undergo more segregation and deterioration. In most of the cases, the initial metric results for the networks with higher hierarchy values show comparatively high robustness as compared to low GRC valued network. However, based on the giant cluster size and deterioration analysis, one can infer that the low GRC valued networks retain connectivity, exhibit large giant cluster size, and report low deterioration values on the contrary to the initial robustness estimation. Therefore, it can be argued that the initial robustness estimation of the hierarchical networks using most of the classical robustness measures may be misleading and inadequate.

5.3.5. Correlation between Network Hierarchy and Deterioration

To quantify the relationship between the network hierarchy and the deterioration in case of failures, we use the Pearson correlation coefficient. Fig. 5.9 illustrates the Pearson correlation coefficient between the GRC values and the deterioration for the nodal degree and betweenness

centrality based failures. As can be observed there exists a strong correlation between network hierarchy and deterioration. For most of the metrics, such as the cluster size, diameter, nodal degree, and spectral radius, the correlation value is above 80% depicting a strong correlation. The correlation value for the algebraic connectivity in case of the betweenness centrality based failures is around 70%. However, in case of the nodal degree based failures, the value of the correlation for the algebraic connectivity is 82%. Therefore, it can be inferred that the network hierarchy is closely related to deterioration in case of intentional attacks. The more hierarchical a network, the less robustness it exhibits in case of targeted attacks.

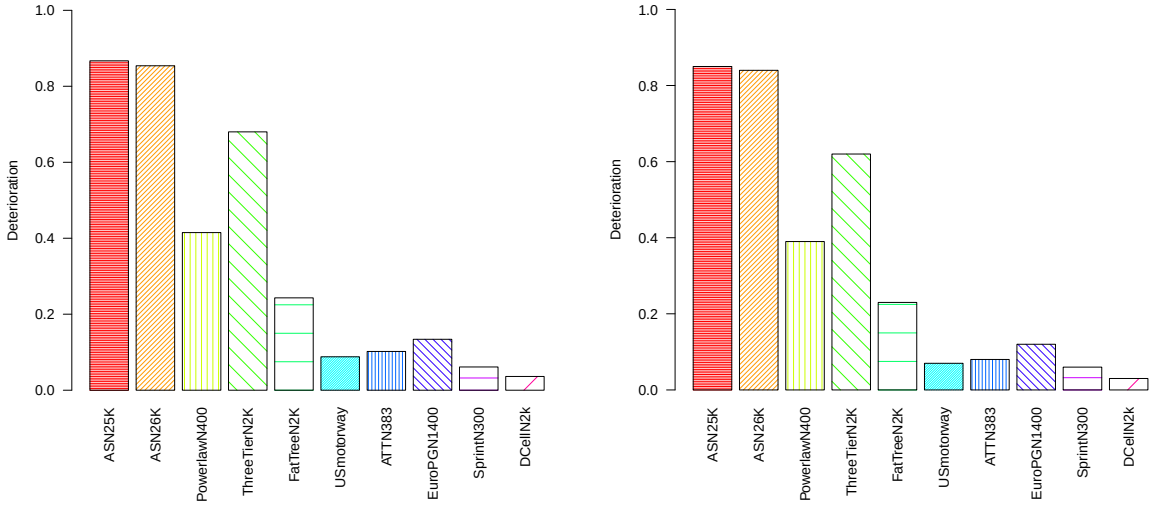


Fig. 5.8. Deterioration in Average Nodal Degree based on: Nodal Degree (Left) and Betweenness Centrality (Right).

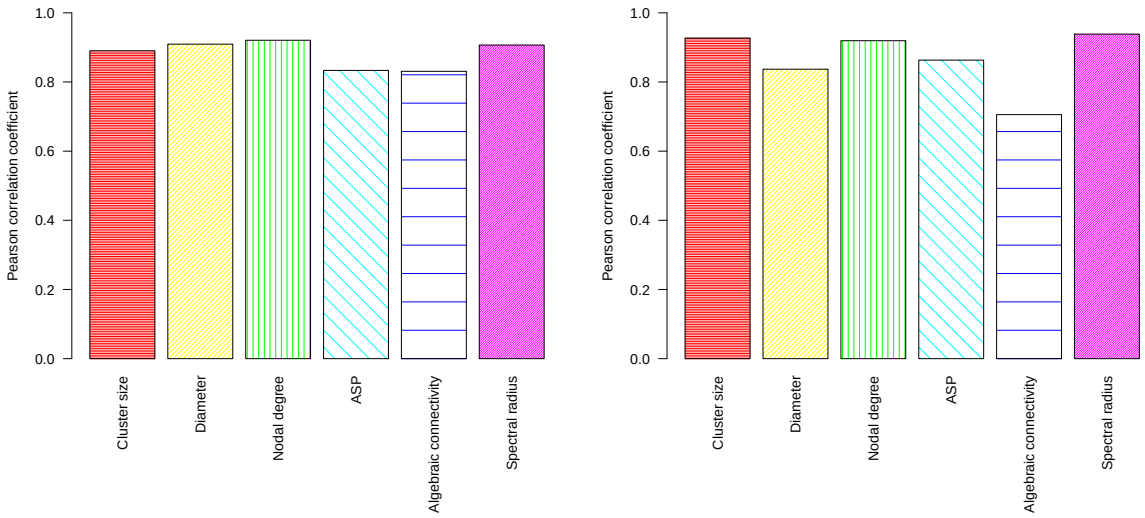


Fig. 5.9. Correlation between the Network Hierarchy and Deterioration in Classical Metrics, for Nodal Degree (Left) and Betweenness Centrality based (Right) Attacks.

5.4. References

[5.1] Strogatz, Steven H. "Exploring complex networks." *Nature* 410.6825 (2001): 268-276.

[5.2] Mones, Enys, Lilla Vicsek, and Tamás Vicsek. "Hierarchy measure for complex networks." *PLoS One* 7.3 (2012).

[5.3] Ravasz, Erzsébet, and Albert-László Barabási. "Hierarchical organization in complex networks." *Physical Review E* 67.2 (2003).

[5.4] Corominas-Murtra, Bernat, *et al*. "On the origins of hierarchy in complex networks." *Proceedings of the National Academy of Sciences* 110.33 (2013): 13316-13321.

[5.5] Manzano, Marc, Kashif Bilal, Eusebi Calle, and Samee Khan. "On the Connectivity of Data Center Networks." *IEEE Communications Letters* 17.11 (2013): 2172-2175.

[5.6] Wimberley, Edward T. Nested ecology: The place of humans in the ecological hierarchy. *JHU Press* (2009).

[5.7] Lane, David. "Hierarchy, complexity, society." *Hierarchy in natural and social sciences*. Springer Netherlands, (2006): 81-119.

[5.8] Trusina, Ala, *et al*. "Hierarchy measures in complex networks." *Physical review letters* 92.17 (2004).

[5.9] Manzano, Marc, Eusebi Calle, and David Harle. "Quantitative and qualitative network robustness analysis under different multiple failure scenarios." *UltraModern Telecommunications and Control Systems and Workshops (ICUMT)* (2011).

[5.10] Bilal, Kashif, *et al*. "On the Characterization of the Structural Robustness of Data Center Networks." *IEEE Transections on Cloud Computing* 1.1 (2013).

[5.11] Bilal, Kashif, Samee U. Khan, and Albert Y. Zomaya. "Green Data Center Networks: Challenges and Opportunities." *Frontiers of Information Technology (FIT), IEEE*, 2013.

[5.12] Costa, L., *et al*. "Characterization of complex networks: A survey of measurements." *Advances in Physics* 56.1 (2007): 167-242.

[5.13] Albert, Réka, Hawoong Jeong, and Albert-László Barabási. "Error and attack tolerance of complex networks." *Nature* 406.6794 (2000): 378-382.

[5.14] Holme, Petter, *et al*. "Attack vulnerability of complex networks." *Physical Review E* 65.5 (2002).

[5.15] Callaway, Duncan S., *et al*. "Network robustness and fragility: Percolation on random graphs." *Physical review letters* 85.25 (2000).

[5.16] Dekker, Anthony H., and Bernard D. Colbert. "Network robustness and graph topology." *Proceedings of the 27th Australasian conference on Computer science*, 2004.

[5.17] Dekker, Anthony H., and Bernard Colbert. "The symmetry ratio of a network." *Proceedings of the 2005 Australasian symposium on Theory of computing* (2005).

[5.18] Jamakovic, A., and Piet Van Mieghem. "On the robustness of complex networks by using the algebraic connectivity." *Wireless Networks, Next Generation Internet*. Springer Berlin Heidelberg (2008): 183-194.

[5.19] Li, C., *et al*. "The correlation of metrics in complex networks with applications in functional brain networks." *Journal of Statistical Mechanics: Theory and Experiment* 2011.11 (2011).

[5.20] Van Mieghem, P., *et al*. "A framework for computing topological network robustness." Delft University of Technology, Report 20101218 (2010).

[5.21] Sydney, Ali, *et al*. "Characterising the robustness of complex networks." *International Journal of Internet Technology and Secured Transactions* 2.3 (2010): 291-320.

[5.22] Manzano, Marc, *et al*. "Endurance: A new robustness measure for complex networks under multiple failure scenarios." *Computer Networks* 57.17 (2013): 3641-3653.

[5.23] Barabási, Albert-László, and Réka Albert. "Emergence of scaling in random networks." *science* 286.5439 (1999): 509-512.

[5.24] Hutcheon, Neil, and Janusz W. Bialek. “Updated and validated power flow model of the main continental European transmission network.” *IEEE PowerTech* (2013).

[5.25] The DIMES project. http://www.netdimes.org/. [5.Online; accessed 28-Sep-2013].

[5.26] Leskovec, Jure, Jon Kleinberg, and Christos Faloutsos. “Graph evolution: Densification and shrinking diameters.” *ACM Transactions on Knowledge Discovery from Data* 1.1 (2007).

[5.27] Cetinkaya, Egemen K., *et al*. “Topology connectivity analysis of Internet infrastructure using graph spectra.” *Ultra Modern Telecommunications and Control Systems and Workshop, IEEE* (2012).

[5.28] Spring, Neil, Ratul Mahajan, and David Wetherall. “Measuring ISP topologies with Rocketfuel.” *ACM SIGCOMM Computer Communication Review* 32.4 (2002): 133-145.

[5.29] Bilal, Kashif, *et al*. “Quantitative comparisons of the state-of-the-art data center architectures.” *Concurrency and Computation: Practice and Experience* 25.12 (2013): 1771-1783

[5.30] Bilal, Kashif, *et al*. “A taxonomy and survey on Green Data Center Networks.” *Future Generation Computer Systems* (Forthcoming).

[5.31] Al-Fares, Mohammad, Alexander Loukissas, and Amin Vahdat. “A scalable, commodity data center network architecture.” *ACM SIGCOMM Computer Communication Review* 38.4 (2008).

[5.32] Guo, Chuanxiong, *et al*. "Dcell: a scalable and fault-tolerant network structure for data centers." *ACM SIGCOMM Computer Communication Review* 38.4 (2008).

[5.33] Rowe, Ryan, *et al*. "Automated social hierarchy detection through email network analysis." *Proceedings of the 9th WebKDD and 1st SNAKDD workshop on Web mining and social network analysis* (2007).

[5.34] Carmel, Liran, David Harel, and Yehuda Koren. "Drawing directed graphs using one-dimensional optimization." *Graph Drawing*. Springer Berlin Heidelberg (2002).

[5.35] Krackhardt, David. "Graph theoretical dimensions of informal organizations." *Computational organization theory* 89.112 (1994): 123-140.

[5.36] Opsahl, Tore, Filip Agneessens, and John Skvoretz. "Node centrality in weighted networks: Generalizing degree and shortest paths." *Social Networks* 32.3 (2010): 245-251.

[5.37] Luque, Bartolo, and Ricard V. Solé. "Phase transitions in random networks: simple analytic determination of critical points." *Physical Review E* 55.1 (1997): 257-260.

[5.38] Manzano, Marc, Victor Torres-Padrosa, and Eusebi Calle. "Vulnerability of core networks under different epidemic attacks." *UltraModern Telecommunications and Control Systems and Workshops* (2012).

[5.39] West, Douglas Brent. Introduction to graph theory. Vol. 2. *Upper Saddle River: Prentice hall*, 2001

6. QUANTITATIVE COMPARISONS OF THE STATE OF THE ART DATA CENTER ARCHITECTURES

This paper is published in Concurrency and Computation: Practice and Experience, vol. 25, no. 12, pp. 1771-1783, 2013. The authors of the paper are Kashif Bilal, Samee U. Khan, Limin Zhang, Hongxiang Li, Khizar Hayat, Sajjad A. Madani, Nasro Min-Allah, Lizhe Wang, Dan Chen, Majid Iqbal, Cheng-Zhong Xu, and Albert Y. Zomaya.

6.1. Introduction

A data center is a pool of computing resources clustered together using communication networks to host applications and store data. Major Information and Communication Technology (ICT) components of the data center are: (a) servers and (b) network infrastructure. Conventional data centers are modeled as a multi-layer hierarchical network with thousands of low cost commodity servers as the network nodes. Data centers are experiencing exponential growth in number of hosted servers. Google, Microsoft, and Yahoo already host hundreds of thousands of servers in their respective data centers [6.1, 6.2]. Google had more than 450,000 servers in 2006 [6.3, 6.4] and the number of servers is doubling every 14 months at the Microsoft data centers [6.5].

Increased number of servers demands fault tolerant, cost effective, and scalable network architecture with maximum inter-node communication bandwidth. Another important aspect of data center design is the use of low cost commodity equipment. The server portion of data centers has experienced enormous commoditization and low cost commodity servers are used in data centers instead of high-end enterprise servers. However, the network portion of the data center has not seen much commoditization and still uses enterprise-class networking equipment

[6.6]. Increased number of servers demands high end-to-end bisection bandwidth. The enterprise-class network equipment is expensive, power hungry, and is not designed to accommodate Internet-scale services in data centers. Therefore, the use of enterprise-class equipment experiences limited end-to-end network capacity, non-agility, and creation of fragmented server pools [6.6].

Data Center Network (DCN) is typically based on the three-tier architecture [6.7]. Three-tier data center architecture is a hierarchical tree based structure comprised of three layers of switching and routing elements having enterprise-class high-end equipment in higher layers of the hierarchy [6.7, 6.8]. Unfortunately, deployment of even the highest-end enterprise-class equipment may provide only 50% of end-to-end aggregate bandwidth [6.9]. To accommodate the growing demands of data center communication, new DCN architectures are required to be designed.

Most of the internet communication in future is expected to take place within the data centers [6.10]. Many applications hosted by data centers are communication intensive, such as more than 1000 servers may be touched by a simple web search request. Communication pattern in a data center may be one-to-one, all-to-all, or one-to-all [6.11]. The major challenges in the data center network design include: (a) scalability, (b) agility, (c) fault tolerance, (d) end-to-end bisection bandwidth, (e) robustness against single point of failure, (f) automated naming and address allocation, and (g) backward compatibility.

DCN architecture is a major part of data center design acting as a communication backbone and requires extreme consideration. Numerous DCN architectures have been proposed in the recent years [6.9, 6.10, 6.12-6.18]. This paper provides a comparative study and analysis

of major DCN architectures that are proposed in recent years by implementing: (a) proposed network architectures, (b) customized addressing scheme, (c) customized routing schemes, and (d) different network traffic patterns.

We have implemented the fat-tree based architecture [6.9], recursively defined architecture [6.12, 6.13], and legacy three-tier DCN architecture to compare the performance under six different network traffic patterns. For the fat-tree DCN architecture, we implemented the *n*-pod based network interconnection design, customized network addressing scheme for servers and switches at different levels, and customized two-level routing algorithm. For the recursive based DCell DCN architecture, we applied customizable *n*-level network architecture (up to four levels scalable for more than 3.6 million servers), a generic network addressing scheme, and the DCell routing algorithm. DCell routing algorithm [6.12] returns a series of nodes (e.g., [001] [010]) as intermediate hops between source to destination. We formulated an algorithm to find the network address based end-to-end path and implemented source based routing in the ns-3 simulator. Moreover, the DCell routing algorithm pseudocode had some missing information for implementation and working. For the legacy three-tier DCN architecture, we implemented customizable network architecture as reported in [6.7, 6.8]. We used the Equal Cost Multi-Path (ECMP) [6.19] routing to obtain realistic results for the three-tier DCN architecture. Presumably, it is the very first comparative study of DCN architectures employing implementation and simulation techniques.

A simple simulation analysis introduced in this paper allows us to compare the behavior and performance of the considered DCN architectures under different workload and network conditions. The DCN architectures used in the analysis [6.9, 6.12] have been implemented on a

small-scale system, with 20 servers in the case of DCell model [6.12] and 10 machines in the fat-tree model [6.9]. The simulation analysis may be considered as a general testbed for realistic networks with large number of hosts and various communication and traffic patterns. The analysis may also be used for the "green data centers'' for designing energy-efficient communication protocols in DCN [6.20 – 6.26].

6.2. Simulations and Comparative Study

6.2.1. Environment

The main aim of the empirical simulation analysis presented in this section is to provide a comprehensive insight of different DCN architectures in a realistic manner. Three DCN core architectural models, namely: (a) the legacy three-tier architecture, (b) fat-tree based architecture, and (c) recursively build DCell architecture, have been used for the simulation of the multi-level DCN performance. We used ns-3 discrete-event network simulator for implementing the considered DCN architectures [6.32]. The ns-3 simulator allows modeling of various realistic scenarios. The most important salient features of the ns-3 simulator are: (a) implementation of real IP addresses, (b) BSD socket interface, (c) multiple installations of interfaces on a single node, (d) real network bytes contained in simulated packets, and (e) packet traces can be captured and analyzed using tools like Wireshark. In this work, the DCN architectures uses: (a) the customized addressing scheme and (b) the customized routing protocols that strongly depend on the applied addressing scheme (e.g., [6.9]). Therefore, ns-3 deemed as the most appropriate network simulator for our work. One of the major drawbacks of using the ns-3 simulator is a lack of the network switch module in the ns-3 library. Moreover, the conventional Ethernet protocol cannot be implemented in ns-3. Therefore, we configured Point-To-Point links for the connection

of switches and nodes. Moreover, we also implemented customized routing protocols for the DCN architectures in ns-3. All of our implementation will be made publically available for researchers and students.

6.2.2. Implementation Details

The considered DCN architectures have been implemented by using the multiple network interfaces at each node as required. We implemented the three-tier architecture with an oversubscription ratio of 4:1 at the access layer and 1.5:1 at the aggregate layer. We used the interconnection architecture for three-tier architecture as reported in [6.7, 6.8], and used ECMP routing for enhanced performance, as available in the high-end switches. In the case of fat-tree based topology, the primary and secondary routing tables are generated dynamically and are based on the number of pods. The realistic IP addresses have been assigned to all of the nodes within the system and linked to appropriate lower layer switches. Three layers of switches have been created, interconnected, and populated with primary and secondary routing tables. We have tailored the general simulator model by extending it with an additional routing module for processing two layered based primary and secondary routing tables in ns-3. A simulation representation of 8-pod fat-tree is shown in Fig. 6.1.

In the DCell architecture, the DCell routing protocol is implemented to generate the end-to-end path at the source node. We have specified a scalable addressing protocol for this model. The DCell routing lacks the generic protocol description and a specific routing scenario is discussed by authors. Moreover, DCell routing does not take the Internet Protocol (IP) addressing scheme into consideration. We generalized and implemented the routing protocol,

which is now fully compatible with the IP. We implemented the source based routing procedure to route the packets from the source to destination using the IP.

We found some important details missing in the DCell routing protocol presented in Section 4.1 of [6.12]. In the function GetLink, the authors state that if (sk-m < dk-m), then the link interconnecting both of the sub-DCells can be found as "([sk-m, dk-m -1], [dk-m, sk-m])". The "else clause" for the aforementioned "if statement" is missing, which makes the routing algorithm incomplete and erroneous. We formulated the missing "else clause" to complete the algorithm. That is to say that if (sk-m >= dk-m), then the interconnection link can be found as "([sk-m, dk-m], [dk-m, sk-m -1])". Moreover, the intermediate path between nodes 021 and 121, presented in Section 4.1 (Theorem 4) of [6.12] has a typographical error that may mislead and confuse readers. The underlined node within the path ([0,2,1], [0,2,0], [1,0,0], [0,0,0], [1,0,0], [1,0,1], [1,2,0],[1,2,1]) should be [0,0,1] instead of [1,0,0]. For reference, a simulation representation of 3 level3 DCells is shown in Fig. 6.1.

6.2.3. Traffic Patterns

Benson *et al.* observed an on-off network traffic behavior within data centers. The network traffic logs collected at 19 various data centers provided evidence of the on-off network traffic and short-lived traffic bursts [6.33]. To generalize our simulation results, we used six different network traffic patterns to evaluate the DCN architectures for one-to-one, one-to-many, and all-to-all communications, namely: (a) uniform random (b) exponential random, (c) one-to-one for one second (*1-1-1*), (d) one-to-one for random time interval (*1-1-R*), (e) one-to-many for one second *(1-M-1),* and (f) one-to-many for random time interval (*1-M-R*).

In uniform random and exponential random traffic generation scenarios, every node within the data center communicates with some other arbitrarily chosen node. Inter-node communication occurs at random time intervals following the uniform random distribution and exponential random distribution, respectively. In the *1-1-1* traffic generation pattern, every node within the network communicates with some other randomly chosen node for an on period of one second. That is to say that the sender nodes send the data at a Constant Bit Rate (CBR) for flow duration of one second. For the *1-1-R* traffic, the sender nodes send the CBR data in an on period for a randomly chosen time interval between 0.1 to 5.0 seconds, followed by an off period of random time interval.

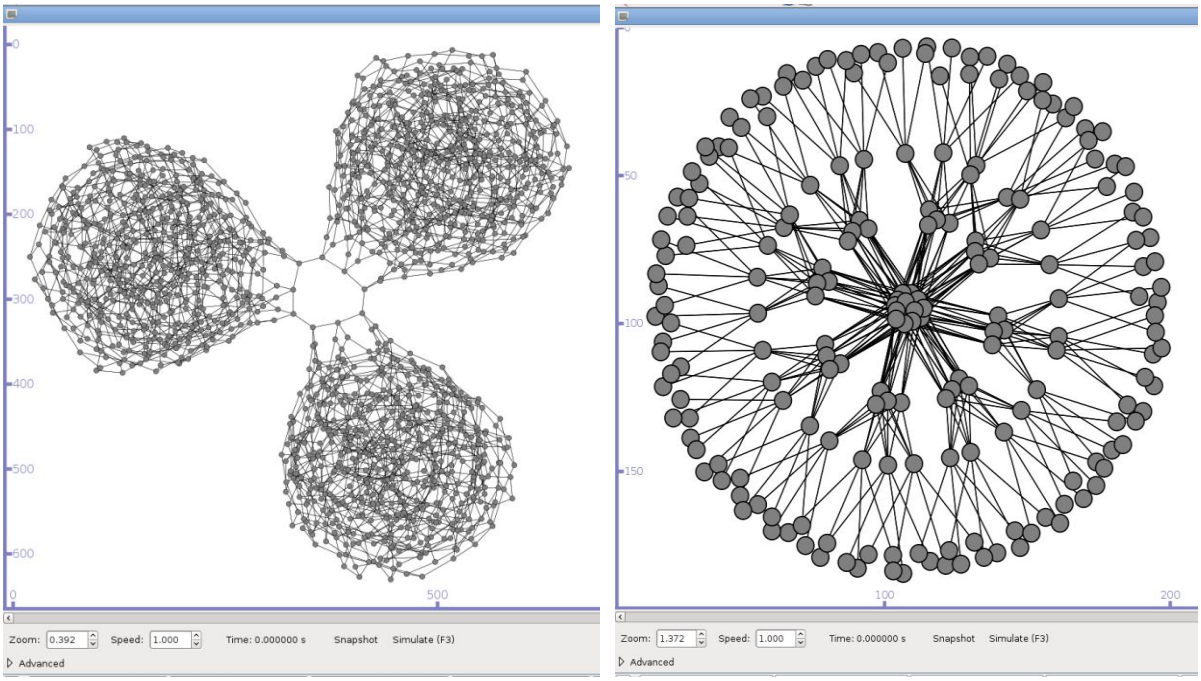


Fig. 6.1. 3 DCell3 and 8-pod FatTree NS-3 Simulation.

In the *1-M-1* traffic generation scenario, a single node communicates with *n* other arbitrarily chosen nodes for an on period of one second duration. The value for n is also chosen at random from a range of [1-10]. In the *1-M-R* scenario, a single node communicates with *n*

other nodes for an on period of randomly chosen duration. In one-to-many network scenarios, the number of sender nodes is around 1/8 of the network size.

We simulated the aforementioned four traffic generation scenarios with two different data rates for the CBR communication, namely: (a) 1Mbps and (b) 10Mbps. In the 10Mbps data rate, each sender sends 10Mb data to the receiver within a one second time slot. Similar analogy will hold for the 1Mbps data rate.

6.2.4. Comparative Analysis

We have simulated all of the DCN architectures under the six scenarios discussed in Section 3.3. The performances of the considered architectural models have been verified by using the following criteria:

Average packet delay: Average packet delay in the network is calculated using Eq. (6.2).

$$D_{agg} = \sum_{j=1}^{n} d^{j}, \tag{6.1}$$

$$D_{avg} = \frac{D_{agg}}{n}, \tag{6.2}$$

where D_{agg} calculated in Eq. (1) is the aggregate delay of all of the received packets and d_j is the delay of packet j, n is total number of packets received in the network, whereas D_{avg} is average packet delay.

Average network throughput: Average network throughput is calculated using the Eq. 6.3.

$$\frac{\tau = \left(\sum_{i=1}^{n} (P_i) \times \delta \right)}{D_{agg}}, \tag{6.3}$$

where τ is the throughput, P_i is the i^{th} received packet, δ is the size of the packet (in bits), and D_{agg} is the aggregate packets delay.

The parameters used in the simulation of the fat-tree, DCell, and three-tier DCN are documented in Table 6.1, Table 6.2, and Table 6.3, respectively. Simulations were performed by varying the aforementioned parameters under six different traffic scenarios to achieve results in respective topologies. Network topologies with different number of nodes ranging from 16 to 4096 nodes were created for the respective DCN architectures for every traffic pattern. Around 74 different simulation scenarios were created for each of the DCN architecture, resulting in 222 different configurations. The simulation results for the network throughput and average packet delay are shown in Figures 6.2 through 6.9. The FAT, DCell, and 3T in the chart legend represent fat-tree, DCell, and three-tier DCN architectures, respectively.

The simulation results depict a steady behavior for DCN architectures under various traffic patterns and data rates. Because the network throughput is inversely proportional to average packet delay, large packet delays result in small throughput. The throughput and average packet delay for uniform random and exponential random traffic distribution is shown in Fig. 6.6. Figures 6.3 through 6.6 report the results for *1-1-1*, *1-1-R*, *1-M-1*, and *1-M-R* traffic patterns for a data rate of 1Mbps, respectively. Figures 6.7 through 6.10 show the results for 10Mbps.

It can be observed in Figures 6.2 through 6.10 that the fat-tree DCN architecture outperforms the DCell and three-tier architecture in term of throughput and packet delay. The three-tier architecture performance is almost equivalent to that of the fat-tree architecture with a very little difference in the average throughput. The DCell architecture outperforms the fat-tree and three-tier architecture for small network topologies but as the number of nodes within the network is increased, the DCell architecture experiences degradation in the network throughput and exhibits increased average packet delay.

The reason for the steady performance of the fat-tree architecture is the inherent network topology. A large number of network switches are structured in such a way so as to provide more end-to-end bandwidth for better and steady-state performance.

Although the performance of three-tier architecture seems similar to that of the fat-tree architecture, the performance is achieved at a much higher cost. Some important aspects for the better performance of three-tier architecture are: (a) The three-tier architecture uses costly high-end network equipment at the higher layer, (b) the ECMP routing also contributes to the better performance, and (c) the oversubscription ratio of 4:1 and 1.5:1 at the access layer and aggregation layer, respectively. The actual oversubscription ratio may be much higher and may vary form a data center to a data center at the access and aggregation layers. The data provided in [6.33] depicts a great variety in oversubscription ratios at the different layers of the three-tier data centers.

The performance of the DCell architecture depicts a strong dependency on the network size. We illustrate this phenomenon through Fig. 6.1. All of the *inter-DCell* network traffic must pass through the network link connecting both of the *DCells* leading to increased network

congestion, packet delay, and packet loss. Smaller network topologies experience larger throughput because the network traffic load on the *inter-DCell* link is low and the links serve lesser number of nodes. For larger topologies, such as in our case of the network with 4096 nodes, each link connecting two $DCell_3$ experience an oversubscription ratio of 256:1. That obviously decreases the throughput for larger networks. Another reason for the throughput degradation in the DCell is the number of intermediate hops between the sender and the receiver. DCell routing is not a shortest path routing algorithm, and for large network topologies, the number of intermediate hops may be as large as 2^{k+1}-1, without considering the switching in the $DCell_0$ as a hop [6.12]. Intermediate hops including the switching in the $DCell_0$ as a hop may result in more than 20 intermediate hops.

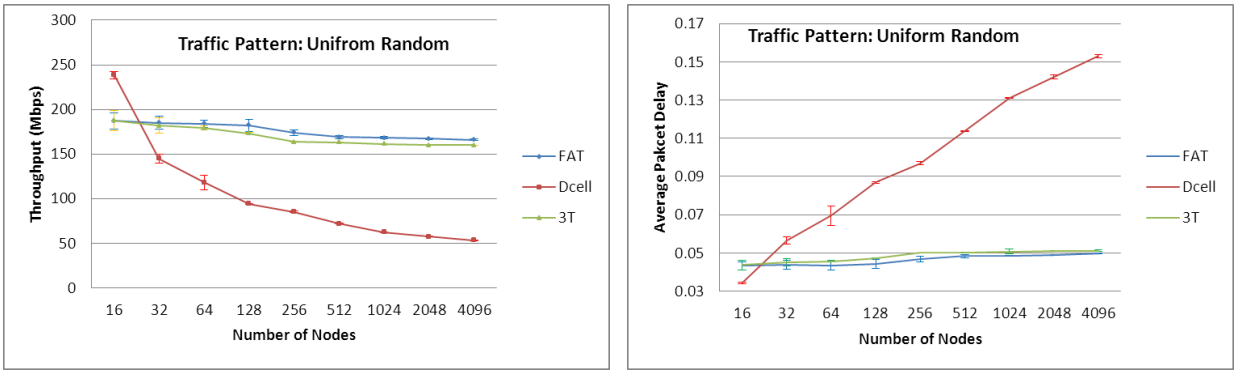
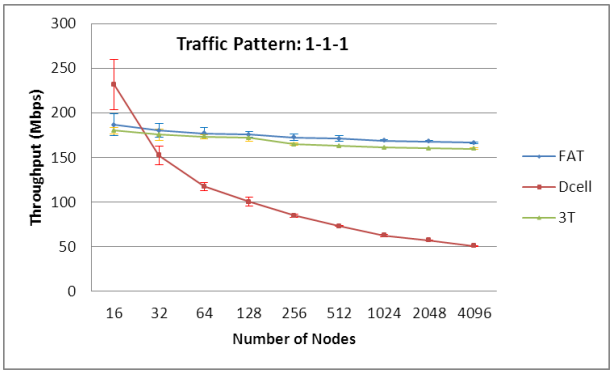


Fig. 6.2. Throughput and Average Packet Delay Using Uniform Random Traffic Distribution.



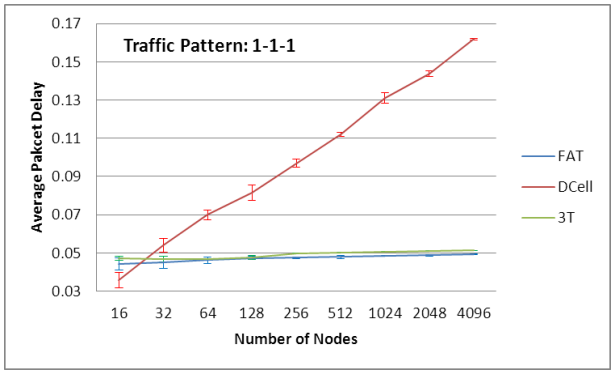
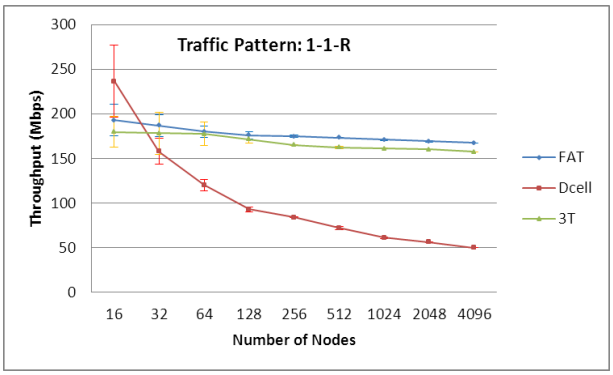


Fig. 6.3. Throughput and Average Packet Delay for 1-1-1 Traffic Pattern with 1Mbps Data Rate.



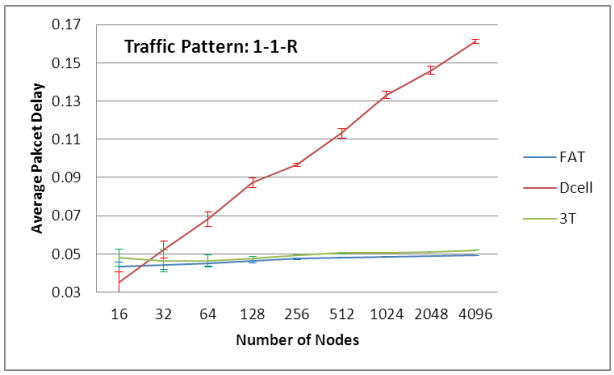
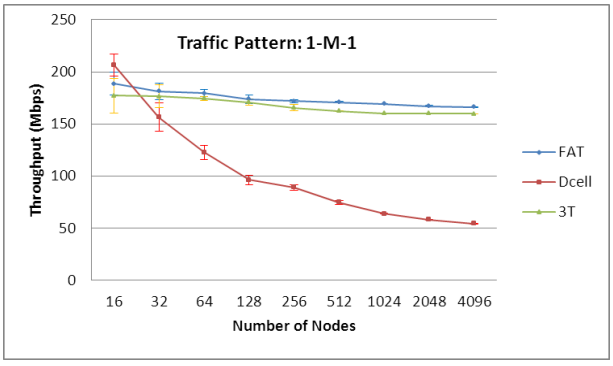


Fig. 6.4. Throughput and Average Packet Delay for 1-1-R Traffic Pattern with 1Mbps Data Rate.



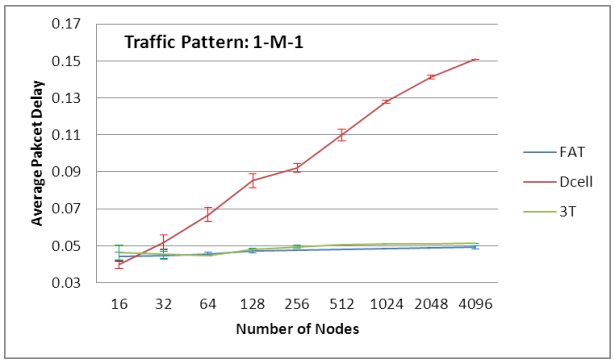
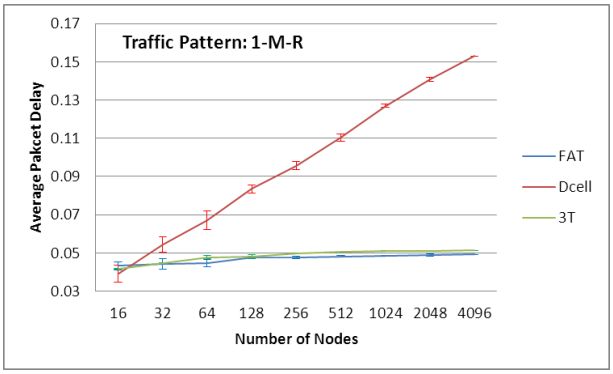


Fig. 6.5. Throughput and Average Delay for 1-M-1 Traffic Pattern with 1Mbps Data Rate.



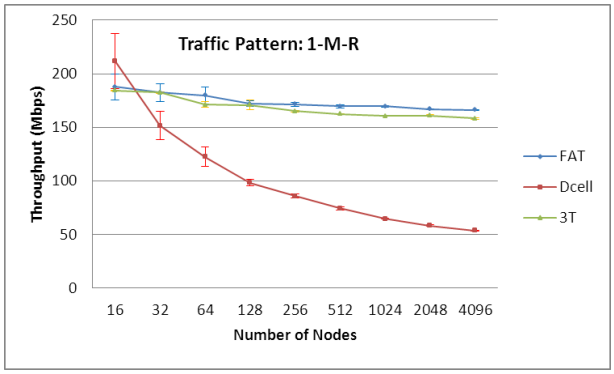
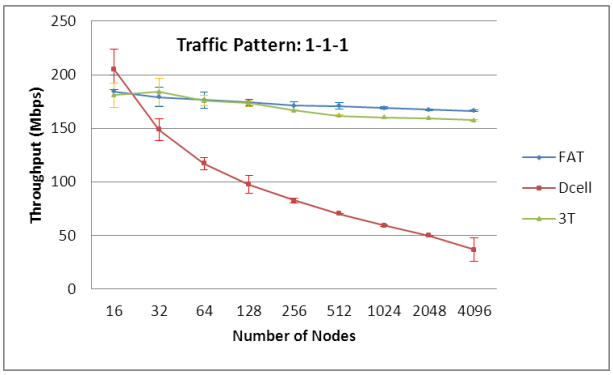


Fig. 6.6. Throughput and Average Delay for 1-M-R with 1Mbps Data Rate.



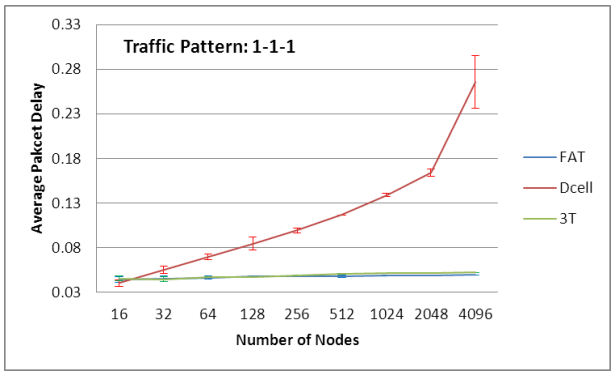
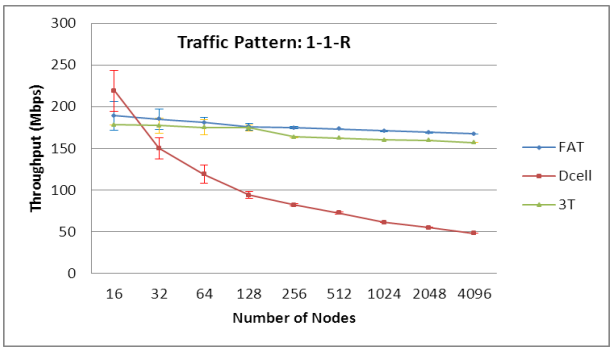


Fig. 6.7. Throughput and Average Delay for 1-1-1 Traffic Pattern with 10Mbps Data Rate.



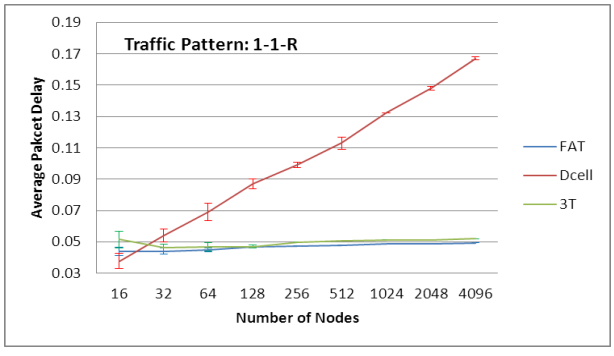
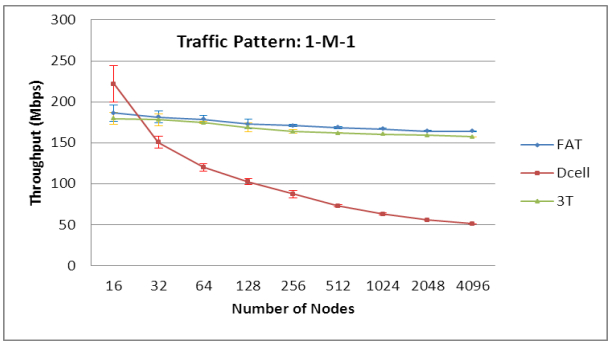


Fig. 6.8. Throughput and Average Delay for 1-1-R Traffic Pattern with 10Mbps Data Rate.



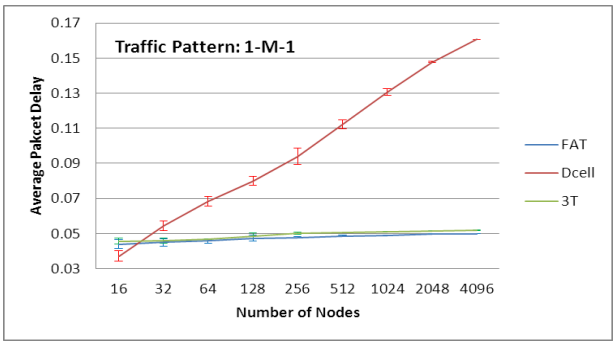
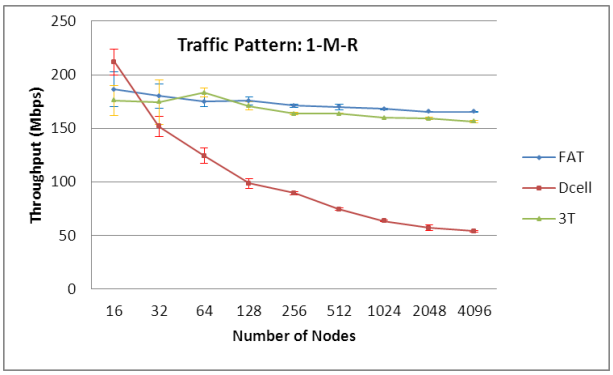


Fig. 6.9. Throughput and Average Delay for 1-M-1 Traffic Pattern with 10Mbps Data Rate.



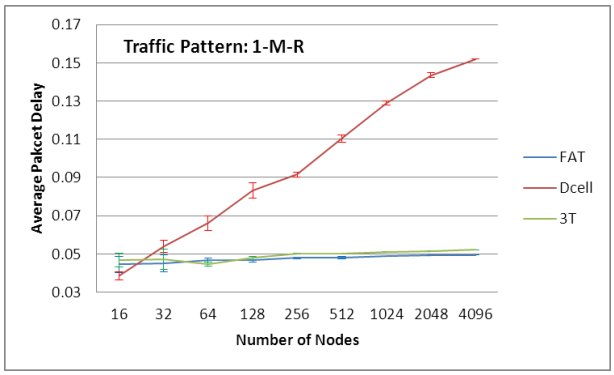


Fig. 6.10. Throughput and Average Delay for 1-M-R Traffic Pattern with 10Mbps Data Rate.

Table 6.1. Simulation Parameters for the FatTree.

number of pods	4 – 72
number of nodes	16 – 93312
simulation running time	10 – 1000 seconds
packet size	1024 bytes
routing algorithm	two-level routing protocol

Table 6.2. Simulation Parameters for the DCell.

number of levels	0 – 3
number of nodes in $DCell_0$	2 – 8
total nodes in the DCell	16 – 100000
simulation running time	10 –1000 seconds
packet size	1024 bytes
routing algorithm	DCellRouting

Table 6.3. Simulation Parameters for the ThreeTier DCN Architecture.

number of modules	4 – 170
nodes connected with each access layer switch	8
oversubscription ratio at access layer	4:1
oversubscription ratio at aggregate layer	1.5:1
simulation running time	10 – 1000 seconds
packet size	1024 bytes
routing algorithm	ECMP global routing

The simulation results reveal that the performance of fat-tree DCN architecture is independent of the network size. Alternatively, performance of the DCell architecture is heavily dependent on the network size. The performance of the three-tier architecture is dependent on physical topology and oversubscription ratio at different network layers. We hope that our

through investigation of the most commonly used data center architectures will spark further investigation in developing scalable data center architectures.

6.3. References

[6.1] Carter A. 2007. Do it green: media interview with Michael Manos. http://edge.technet.com/Media/Doing-IT-Green/, accessed, Feb. 20, 2012.

[6.2] Rabbe L. 2006. Powering the Yahoo! network. http://yodel.yahoo.com/2006/11/27/powering-the-yahoo-network/, accessed February 20, 2012.

[6.3] Arnold S. 2007. Google Version 2.0: the Calculating Predator. Infonortics Ltd.

[6.4] Ho T. 2007. Google architecture. http://highscalability.com/google-architecture, accessed February 20, 2012.

[6.5] Snyder J. 2007. Microsoft: datacenter growth de fi es Moore ' s law. http://www.pcworld.com/article/id,130921/article. html, accessed February 20, 2012.

[6.6] Sengupta S. Cloud data center networks: technologies, trends, and challenges. *ACM SIGMETRICS Performance Evaluation Review* 2011; 39 (1):355 – 356.

[6.7] Kliazovich D, Bouvry P, Khan SU. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*.

[6.8] Cisco Data Center Infrastructure 2.5 Design Guide. Cisco press, March 2010.

[6.9] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture. *In Proceedings of the ACM SIGCOMM 2008 conference on Data communication (Seattle, WA)*. 2008; 63 – 74.

[6.10] Mysore RN, Pamboris A, Farrington N, Huang N, Miri P, Radhakrishnan S, Subramanya V, Vahdat A. Portland: a scalable fault-tolerant layer 2 data center network fabric. *In Proceedings of the ACM SIGCOMM 2009 conference (Barcelona, Spain)*. 2009; 39 – 50.

[6.11] Chen K, Hu CC, Zhang X, Zheng K, Chen Y, Vasilakos AV. Survey on routing in data centers: insights and future directions. *IEEE Network* 2011; 25 (4):6 – 10.

[6.12] Guo, C, Wu H, Tan K, Shi L, Zhang Y, Lu S. Dcell: a scalable and fault-tolerant network structure for data centers. *ACM SIGCOMM Computer Communication Review* 2008; 38 (4):75 – 86.

[6.13] Guo C, Lu G, Li D, Wu H, Zhang X, Shi Y, Tian C, Zhang Y, Lu S. BCube: a high performance, server-centric network architecture for modular data centers. *In Proceedings of the ACM SIGCOMM 2009 conference (Barcelona, Spain)*. 2009; 63 – 74.

[6.14] Greenberg A, Hamilton JR, Jain N, Kandula S, Kim C, Lahiri P, Maltz D, Patel P, Sengupta S. VL2: a scalable and flexible data center network. *In Proceedings of the ACM SIGCOMM 2009 conference (Barcelona, Spain)*. 2009; 51 – 62.

[6.15] Li D, Guo C, Wu H, Tan K, Zhang Y, Lu S. FiConn: using backup port for server interconnection in data centers. *In Proceedings of the IEEE INFOCOM*. 2009; 2276 – 2285.

[6.16] Wang G, David G, Kaminsky M, Papagiannaki K, Eugene T, Kozuch M, Ryan M. c-Through: part-time optics in data centers. *In Proceedings of the ACM SIGCOMM 2010 conference (New Delhi, India)*. 2010; 327 – 338.

[6.17] Farrington N, George P, Sivasankar R, Hajabdolali B, Vikram S, Yeshaiahu F, George P, Vahdat A. Helios: A hybrid electrical/optical switch architecture for modular data centers. *In Proceedings of the ACM SIGCOMM 2010 conference (New Delhi, India).* 2010; 339 – 350.

[6.18] Abu-Libdeh H, Costa P, Rowstron A, O ' Shea G, Donnelly A. Symbiotic routing in future data centers. *In Proceedings of the ACM SIGCOMM 2010 conference (New Delhi, India)*. 2010; 51 – 62.

[6.19] Hopps C. 2000. Analysis of an equal-cost multi-path algorithm. RFC 2992, Internet Engineering Task Force.

[6.20] Bilal K, Khan SU, Kolodziej J, Zhang L, Hayat K, Madani SA, Min-Allah N, Wang L, Chen D. (Forthcoming). A survey on Green communicati ons using Adaptive Link Rate. *Cluster Computing 2012a*. DOI: 10.1007/s10586-012-0225-8

[6.21] Bilal K, Khan SU, Kolodziej J, Zhang L, Hayat K, Madani SA, Min-Allah N, Wang L, Chen D. A comparative study of data center network architectures. *In Proceedings of 26th European Conference on Modelling and Simulation. Koblenz: Germany*; 2012b.

[6.22] Bianzino P, Chaudet C, Rossi D, Rougier J. A survey of green networking research. *Communications Surveys and Tutorials, IEEE* 2012; 14 (1):3 – 20.

[6.23] Zeadally S, Khan SU, Chilamkurti N. Energy-efficient networking: past, present, and future. *The Journal of Supercomputing* 2012; 62 (3):1093 – 1118.

[6.24] Khan SU, Zeadally S, Bouvry P, Chilamkurti N. Green networks. *The Journal of Supercomputing*

[6.25] Khan SU, Wang L, Yang L, Xia F. Green computing and communications. *The Journal of Supercomputing*.

[6.26] Wang L, Khan SU. Review of performance me trics for green data centers: a taxonomy study. *Journal of Supercomputing*. DOI: 10.1 007/s11227-011-0704-3

[6.27] Mahadevan P, Sharma P, Banerjee S, Ranganathan P. Energy aware network operations. *INFOCOM Workshops 2009, IEEE*. 2009; 1 – 6.

[6.28] Leiserson, CE. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers* 1985; 34 (10):892 – 901.

[6.29] Zhang Y, Su A, Jiang G. Understanding data center network architectures in virtualized environments: a view from multi-tier applications. *Computer Networks* 2011; 55 (9):2196 – 2208.

[6.30] Popa L, Ratnasamy S, Iannaccone G, Krishnamurthy A, Stoica I. A cost comparison of datacenter network architectures. *In Proceedings of the 6th International Conference (Philadelphia, Pennsylvania)*. 2010; 1 – 16.

[6.31] GyarmatiL,Trinh T.How can architecture help to reduce energy consumption in data centernetworking? *In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (Passau, Germany)*. 2010; 183 – 186.

[6.32] ns-3. 2012. http://www.nsnam.org/, accessed February 21, 2012.

[6.33] Benson T, Anand A, Akella A, Zhang M. Understanding data center traffic characteristics. *SIGCOMM Computer Communication Review* 2010; 40 1:92 – 99.

7. THERMAL-AWARE RESOURCE ALLOCATION: TOWARDS DEVELOPING GREENER CLOUD COMPUTING SCHEDULERS

This paper is submitted to *Journal of Parallel and Distributed Computing*. The authors of the paper are Kashif Bilal, Saif U. R. Malik, and Samee U. Khan.

7.1. Introduction

Cloud computing is an emerging paradigm that allows a shared pool of resources, such as networks, servers, storages, applications, and services to be accessed conveniently and on-demand [7.1]. Moreover, the resources can be rapidly provisioned or released with a minimal management effort or service provider interaction. Data Center (DC) being the architectural foundation of the cloud plays a vital role in the economic and operational success of the cloud computing paradigm [7.2]. Furthermore, DCs constitute the communication backbone of the cloud and are of paramount importance towards the system integrity.

The DCs have a real need for tens to hundreds of Gbps of bandwidth and a deterministic QoS that is satisfied by thousands of interconnected servers [7.3, 4-6]. Moreover, to improve the services for the high performance computing applications, in the recent years, DCs have been increasingly deployed. To accommodate the ever-increasing user and application demands, the DCs are growing exponentially in the number of hosted servers [7.7]. The concerns over the environmental impact, energy needs, and electricity cost of the DCs are escalating [7.8]. Many ICT giants, such as Google, Yahoo, and Microsoft have already hosted hundreds of thousands of servers in their respective DCs. Based on the energy consumption of Google DC, a report suggested that Google was possibly running about 900,000 servers in 2010 [7.9]. The total annual percentage electricity consumption of networking devices in US alone contributed to

0.07% in the year 2000 that is more than 6 TWh (Tera-Watt/hr) [7.10, 7.11]. In the year 2007, Japan, Italy, and UK reported the percentage energy consumption of 04%, 01%, and 0.7%, respectively [7.12, 7.13].

Besides being a massive consumer of power, the ICT sector is also liable for the emission of Green House Gases (GHG) that is a major contributor in global warming. In a report [7.14], the ICT sector is declared responsible for 12% of the total emission of CO2 by the year 2020. The servers consume 80% of the total electricity usage, while the network devices and storages account 10% [7.15]. The aforesaid is due to the fact that serves are the most frequently used entity of DCs. The discussion above ratifies the apparent need and impetus for the energy efficient networking in DCs. Because of the increasing demands of energy from the Information and Communication Technology (ICT) sector and the alarming consumption of natural resources, the topic of energy-efficient ICT has gained significant importance in the recent years [7.8].

To deliver the specified level of performance, the number of computational devices put in use at all levels of DC has significantly increased. As a result, the rate at which the heat is emitted by the devices has also increased. In the said perspective, the cost to stabilize the temperature in the DC has drastically increased and become almost equal to the cost of operating computational systems. The imbalance heat formations within a DC can create a hotspot that may cause servers to throttle down, increasing the possibility of failure. The cost to stabilize the temperature in the DC has drastically increased and become almost equal to the cost of operating computational systems. A thermal management policy can have many benefits in DC architecture. One of the major benefits is the reduction in cooling cost. In a typical DC

specification, the annual electricity cost of cooling alone is $4-8 M that includes purchasing and installing the air conditioning units [7.16]. Through intelligent thermal management, such a hefty cost can be lower down to $1-3 M, as advocated in [7.16]. Moreover, an intelligent thermal management can increase hardware reliability and improve operational efficiencies [7.17, 7.18, 7.19].

The topic of thermal aware scheduling in DCs is approached from various dimensions by different research communities. The increasing cost of energy consumption, including the cost of cooling down the DC, calls for new strategies to improve the energy efficiency in DCs. Several strategies have been proposed, such as [7.20-7.22] that discussed thermal-aware resource allocation strategies to minimize the energy consumption in DCs. In this paper, we analyze a real DC workload obtained from Center of Computational Research (CCR), State University of New York at Buffalo. We perform the thermal analysis of the aforesaid workload using three statistical techniques: (a) Mean Procedure, (b) Correlation Procedure, and (c) Vector Autoregressive Moving-average processes with Exogenous Regressors (VARMAX) model. The statistical techniques are used to examine the thermal behavior and pattern displayed by the servers when jobs are allocated. Moreover, the statistical techniques are also used to define the radius of ambient effect as a result of job allocation.

The results and findings from the workload analysis are used to investigate the thermal dynamics of the DC using five scheduling heuristics: (a) First Come First Serve (FCFS), (b) Shortest Job First (SJF), (c) Longest Job First (LJF), and (d) Thermal-aware Scheduling Algorithm [7.24], and (e) Genetic Algorithm (GA) based scheduling [7.23]. We choose three classical heuristics and two thermal-aware heuristics. Our aim in selecting the scheduling

heuristics is to highlight the improvements of thermal-aware heuristics over the classical heuristics. When evaluating the heuristics, we use the results from our workload analysis to measure the effect of above heuristics on the thermal dynamics of the servers. The results reveal that the heuristics are inefficient in maintaining thermal balance among various pods in the DC. The nature of the workload and the comparison results motivated us to propose a new scheduling heuristic, Thermal Aware Resource Allocation (TARA). In TARA, the jobs are allocated to the servers considering the thermal signatures of the servers and ambient effect on other servers. The results from our simulation revealed that TARA can achieve better results and thermal balances among the pods. The detailed discussion and analysis of the heuristics along with the results are discussed in later sections. The contributions of the paper can be summarized as follows:

- detailed thermal analysis of job allocation on a real DC workload;
- using statistical techniques, Mean Procedure, Correlation Procedure, and VARMAX model, to investigate the thermal impact of job allocation to servers and ambient effect on other related servers;
- proposing a scheduling heuristic, named as TARA, based on the findings and results from the workload analysis that can attain thermal uniformity in the DC.

7.2. Thermal Analysis

In this section we discuss the tools and techniques that are used to analyze the thermal dynamic of DC. The jobs and the logs from the CCR dataset are used as an input for our simulation of the thermal aware strategies. The purpose of the analysis is to find out the thermal behavior and patterns exhibit by the servers when the task allocation and execution is performed.

Before going deep into the details of the tools and techniques used for the analysis, we briefly discuss some details of the workload in the following section.

7.2.1. Workload Analysis

The traces from CCR represent a one-month workload with more than 22,000 (127,000 tasks) jobs. All jobs submitted to the CCR are logged for a period of a month from 20 Feb. 2009 to 22 Mar. 2009. The data center had 1045 distinct dual core servers. A server was based on the Dell 1056 PowerEdge SC1425 processor with 3.0 GHz speed, running x86-64 Linux operating system. The CCR data center was organized into 33 pods and each pod had 32 servers. The peak load in CCR data center exceeds the available resources over the course of time. In the said perspective, the jobs have to wait in the queue for the execution. A job can be comprised of several tasks and may require more than one processor to execute. The nature of the jobs in CCR dataset is heterogeneous that implies every job has a different execution time and number of processors required. Moreover, the job arrival rate in the workload is not uniform. In the said perspective, the resources of a DC can be underutilized, fully utilized, or over utilized, depending on the arrival rate of the jobs at a specific time interval.

The jobs can be classified into different categories based on the number of CPUs required and execution time. We classified jobs as: (a) thin, jobs that require a single CPU for the execution, (b) thick, where the numbers of CPUs required are more than one, (c) short, where jobs have execution time less than an hour, and (d) long, where the execution time of the job is more than an hour. The analysis revealed that 79% of the overall jobs submitted to CCR DC belonged to “thin” category and only 21% belonged to “thick” category. Moreover, 50% of the jobs belonged to “short” category and the rest of the jobs belonged to “long” category of jobs.

The workload characterization highlights some of the important facts, such as the job arrival rate and size of the jobs that makes the thermal analysis and prediction a complex task. The thermal analysis in presence of the aforementioned uncertainties is a challenging task. Moreover, the spatial information of the data center was unavailable in the workload that significantly increased the complexity of the analysis. We used statistical techniques, such as: (a) Mean Procedure, (b) Correlation Procedure (specifically Pearson correlation coefficient), and (c) Vector Autoregressive Moving-average processes with Exogenous Regressors (VARMAX) model, to investigate the thermal behavior displayed by the servers as a result of job allocation. The SAS tool is used for the implementation of the strategies and for the generation of the graphs. The purpose of deploying statistical techniques is to: (a) capture the thermal dynamics of the DC, (b) the thermal effect of one server on the others, and (c) on the data center environment, as a result of job allocation. The details of the techniques are discussed in the following sections.

7.2.2. Mean Procedure

The Mean Procedure computes the summary statistics of numeric variables for all the observations in a dataset [7.49]. The Mean Procedure is a basic procedure within the BASE SAS tool [7.50]. To apply the technique in our workload thermal analysis, we categorized the servers into various states, such as busy (represented by a 1) and idle (represented by a 0) state. A busy processor emits more heat, and impacts the ambient temperature and thermal signature of the nearby (within the thermal radius) nodes.

Table 7.1. Results from Mean Procedure.

B0102	Frequency	Percent	Cumulative Frequency	Cumulative Percent
00	150	21.68	150	21.68
01	39	5.64	189	27.31
10	34	4.91	223	32.23
11	469	67.77	692	100.00

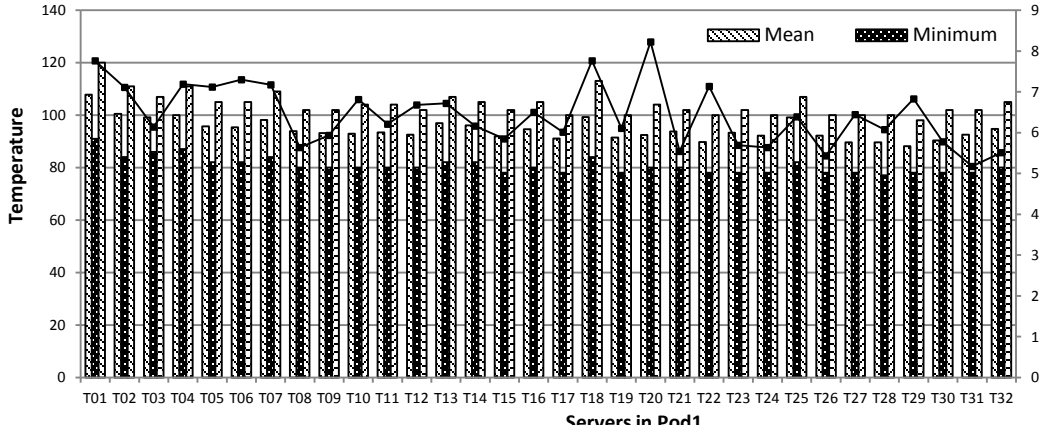


Fig. 7.1. Simple Statistic Measures.

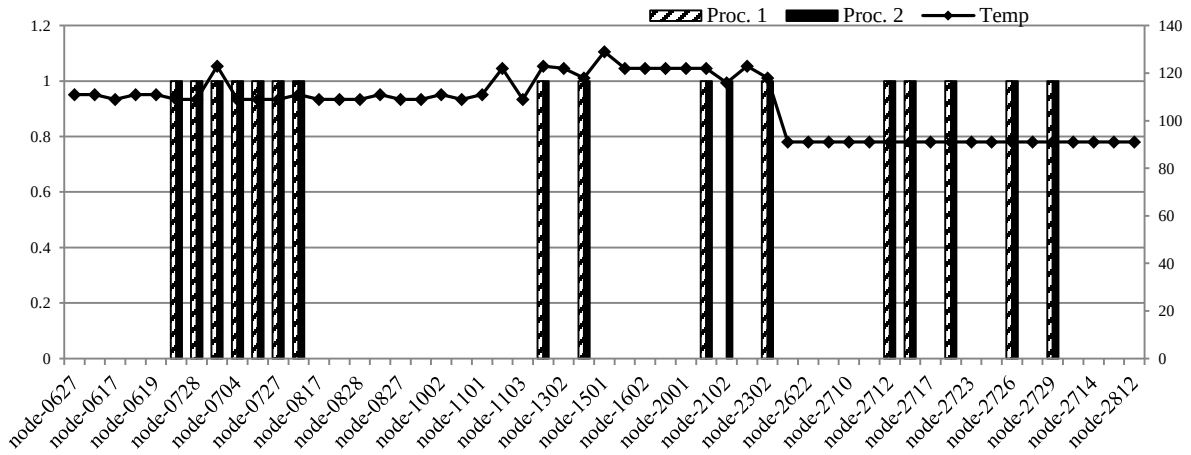


Fig. 7.2. Thermal Signatures vs. Processor On-Off States on Fri. 20 Feb. 2009 09:01 AM.

As stated previously, all of the servers were of dual core. Therefore, a single server can have four different states: (a) 00, both of the processors are idle, (b) 01, Processor 1 is idle and Processor 2 is busy, (c) 11, both processors are busy, and (d) 10, Processor 1 is busy and Processor 2 is idle. We perform the Mean Procedure technique on the individual servers to analyze the amount of time a server was executing a task based on the busy-idle state of the processors. Moreover, we also performed an aggregate analysis of all of the servers to identify the different states of the servers over a period of time. The results of our analysis are shown in

Table 7.1. We can observe that 90% of the time the processors have the same statistics that indicates that both of the processors of a server are in the same state. Fig. 7.1 depicts some of the statistical measures, such as mean, standard deviation (σ), minimum, and maximum thermal values of all of the servers in Pod 1. The said measures are the results of the Mean Procedure, when applied to the thermal readings of Pod 1 from the workload. As seen in Fig. 7.1 the Server T01 has the largest mean value of 107.76°F followed by T18 and T19 having 99.22 °F and 99.12°F mean value, respectively. The highest mean value indicates that the servers were either running more tasks or were geographically located at a position where the ambient temperature was high. The σ values of the servers are low, indicating that the dispersion and variation of the temperature is very close to the mean. Similarly, the minimum and the maximum temperature of the servers T01, T18, and T19 are higher than the rest of the servers in the pod.

We calculate the values for all of the servers within all of the 33 pods. In every pod, the servers we mentioned above (T01, T18, and T19) have the same pattern of thermal signatures. Moreover, a similar pattern of thermal signature were followed by server T02, T03, and T04, where thermal signature of T03 was always less than the server T02 and T04. Furthermore,

server T26, T27, and T28 follow a similar pattern of having thermal signatures within the same range. The on-off states of the processors can assists to identify whether the change in temperature of the server is due to the job execution or ambient effect. Fig. 7.2 depicts the thermal signature and the on-off states of various servers on Fri. 20 Feb. 2009, 09:00 AM (epoch time: 1235120445). The thermal signatures of different nodes, such as node-0817, node-1302, and node-2102 are greater than 100°F. However, the processors of the said servers are either off or only one of the processors is on. Therefore, we can infer from Fig. 7.2 that the high temperature of the server was possibly because of the ambient effect. The node-0801 represents Pod number 08 and Server number 01. It can be observed from Fig. 7.2 that the thermal signatures of Server 01 in all of the pods are higher than rest of the servers in the same pod. The aforesaid confirms the thermal pattern of the servers discussed above in Fig. 7.1.

7.2.3. Correlation Procedure

The Correlation Procedure (CORR) in SAS computes the correlation coefficients. The CORR measures the strength of a relationship between variables, having a value between $+1$ and -1. The signs ($-$ or $+$) of the correlation value defines the direction of the relationship between the variables. The correlation value of $+1$(highest correlation) means that when the value of one variable increases the other will also increase. The correlation value of -1 (lowest correlation) indicates that when one variable increase the other decreases. The correlation value of 0 means no relationship between the variables. In our study, we compute the correlation between the temperatures of the servers to identify the thermal effect of one server on the others. In the said perspective, we use the Pearson Correlation Coefficient (PCC) to compute the correlation of one server with the rest of the servers in the pod. The PCC between two variables is defined as the

covariance of two variables divided by the product of their σ value. The formula used to compute the PCC for the population is generally represented as ρ(rho) and can be calculated as:

$$\rho_{(X,Y)} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \tag{7.1}$$

where $\mathrm{cov}(X, Y)$ is the covariance of X and Y. In our case, the X is the set of thermal values of Server 1 and Y is the set of thermal values of Server 2. We have 33 pods and each pod has 32 servers. Therefore, we compute the $\rho_{(X,Y)}$ for all of the servers with every other server in the pod, represented and calculated as:

$$\sum_{i=0}^{N} \sum_{j=1}^{n} \rho_{(X_i,Y_j)} = \sum_{i=0}^{N} \sum_{j=1}^{n} \frac{cov(X_i,Y_j)}{\sigma_{X_i} \sigma_{Y_j}}, \tag{7.2}$$

where N is the total number of pods and n is the number of servers in the pod. Initially, we applied the CORR to only few pods. The PCC when applied to the sample data is commonly represented as r, and can be calculated by substituting the estimates of the covariances and variances based on the sample as follows:

$$r = \frac{\sum_{i=0,\forall i \in N' \wedge N' \in N}^{N'} \sum_{j=1}^{n} \{(X_i - \bar{X})(Y_j - \bar{Y})\}}{\sqrt{\sum_{i=0,\forall i \in N' \wedge N' \in N}^{N'} (X_i - \bar{X})^2} \sqrt{\sum_{j=1}^{n} (Y_j - \bar{Y})^2}}, \tag{7.3}$$

where N' is the selected number of pods from the total number of pods N. First, we observe the values for the sample size and then we calculated the values for the whole population of servers. As stated above, if the servers have high correlation, then the thermal dynamics of both servers follows same pattern. The higher correlation result indicates high relationship among the servers.

Lower correlation result among the servers indicates that the change in the thermal signature of one server have low impact on other server. The correlation results are used specifically for two reasons: (a) to identify the location of the servers with respect to the other

related servers in a pod and (b) to define the radius of the server's thermal effect of job allocation to other servers. The Fig. 7.3 and Fig. 7.4 below depict the correlation of servers with other servers in a pod. Fig. 7.3 shows the correlation results of first ten servers of pod 01 with each other. Moreover, we also consider the current state (as in Mean Procedure) of the server while measuring the correlations. The state of the servers is an important aspect in identifying the location of the servers, as it clarifies the increase in the thermal signatures of the servers. For instance, in Fig. 7.4 the correlation of T02 with T03 and T04 is 0.802 and 0.795, respectively. If we analyze the server states in Fig. 7.3, then we can observe that even when the server state is idle (00) the correlation of T03 and T04 with T02 is high. The aforesaid elucidates that the raise in the thermal signatures of T03 and T04 is highly affected by the raise in the thermal signature of T02. We define the radius of the thermal effect by considering the states and high correlation of servers to other related servers.

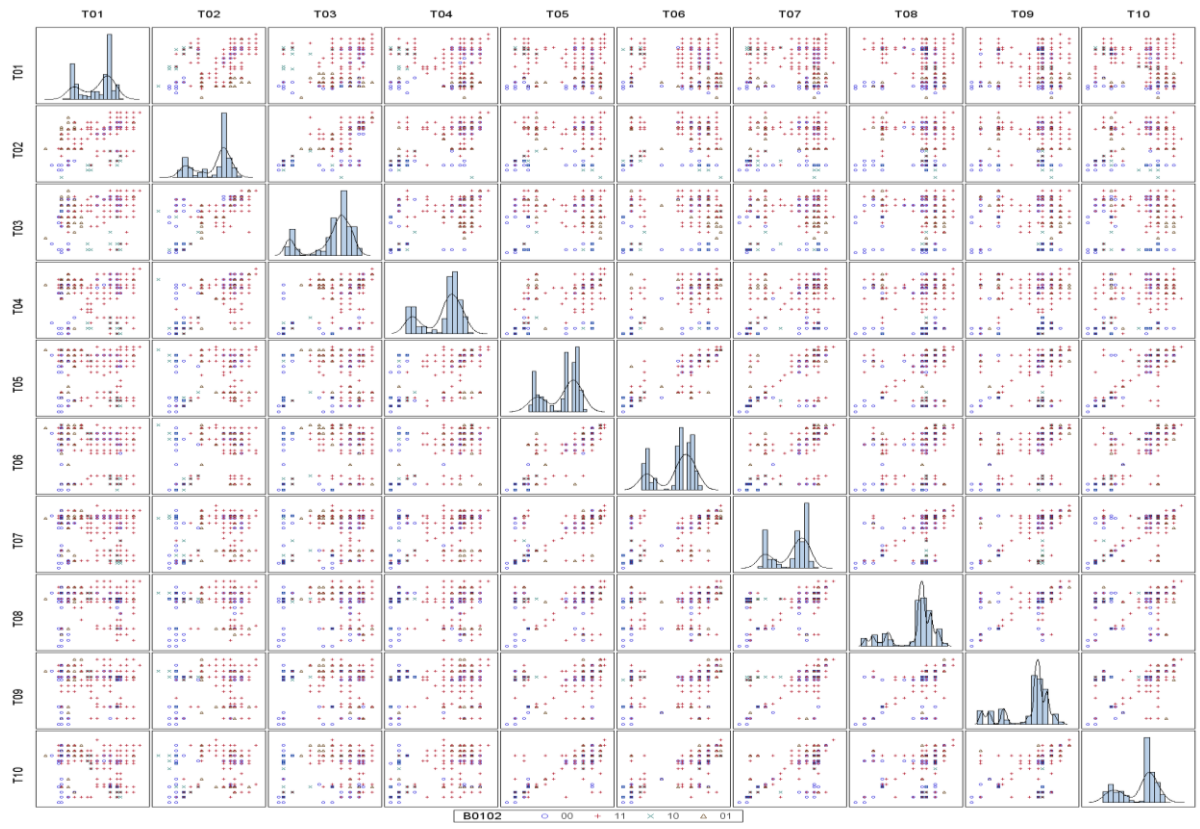


Fig. 7.3. Simple Correlation of First Ten Servers in Pod 1 Using Different States of Server.

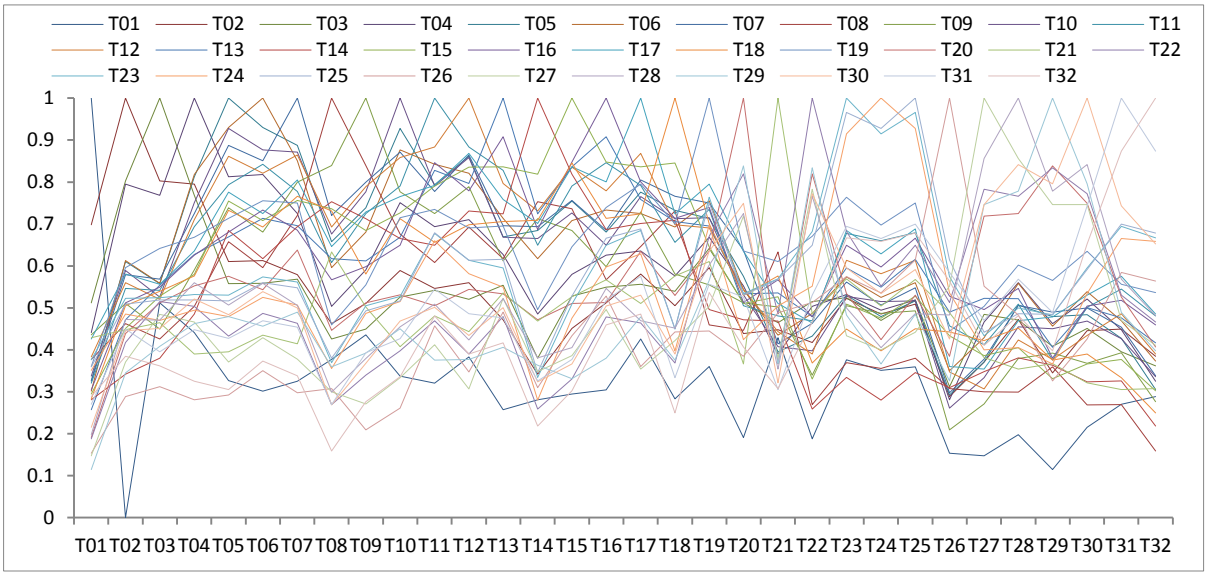


Fig. 7.4. The PCC of Servers in Pod 1.

7.2.4. VARMAX Model

The Vector AutoRegressive Moving-Average model with eXogenous variables (VARMAX) is used to capture the linear interdependencies among multiple time series variables that in our case are the temperature of the servers over time. Given a time series of data, the VARMAX model is used for understanding and predicting values in the given data. In practice, the prediction or forecasting model may get affected by some other observable variables that are determined outside the system interests, termed as exogenous variables. In our study, the temperature of the servers is affected by the execution of tasks, so the task allocation is an endogenous variable (variables within the system). Moreover, the temperature can also be affected by the CRAC supplied temperature and also by the ambient temperature, which are the examples of exogenous variables. The valid assumptions related to exogenous variables allow modeling strategies to reduce computational expense and help isolate invariants of underlying mechanisms [7.51].

The VARMAX Procedure

Number of Observations 583
Number of Pairwise Missing 0

Variable	Type	N	Mean	Standard Deviation	Min	Max
T27	Dependent	583	88.89708	6.43202	78.00000	98.00000
T28	Dependent	583	89.10292	6.01228	77.00000	98.00000
T30	Dependent	583	89.88165	5.71169	78.00000	98.00000
B27	Independent	583	0.48542	0.50022	0.00000	1.00000
B28	Independent	583	0.47684	0.49989	0.00000	1.00000
B30	Independent	583	0.43053	0.49558	0.00000	1.00000

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
T27	CONST1	27.80807	2.33597	11.90	0.0001	1
	XL0_1_1	4.90534	0.82030	5.98	0.0001	B27(t)
	XL0_1_2	-1.58335	0.79365	-2.00	0.0465	B28(t)
	XL0_1_3	1.54001	0.30430	5.06	0.0001	B30(t)
	AR1_1_1	0.54813	0.06153	8.91	0.0001	T27(t-1)
	AR1_1_2	-0.00299	0.06239	-0.05	0.9618	T28(t-1)
	AR1_1_3	0.02253	0.04339	0.52	0.6038	T30(t-1)
	AR2_1_1	0.03321	0.05840	0.57	0.5699	T27(t-2)
	AR2_1_2	0.01962	0.06185	0.32	0.7512	T28(t-2)
	AR2_1_3	0.03994	0.04308	0.93	0.3542	T30(t-2)
T28	CONST2	25.03064	2.46716	10.15	0.0001	1
	XL0_2_1	-1.47880	0.86637	-1.71	0.0884	B27(t)
	XL0_2_2	3.84261	0.83822	4.58	0.0001	B28(t)
	XL0_2_3	1.57167	0.32139	4.89	0.0001	B30(t)
	AR1_2_1	-0.06124	0.06499	-0.94	0.3465	T27(t-1)
	AR1_2_2	0.61261	0.06589	9.30	0.0001	T28(t-1)
	AR1_2_3	0.02620	0.04583	0.57	0.5677	T30(t-1)
	AR2_2_1	-0.10857	0.06168	-1.76	0.0789	T27(t-2)
	AR2_2_2	0.18733	0.06533	2.87	0.0043	T28(t-2)
	AR2_2_3	0.04158	0.04550	0.91	0.3611	T30(t-2)
T30	CONST3	18.79489	2.45995	7.64	0.0001	1
	XL0_3_1	-2.52295	0.86384	-2.92	0.0036	B27(t)
	XL0_3_2	4.20055	0.83577	5.03	0.0001	B28(t)
	XL0_3_3	0.82251	0.32045	2.57	0.0105	B30(t)
	AR1_3_1	-0.08975	0.06480	-1.39	0.1666	T27(t-1)
	AR1_3_2	0.21052	0.06570	3.20	0.0014	T28(t-1)
	AR1_3_3	0.48991	0.04570	10.72	0.0001	T30(t-1)
	AR2_3_1	-0.01402	0.06150	-0.23	0.8197	T27(t-2)
	AR2_3_2	-0.13757	0.06514	-2.11	0.0351	T28(t-2)
	AR2_3_3	0.31873	0.04537	7.03	0.0001	T30(t-2)

Fig. 7.5. Output Results of VARMAX Model.

There are three parts associated with VARMAX: **(a)** autoregressive, **(b)** moving average, and **(c)** exogenous variable that is why the model is referred to as the VARMAX (p, q, s), where p is the order of the autoregressive part, q is the order of moving average, and s represents the exogenous variables. The VARMAX model has a form that can be written as:

$$y_t = \sum_{i=1}^{p} \Phi_i y_{t-1} + \sum_{i=0}^{s} \Theta_i^* X_{t-1} + \varepsilon_t - \sum_{i=1}^{q} \Theta_i \, \varepsilon_{t-i,} \tag{7.4}$$

where $y_t = (y_{1t}, y_{2t}, \dots, y_{kt})$ are the output variables of interests, which can be affected by the input variables $X_t = (X_{1t}, X_{2t}, \dots, X_{rt})$ that are determined outside the system. Therefore, y_t and X_t are endogenous and exogenous variables, respectively. The $\varepsilon_t = (\varepsilon_{1t}, \varepsilon_{2t}, \dots, \varepsilon_{ut})$ is referred as the unobserved noise variable. The Φ_i and Θ_i are $k \times k$ matrices of autoregressive coefficients, and Θ_i^* is $k \times r$ matrix of coefficients. The complete details of the aforesaid matrices and VARMAX model can be seen at [7.52]. In our study, the on/off states of the processors are taken as the independent variables, the exogenous variables are the supplied temperature from the CRAC, and the thermal signatures of the servers are considered as the dependent variables.

The Fig. 7.5 depicts the descriptive statistics along with the estimated parameters of the fitted model for three servers (Server 27, Server 28, and Server 30) with respect to their states. Moreover, the figure also depicts the parameter estimates and their significance that indicates how well the model fits the data. The column N depicts the number of non-missing observations. The entry T27 and B27 in the Variable column shows the thermal signatures and on (1)/off (0) states of the Server 27 in Pod 1, respectively. The Type column specifies that the variables are either dependent or independent. In our case, the thermal signatures of the servers are dependent on the execution of the processors. For instance, if the processors of a particular server are executing a job, then the thermal signature of the server is increased. The fitted model for the variables in Fig. 7.5 is given as:

$$y_t = \begin{pmatrix} 0.548 & 0.001 & 0.023 & 0.033 & 0.020 & 0.040 \\ -0.061 & 0.613 & 0.026 & -0.110 & 0.187 & 0.042 \\ -0.090 & 0.211 & 0.490 & -0.014 & -0.140 & 0.320 \end{pmatrix} y_{t-1} + \varepsilon_t \qquad (7.5)$$

The table of parameter estimates in Fig. 7.5 lists the parameters in the model. Moreover, for each parameter, the table shows the estimated value, standard error, and t value of the

estimates. In our case, there are 30 parameters in the model. The constant term is T27 (thermal signature of Server 27) that we are trying to forecast using the thermal values of T28, T30 and the on/off states of the processors. Our model attempts to estimate the thermal value of T27 from two preceding thermal values of T27, T28, and T30. The autoregressive parameters are labeled as AR1_1_1 that represents the coefficient of the lagged value of the change in thermal value, whose estimate is 0.54813.

By using all of the aforementioned statistical techniques, our aim is to analyze the thermal patterns and behavior shown by the servers in the CCR DC workload. It is noteworthy that the workload does not provide any information about the spatial location of the servers and pods. Therefore, analyzing and forecasting the effect of task execution on the server and to the other related servers become a challenging task. Based on the findings and analysis performed on the workload, we propose a thermal aware strategy for task allocation in a DC. The details and working of the strategy is discussed in the next section.

7.3. Thermal Aware Resource Allocation (TARA) Strategy

We propose a thermal aware resource allocation strategy that manage and control the thermal dynamics of DC. The goal is to reduce the thermal imbalances within the pods and servers of the DC. The jobs are allocated to the servers based on the thermal signatures and ambient effect on the other related servers. We propose two variants of TARA strategy that adopts two different formulations to allocate tasks to the servers. Before going deep into the details of the TARA strategy, let us briefly discuss the system model.

7.3.1. System Model

A DC is comprised of computing resources, such as servers and the network infrastructure, such as switches, interconnecting all of the computing resources. A DC follows a hierarchical model, where the computing resources reside at the lowest layer as depicted in Fig. 7. 6. The network infrastructure can be considered as a multilayer graph [7.2]. The servers, access switches, and aggregate switches are assembled in modules (referred to as pods) and are arranged in three layers, namely: **(a)** access, **(b)** aggregate, and **(c)** server layer. The core layer is used to connect all of the independent pods together. The DC (DC) can be divided in two logical sections: **(a)** Pods (zones) and **(b)** Core Layer Switches, as below:

$$DC = Pod_{\forall i \in k}(i) \cup C_{\forall q \in r}(q), \tag{7.6}$$

where $\mathrm{C(q)}$ is the set of core layer switches and r is the total number of core switches (γ) in the network. $\mathrm{Pod(i)}$ is the set of pods and k is the total number of pods in the DC. Each access layer switch (α) is connected to n number of servers (S) in a pod. Moreover, every α is connected to every aggregate switch (δ) in the pod. The number of servers (including S, α, and δ) in $\mathrm{Pod(i)}$ can be calculated as:

$$Pod\ (i) = S^{i}_{(n \times m)} \cup \alpha^{i}_{m} \cup \delta^{i}_{w} \tag{7.7}$$

where $\mathrm{S^{i}_{(n\times m)}}$ represents a set of servers connected to α in $\mathrm{Pod\ (i)}$. The $\alpha^{\mathrm{i}}_{\mathrm{m}}$ represents access layer switches in $\mathrm{Pod\ (i)}$, where m is the total number of α in $\mathrm{Pod\ (i)}$. The $\delta^{\mathrm{i}}_{\mathrm{w}}$ represents aggregate layer switches and w is the number of δ in $\mathrm{Pod\ (i)}$. The components in DC work cooperatively to accomplish the assigned tasks. The mechanical energy consumed and almost all of the power drawn by the computing devices is dissipated as heat. We model the heat dissipation of servers within a DC, represented as ς_s that can be calculated as follows:

$$\varsigma_s^{i,\alpha} = \left(\varsigma_0 + \varsigma_p + \varsigma_m\right)^{i,\alpha}, \tag{7.8}$$

where

$$\varsigma_p^{i,\alpha} = \left(\varsigma_{rw} + \varsigma_{op}\right)^{i,\alpha}. \tag{7.9}$$

The $\varsigma_0^{i,\alpha}$ represents the heat dissipated as a result of the static power to keep the server active, and $\varsigma_p^{i,\alpha}$ represents the heat dissipation when the processing is being performed. The $\varsigma_0^{i,\alpha}$ is fixed that does not change and is independent of workload. However, $\varsigma_p^{i,\alpha}$ is dynamic and is dependent on the workload.

The $\varsigma_m^{i,\alpha}$ represents the heat dissipated by the memory that includes energy consumed during the memory refresh operations. The $\varsigma_p^{i,\alpha}$ is further decomposed into $\varsigma_{rw}^{i,\alpha}$ that represents the heat dissipation because of the read and write operations, and $\varsigma_{op}^{i,\alpha}$ is the heat dissipation as a result of the processing performed. The heat dissipated by all the servers in Pod (i), represented as $\S_s^i$, can be calculated as:

$$\S_s^i = \sum_{p=1}^{m} \sum_{x=1}^{n} \left(\varsigma_x^{i,m}\right), \tag{7.10}$$

where the $\varsigma_x^{i,m}$ represents the heat dissipation of S_x connected to m number of α switches in Pod (i). As stated above the DC is comprised of network infrastructure and servers. Therefore, the heat dissipation of the pod(i), represented as τ_ρ^i, can be calculated as:

$$\tau_\rho^i = \S_s^i + \S_\partial^i + \S_\wp^i, \tag{7.11}$$

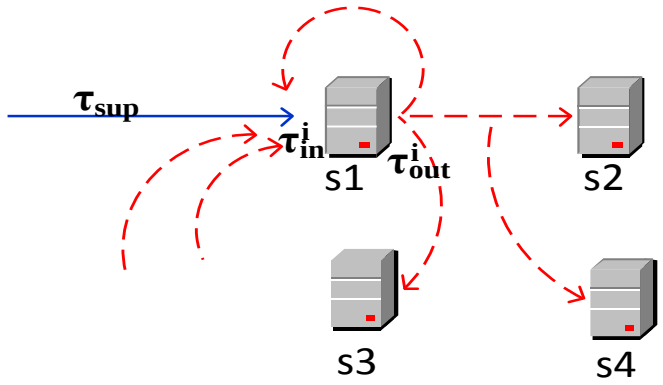


Fig. 7.6. Heat Exchange among Servers.

Two temperatures are associated with each server, the (a) input temperature (τ^{i}_{in}) and (b) output temperature (τ^{i}_{out}). The τ^{i}_{in} represents the input ambient temperature of server that includes the heat received from other servers and surroundings. As depicted in Fig. 7.6, τ^{i}_{in} of s1 involves the recirculation (red dotted lines) of hot air from other servers and cooling temperature (τ_{sup}) from CRAC. The heat dissipated by any server i∈ ℵ will change the τ^{i}_{out}. The variables τ^{i}_{in} and τ^{i}_{out} represent the temperature of the surroundings and not the server. However, the heat dissipated by the server (π^{i}_{out}) effects the values of τ^{i}_{in} and τ^{i}_{out}. The input temperature of a server (π^{i}_{in}) can be calculated as:

$$\pi^{i}_{in} = \varrho^{i}(\tau^{i}_{in}), \tag{7.12}$$

where

$$\tau^{i}_{in} = \sum_{j=1}^{ℵ}(\pi^{j}_{out}) + \tau_{sup}. \tag{7.13}$$

The ϱ is an air coefficient that represents the product of air density, heat of air, and flow rate of air. The π^{i}_{out} can be calculated as:

$$\pi_{out}^{i} = \pi_{in}^{i} + \text{ב}^{i}, \tag{7.14}$$

where (7.15)

$$\text{ב}_{i} = \varrho^{i}(\tau_{out}^{i} - \tau_{in}^{i}).$$

The ב_{i} represents the heat dissipation of a server $\text{i} \in$ א in proportion to the power consumed for processing. The current temperature of S_i in Pod(j) is denoted as $t_{cur}^{i,j}$ that can be calculated as:

$$t_{cur}^{i,j} = \pi_{in}^{i} + \Delta t(c_i), \tag{7.16}$$

where $\Delta t(c_i)$ represents the anticipated change in the temperature caused by executing a task c_i on S. According to the abstract heat model of a DC, as discussed in [7.53], the heat distribution and its effect on the surrounding machines can be represented as cross interference coefficient matrix. We follow the same model and compute the heat distribution of the servers using a matrix, represented as $h_{n\times n} = \{\partial_{i,j}\}$, which denotes the thermal effect of S_i on S_j and can be computed as:

$$\partial_{i,j} = \tau_{out}^{i} \times k \times \frac{1}{\hat{h}_j}, \tag{7.17}$$

where k is the thermal conductivity constant of the air and $\hat{h}$ is the hop count of S_j from S_i. In the following section we will discuss the two variants of TARA strategy in details.

7.3.2. TARA-I

When a new job arrives to the DC, the resource manager selects the pod with a minimum average thermal signature to allocate the job. The servers are sorted in the selected pod based on the thermal signatures, such that the server that has the lowest temperature is first in the order. The job is assigned to the server that is first in the order. Initially all the servers have same thermal signatures. When the job is assigned to the server the thermal signature of the server

increases from $t_{cur}^{i,j}$ to $t_{cur}^{i,j} + \Delta t(c_i)$. The increase in thermal signature of the server will in return have the ambient effect on other servers, as in (16). It is noteworthy that the closer the server is, the higher the ambient effect to that server. We assume that the servers within the $\hat{\mathrm{h}} = r$ are the ones affected by the ambient effect of task allocation. The steps performed by TARA-I to allocate jobs to the servers are depicted in Fig. 7.7. The TARA-I strategy adopts a greedy approach to allocate a job. Initially all the servers have same thermal signatures. Suppose a new job is arrived and all the servers in Fig. 7.8 have the same thermal signatures. The server s1 will be selected by TARA-I. The thermal signature of s1 will increase as a result of job allocation. In the said perspective, if we assume $r = 3$, then the ambient effect will increase the temperature of the neighboring servers in $\hat{\mathrm{h}} = 3$. Therefore, the temperature of $s2, s3$, and $s4$ also increases (red dotted line in Fig. 7.8). When the next job arrives, all the servers are sorted again and the server with the lowest temperature is selected. As a result, the next job is allocated to $s5$ that in return have ambient effect on the neighboring servers (solid red line in Fig. 7.8). By doing the aforesaid, TARA-I selects a server with a minimum thermal signatures by considering the ambient effect as a result of job allocation. The aim is to distribute the workload so that the thermal imbalance and occurrence of hotspots within a DC can be reduced. The TARA-I considers the ambient effect while allocating new jobs to the servers. However, if we analyze closely, then we will observe that in Fig. 7.8, selecting s5 for the next job is not the best option. The reason is that the server $s2, s3$, and $s4$ are affected in both of the allocations and their temperatures are raised even without executing a job. The aforesaid scenario can possibly create a hotspot in a long run. In the said perspective, we propose another variant of TARA strategy, termed as TARA-II that is more efficient and resolves the issue discussed above.

1: **for** $i \leftarrow 1$ to k **do**

2: $\tau_{\rho}^{i} = \S_{s}^{i} + \S_{\partial}^{i} + \S_{\mathscr{g}}^{i}$

3: **end for**

4: **Select** $min\ (\tau_{\rho}^{i})$

5: **Get** $\varsigma_{s}^{i,\alpha}\ \forall\ s \in n\ \wedge \alpha \in m$

7: **Select** $\varsigma_{s}^{i,\alpha}$, **such that** $\varsigma_{s}^{i,\alpha} < \varsigma_{y}^{i,\alpha}\ \forall\ y \in n\ \wedge \alpha \in m \wedge y \neq s.$

8: **Allocate** c to $\varsigma_{s}^{i,\alpha}$, $iff\ \varsigma_{s}^{i,\alpha} + \Delta t(c) < \tau_{max}^{\varsigma}$

Fig. 7.7. Steps Involved in TARA-I.

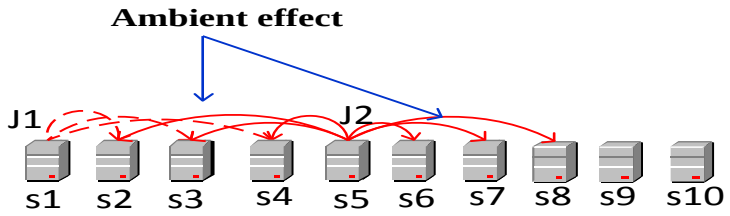


Fig. 7.8. TARA-I Allocation and Ambient Effect on Servers.

7.3.3. TARA-II

In TARA-II the process of task allocation is based on the pre-calculations of ambient effect that occur as a result of task allocation. Whenever a new job arrives, the resource manager selects the pod with a minimum average thermal signature. Once the pod is selected, servers are sorted in an ascending order, such that the server with the lowest thermal signature is ranked first. After the servers are sorted, the scheduler performs the pre-calculations. The pre-calculations are executed in a way that the scheduler picks every server one by one and computes the $t_{cur}^{i,j}$ (using (15)) along with the ambient effect on other servers, as in (16). Once the pre-calculations are completed for all the servers, the results are normalized and the server with the

minimum ambient effect and $t_{cur}^{i,j}$ is selected for the allocation. Our aim to perform the pre-calculations is to select the server that has the minimum ambient effect among all the servers. Performing the pre-calculations allow us to uniformly distribute the workload and balance the thermal dynamics of the DC. In Fig. 7.8, if TARA-II is used, then for first job the server $s1$ is selected. For the next job, pre-calculations are performed again and as a result the server $s8$ will be selected, as opposed to $s5$ when TARA-I was used. The servers $s2, s3, s4, s5, s6,$ and $s7$ are not selected because they have higher ambient thermal effect and thermal signatures within the $\hat{\mathrm{h}} = 3$.

As stated in Section III, the nature of the workload, such as the job size, job arrival rate, and number of processors required by the job is not uniform. In TARA-II, we observed that at certain time intervals the average thermal signatures of the pods may fluctuate and becomes non-uniform. The reason for the above is that long jobs are assigned to one or more servers in a pod, causing the thermal signatures of the servers to escalate that ultimately increase the average thermal signature of the pod. In the said perspective, to stabilize and balance the thermal signatures of the pods, we use task migrations in TARA-II. Task migration is an expensive operation that involves a lot of network activities at all layers of DC, such as access, aggregate, and core. However, in this paper our focus is on the thermal dynamics of the DC, so we will not go into the details of the cost involve in task migration. The following paragraph will discuss in details the process of task migration in TARA-II.

The $\varsigma_s^{i,\alpha}$ for all the $S \in Pod(i)$ is measured and observed through sensors periodically, as in Fig. 7.9. Whenever the value of $\varsigma_s^j, \forall j \in n$ exceeds the maximum threshold temperature of the server (τ_{max}^{ς}), the local controller migrates some tasks from S^j to S^l, where S^j and S^l are

connected to the same α. For the tasks to be migrated successfully to S^l, the constraint $\varsigma_s^l + \Delta T < \tau_{max}^{\varsigma}$, must be satisfied. The ΔT represents the anticipated increase in the temperature as a result of task migration. If the task migration is not possible among the serves under α_i, then the servers belonging to $\alpha_j, \forall j \in m \wedge j \neq i$ are considered for the migration. The α_i and α_j belongs to the same pod. When the migration is performed within the same pod it is known as Intra-pod migration. Moreover, if there is no server available for the migration within the same pod, then Inter-pod migration is performed by enforcing the same constraints as in Intra-pod migration. For Inter-pod migration, the centralized controller periodically monitors the average thermal values of each pod that it receives from the sensors. Whenever the thermal signature of the $Pod(i)$ ($\tau_\rho^i = \S_s^i + \S_\partial^i + \S_\not{g}^i$) starts to exceeds the maximum thermal threshold value of the pod (τ_{max}^ρ), the centralized controller instructs the local controller of $Pod(i)$to migrate some tasks to $Pod(j), \forall j \in k \wedge \text{j} \neq \text{i}$. The migration can be successfully performed only if the $\tau_\rho^i + \Delta T < \tau_{max}^\rho$. The server selection and task allocation performed during Inter-pod migration is same as discussed above in Intra-pod migration. The centralized controller only has the coarse-grain information of the τ_ρ^i. The allocation of migrated tasks to the servers is performed by the local controller through the use of fine-grained thermal information of servers. The intra-pod and inter-pod migrations are focused on maintaining the unified thermal threshold value in all the pods. The thermal signatures of servers in DC evolve in order of minutes. Moreover, the power states of servers can change as frequent as milliseconds. Therefore, the threshold temperatures are not absolute values; rather it is a range within which the thermal signatures of the servers should lie. In the next section we will discuss the implementation results of our proposed strategies along with the other existing strategies.

1: **for** *i* ←*1* to *k* **do**
2: $\tau_{\rho}^{i} = \S_{s}^{i} + \S_{\partial}^{i} + \S_{\wp}^{i}$
3: **end for**
4: **Select** $min\ (\tau_{\rho}^{i})$
5: **Sort** $\varsigma_{s}^{i,\alpha}\ \forall\ s \in n\ \wedge \alpha \in m$, **such that** $\varsigma_{s}^{i,\alpha} < \varsigma_{s}^{i+1,\alpha} < \cdots < \varsigma_{s}^{i+n,\alpha}$
6: **for** *j* ←*1* to *m* **do**
7: **for** *i* ←*1* to *n* **do**
8: **Select** $\varsigma_{s}^{i,j}$ and **compute updated** $t_{cur}^{i,j}$ **and the ambient effect on other server, as in** (16)
9: **end for**
10: **end for**
11: **for** *j* ←*1* to *m* **do**
12: **for** *i* ←*1* to *n* **do**
13: **for** *k* ←*0* to *r* **do**
14: $NTemp_{s}^{i,j} = \varsigma_{s}^{i+k,j}$
15: **end for**
16: $N_{s}^{i,j} = NTemp_{s}^{i,j}/r$
17: **end for**
18: **end for**
19: **Select** S_i, **such that** $\min(N_{s}^{i,j})$
20: **Allocate** *c* to S_i, $iff\ \varsigma_{s}^{i,\alpha} + \Delta t(c) < \tau_{max}^{\varsigma}$
21: **If** $\varsigma_{s}^{i,\alpha} > \tau_{max}^{\varsigma}\ \forall\ s \in n\ \wedge \alpha \in m$, **then**
22: **Migrate-task** c **from** S_i**to** S_j, $iff\ \varsigma_{j}^{i,\alpha} + \Delta t(c) < \tau_{max}^{\varsigma}$
23: **end if**

Fig. 7.9. Steps Involved in TARA-II.

7.4. Results and Discussion

As stated in Section 2 that the workload traces used for the simulations are obtained from CCR, State University of New York at Buffalo. The complete detail of the workload is provided in Section 2. We execute the proposed strategies TARA-I and TARA-II on the aforesaid workload to obtain realistic results. Moreover, we also perform a comprehensive comparison of the proposed strategies with three classical and two thermal aware scheduling approaches. The purpose of the comparison is to highlight the improvements achieved by the proposed strategies. The jobs and the logs from the CCR dataset are used as an input for our simulation of the proposed and all of the other studied strategies. We use the term “hot” and “cool” for a job that

indicates the high thermal impact and low thermal impact on the servers, respectively, as in [7.20, 7.25]. Similarly, the server is termed as “hot” or “cool”, if the thermal signature of the server is high or low, respectively. The thermal impact of a job is usually measured based on attributes, such as the length of the job and number of processors required. Before going deeper into the details of the comparison, let us briefly discuss the approaches used for the comparison.

The FCFS (sometimes referred as first-in, first-out) is possibly the most straightforward scheduling approach. The FCFS is instinctively fair, where the jobs are executed based on the order they are received to the scheduler. Therefore, the jobs that are submitted first are allocated first. However, the policy is non-preemptive, where longer jobs can add delays. In the said perspective, the order in which the jobs are received is very critical. If a longer job is first in the order, then mean wait time for all of the jobs will be high.

The Shortest Job First (SJF) [7.54] sometimes referred to as the shortest job next, is a scheduling strategy that selects the waiting process with the smallest execution time to execute next. Unlike FCFS, the SJF scheduling policy reduces the mean waiting time of the jobs as shortest jobs are executed first. However, the SJF is also non-preemptive, where the jobs that are long may starve for the execution. The SJF strategy is used in specialized environments where accurate estimates of running times are available [7.55].

In the Longest Job First (LJF) [7.36] scheduling policy the job that requires the longest processing time is executed first. The LJF policy sorts the jobs in an increasing order, such that the jobs that requires more time for completion is scheduled first. The mean waiting time of the jobs is high in LJF as longer jobs are executed first. Moreover, the shorter jobs have to suffer due to extended wait periods. The LJF is useful for the jobs that are submitted for batch processing

[7.41]. The batch jobs are typically long, where the resources are longer bound to the jobs, causing less resource fragmentation, and increase the utilization and throughput of the system [7.41].

The approach in [7.23] follows the steps of Genetic Algorithm (GA). The first step is to construct a set of feasible solutions. The selected solutions are then mutated (randomly interchange the task allocations within the solution) and mated (randomly select pairs of solution and exchange the subset of two task assignment to get two new solutions). The fitness function that checks the highest inlet temperature of the selected assignments is applied to all of the solutions formed as a result of mating and mutation, including the original solution. Finally, the solution having the lowest inlet temperature value from the set of highest inlet temperature values, obtained as a result of fitness function, is selected as a final solution.

The approach in [7.24], Thermal Aware Scheduling Algorithm (TASA), is based on the theory of coolest inlet that attempts to schedule the larger jobs to the servers that have lowest thermal signatures. In TASA algorithm the servers are sorted in an ascending order of thermal signatures, such that the server with a lowest thermal signature is first in the order. Similarly, the jobs are sorted based on the task-temperature profile of the jobs, such that the job that has the highest thermal impact is first in the order. The goal is to maintain thermal balance within the pods of a DC by allocating longer jobs to the servers that have lowest thermal signatures.

Fig. 7.10 depicts the average minimum and maximum thermal signatures of the pods over the period of time, when the studied scheduling heuristics are used. There were 33 pods in the DC and each pod had 32 servers. The thermal readings were taken after every 10 minutes. Fig. 7.10 clearly differentiates the thermal dynamics of the DC when various scheduling approaches

are used. It can be infer from the above that each scheduling strategy has different thermal effect on the servers and ambient environment.

In SJF (Fig. 7.10(a)), high temperature peaks are frequent as compared to low temperatures. The job allocation in SJF is static that can create a situation, where small jobs are assigned to the servers with low thermal signatures and longer jobs are assigned to the servers with high thermal signatures. In the said perspective, "hot" servers becomes "hotter" and "cold" servers stays "cool", resulting in a thermal imbalance among the pods. However, in an ideal situation, the shortest job may be assigned to the highest thermal signature server and longest job may be assigned to the lowest thermal signature server. In the ideal scenario the thermal difference will be small, as depicted at some time intervals of Fig. 7.10(a). The thermal difference in SJF may also be small in a scenario, where only few jobs were submitted for execution, causing majority of the high thermal signature servers to cool down.

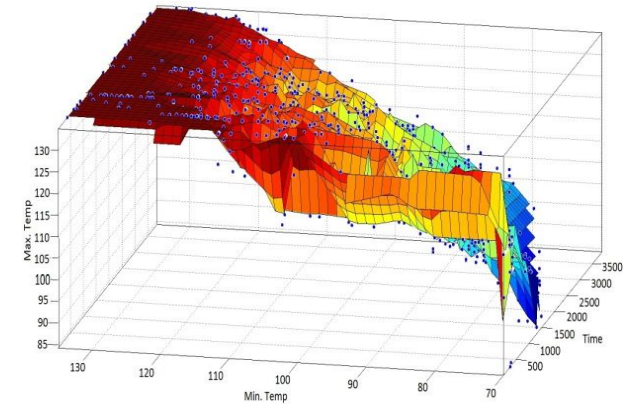
In FCFS (Fig. 7.10(b)), the thermal signatures of the pods have fluctuations, which mean that at many time intervals the thermal status of the pods is unbalance. The reason for the thermal imbalance between the pods is the static assignment of tasks without considering the thermal status of the servers. The aforesaid, possibly creates a scenario when higher task-temperature profile jobs are assigned to servers with high thermal signatures and lowest thermal impact jobs are assigned to low thermal signature servers. In such a scenario, the thermal signatures of "hot" servers increases and thermal signatures of "cold" servers decreases, causing thermal imbalance among the pods.

The reasons for the thermal variations in LJF (Fig. 7.10(c)) are the same as were in SJF. As both of the strategies uses static job assignment, the allocation of "hot" jobs to "hot" servers

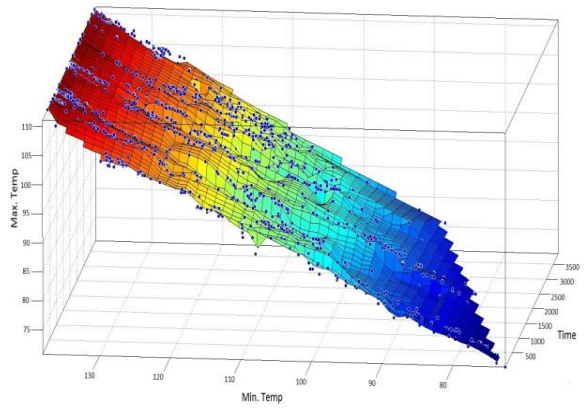
and “cool” jobs to “cool” server is possible that causes the thermal signatures of the pods to fluctuate.

However, the thermal differences in TASA (Fig. 7.10(d)) are low, as compared to SJF, FCFS, and LJF. The first reason is that the scheduling decisions are made considering the thermal status of the servers. Secondly, the “hot” jobs are assigned to “cool” servers that will allow the servers with higher thermal signatures to cool down, while the servers with low thermal signatures execute longer jobs, resulting in a thermal balance among the pods.

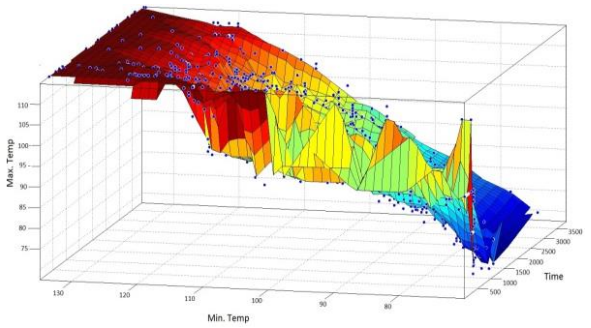
In GA-based (Fig. 7.10(e)), the reason for the imbalance thermal signatures is the random nature of the GA based approach. The selection of the feasible solution, the mutation, and the mating process, all are based on randomization. If the same set of pods and servers are selected in the solutions most of the time, then the fitness function performed on the selected solution will not provide any significant information to avoid the occurrence of the hotspots. Similarly, there is also a possibility that the number of tasks allocated to few pods and servers are relatively low as compared to the rest of the pods and servers in the DC. The aforementioned possibilities will allow some servers to have high thermal signatures while others have low thermal signatures that will ultimately results in the thermal imbalance.



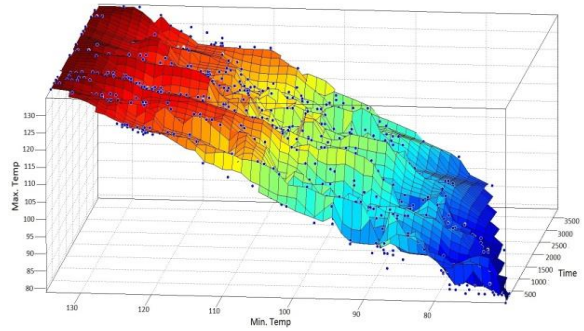
(a) SJF



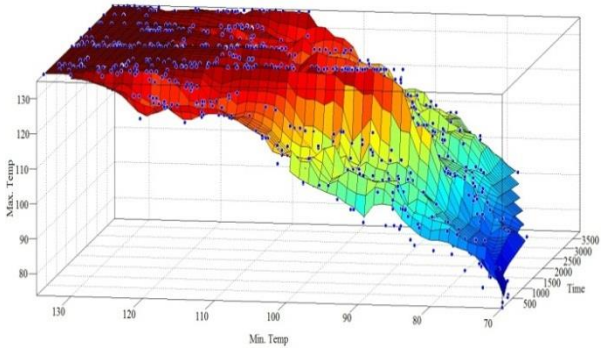
(b) FCFS



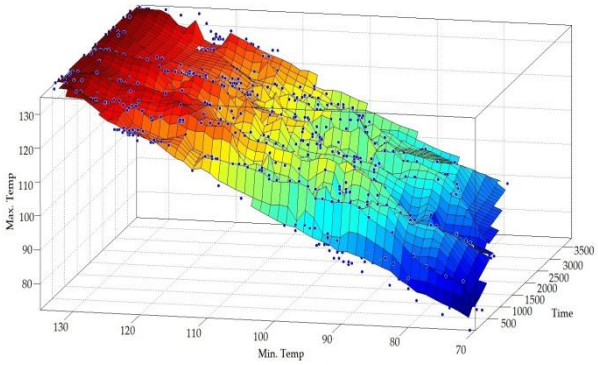
(c) LJF



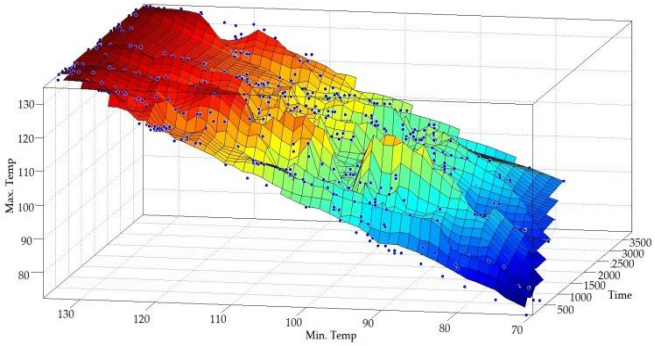
(d) TASA



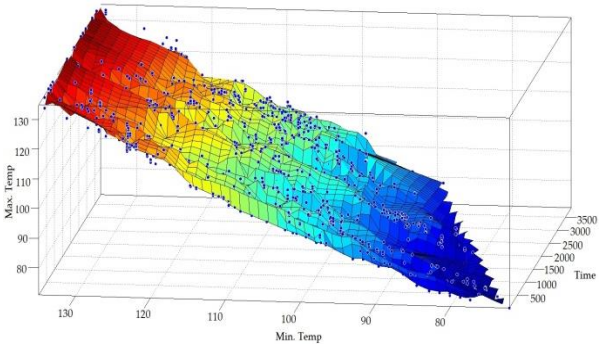
(e) GA-based



(f) Tara-I



(g) Tara-II (without migration)



(h) Tara-II (with both migrations)

Fig. 7.10. Average Minimum and Maximum Thermal Signatures of the Pods.

The thermal imbalances in Fig. 7.10(f), Fig. 7.10(g), and Fig. 7.10(h) are very low as compared to the thermal imbalances of all the other scheduling approaches. The reason for the aforementioned is that the scheduling decisions are taken by considering the thermal status of the servers and the ambient effect on other related servers. The thermal imbalances between Fig 11(f) and Fig. 7.10(g) are almost same. However, the thermal differences are very obvious in Fig. 7.10(h) as compared to Fig. 7.10(f) and Fig. 7.10(g). As stated in previous sections that the heterogeneous nature of the workload can cause thermal spikes within the pod, when two long jobs are assigned in a same pod. Therefore, to mitigate the aforesaid, TARA-II performs Intra-pod and Inter-pod task migrations to keep the thermal balance and to avoid hotspots. In the said perspective, we highlighted the effectiveness of migrations performed in TARA-II on the thermal dynamics of DCs. We plotted the differences of highest and lowest thermal signature of servers at certain time intervals to expose the thermal imbalances. We plotted four different scenarios in TARA-II: (a) no migration is performed, (b) only Intra-pod migrations are performed, (c) only Inter-pod migrations are performed, and (d) both (Intra and Inter-pod) migrations are performed. We can observe in Fig. 7.11 that the thermal differences in the scenarios (a) and (d) are highest and lowest at most of the time intervals, respectively. The results from Fig. 7.11 clarifies that using task migrations TARA-II can effectively maintain the thermal uniformity within the pods of DCs.

The results from our simulations have shown improvements in the thermal dynamics of the DC when the proposed strategies were adopted, as compared to the other studied approaches. Our proposed strategies consider ambient effect while allocating jobs to the servers that helps stabilize the thermal dynamics of the overall DC.

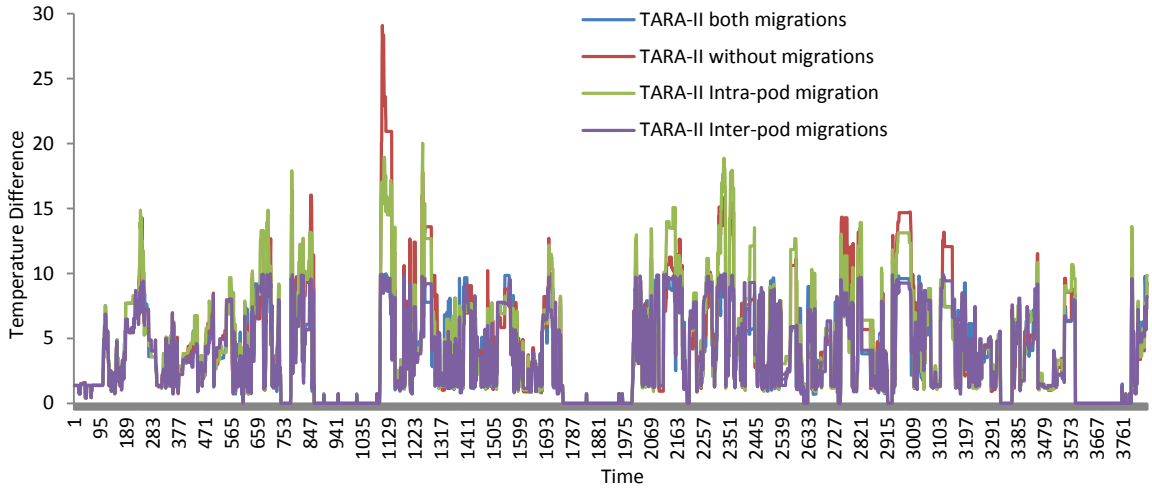


Fig. 7.11. Differences between the Highest and Lowest Thermal Signature of Servers Using TARA-II, when (a) No Migration, (b) only Intra-Pod Migrations, (c) only Inter-Pod Migrations, and (d) Both (Intra and Inter-Pod) Migrations are Performed.

7.5. References

[7.1] S. U. R. Malik, S. U. Khan, and S. K. Srinivasan, "Modeling and Analysis of State-of-the-art VM-based Cloud Management Platforms," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 50-63, 2013.

[7.2] 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. 64-77, 2013.

[7.3] S. U. R. Malik, S. K. Srinivasan, S. U. Khan, and L. Wang, "A Methodology for OSPF Routing Protocol Verification," *in 12th International Conference on Scalable Computing and Communications (ScalCom)*, Changzhou, China, Dec. 2012.

[7.4] L. Wang and S. U. Khan, "Review of Performance Metrics for Green Data Centers: A Taxonomy Study," *Journal of Supercomputing*, vol. 63, no. 3, pp. 639-656, 2013.

[7.5] S. Zeadally, S. U. Khan, and N. Chilamkurti, "Energy-Efficient Networking: Past, Present, and Future," *Journal of Supercomputing*, vol. 62, no. 3, pp. 1093-1118, 2012.

[7.6] J. Kolodziej and S. U. Khan, "Data Scheduling in Data Grids and Data Centers: A Short Taxonomy of Problems and Intelligent Resolution Techniques," *Transactions on Computational Collective Intelligence*, vol. X, pp. 103-119, 2013.

[7.7] K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, U. S. Khan, A. Abbas, N. Jalil, and S. U. Khan, "A Taxonomy and Survey on Green Data Center Networks," *Future Generation Computer Systems*.

[7.8] J. Shuja, S. A. Madani, K. Bilal, K. Hayat, S. U. Khan, and S. Sarwar, "Energy-Efficient Data Centers," *Computing*, vol. 94, no. 12, , 2012, pp. 973-994.

[7.9] J. Koomey, Growth in data center electricity use 2005 to 2010, Oakland, CA: Analytics Press. July, http://www.analyticspress.com/datacenters.html.

[7.10] M. Gupta and S. Singh,"Greening of the Internet,"*Conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 19-26, 2003.

[7.11] M. Gupta and S. Singh, "Using low-power modes for energy conservation in ethernet lans," *26th IEEE international conference on computer communications, INFOCOM* 2007, pp 2451–2455.

[7.12] J. G. Koomey, "Worldwide electricity used in data centers," *Environ Res Lett,* vol. 3, no. 3, pp. 1–8.

[7.13] R. Bolla R. Bruschi C. Lombardo and D. Suino, "Evaluating the energy-awareness of future internet devices," *IEEE 12th international conference on high performance switching and routing (HPSR)*, 2011, pp 36–43.

[7.14] M. Webb, SMART 2020: Enabling the low carbon economy in the information age, *Tech. rep*, Climate Group on behalf of the Global eSustainability Initiative (GeSI).

[7.15] R. Brown, Report to congress on server and data center energy efficiency public law 109–431, *Environ Prot 109:431*.

[7.16] SAS,Step-by-StepProgramming, http://support.sas.com/documentation/cdl/en/basess/58133/HTML/default/viewer.htm#a001360661.htm, accessed Jan. 28, 2014.

[7.17] R. H. Katz, "Tech titans building boom," *IEEE Spectrum*, vol. 46, no. 2, pp. 40-54.

[7.18] L. A. Barroso, "The price of performance," *Queue*, vol. 3, no. 7, 2005, pp. 48-53.

[7.19] W. C. Richard, L. M. William and W. M. Louis, *Theory of Scheduling*, Addison-Wesley Publishing Company.

[7.20] L. Wang, S. U. Khan, and J. Dayal, "Thermal Aware Workload Placement with Task-Temperature Profiles in a Data Center," *Journal of Supercomputing*, vol. 61, no. 3, pp. 780-803.

[7.21] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making Scheduling "cool": Temperature-aware Workload Placement in data centers," *in USENIX*, pp. 61-75, 2005.

[7.22] L. Ramos and R. Bianchini, "C-oracle: predictive thermal management for data centers," *Symposium on High Performance Computer Architecture*, pp. 111–122, 2008.

[7.23] Q. Tang, S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical

approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 11, 2008, pp. 1458–1472.

[7.24] L. Wang, V. Laszewski, G. Dayal, J. He, X. Younge, and T. R. Furlani, "Towards thermal aware workload scheduling in a data center," *International Symposium on Pervasive Systems, Algorithms, and Networks*, pp. 116-122, 2009.

[7.25] E. Masanet, R. Brown, A. Shehabi, J. Koomey, and B. Nordman, "Estimating the energy use and efficiency potential of U.S. data centers," *Proc IEEE,* vol. 99, no. 8, 2011, pp.1440–1453.

[7.26] Y. Cho and N. Chang, "Energy-aware clock-frequency assignment in microprocessors and memory devices for dynamic voltage scaling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 6, 2007, pp. 1030–1040.

[7.27] H. Aydin and D. Zhu, "Reliability-aware energy management for periodic real-time tasks," *IEEE Transactions on Computers*, vol. 58, no. 10, 2009, pp. 1382–1397.

[7.28] P. Choudhary and D. Marculescu, "Power management of voltage/frequency island-based systems using hardware-based methods," *Transactions on VLSI Systems*, vol. 17, no. 3, 2009.

[7.29] E. Kursun and C. Y. Cher, "Temperature variation characterization and thermal management of multicore architectures," *IEEE Micro,* vol. 29, pp.116–126, ISSN 0272-1732.

[7.30] J. X. Yang, "Dynamic thermal management through task scheduling," *IEEE Symposium on Performance, Analysis of Systems and Software*, pp. 191–201, 2008.

[7.31] R. Ayoub and K. Indukuri, "Temperature aware dynamic workload scheduling in multisocket CPU servers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1359 –1372, 2011.

[7.32] A. Lewis and N. F. Tzeng, Thermal-Aware Scheduling in Multicore Systems Using Chaotic Attractor Predictors.

[7.33] A. Merkel and J. Stoess, "Resource-conscious scheduling for energy efficiency on multicore processors," *International European conference on Computer systems*, pp. 153–166. 2010.

[7.34] J. Choi and C. Y. Cher, "Thermal-aware task scheduling at the system software level," *ACM Symposium on Low Power Electronics and Design*, pp. 213–218, 2007.

[7.35] P. Bailis and V. J. Reddi, "Dimentrodon: Processor-level preventive thermal management via idle cycle injection," *In Proc. of the 48th Design Automation Conference (DAC 2011)*, June 2011.

[7.36] D. Karger, C. Stein, and J. Wein, "Scheduling algorithms," *In Algorithms and theory of computation handbook*, pp. 20-20, Chapman and Hall/CRC, 2010.

[7.37] A. Varma, B. Ganesh, M. Sen, S. Choudhury, and B. Jacob, "A control-theoretic approach to dynamic voltage scheduling," *International CCASE*, pp. 255–266, 2003.

[7.38] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, "Power management of datacenter workloads using per-core power gating," *Computer Architecture Letters*, vol. 8, no. 2, 2009, pp. 48–51.

[7.39] M. Annavaram, "A case for guarded power gating for multi-core processors," *In HPCA*, pp. 291-300, 2011.

[7.40] Q. Tang, S. K. Gupta, and G. Varsamopoulos, "Thermal-aware task scheduling for data centers through minimizing heat recirculation," *IEEE International Conference on Cluster Computing,* pp. 129-138.

[7.41] D. G. Feitelson, L. Rudolph, U. Schwiegelshohn, K. C. Sevcik, and P. Wong, "Theory and practice in parallel job scheduling," *In Job scheduling strategies for parallel processing*, pp. 1-34. Springer Berlin Heidelberg, 1997.

[7.42] R. Ayoub, S. Sharifi, and T. S. Rosing, "Gentlecool: Cooling aware proactive workload scheduling in multi-machine systems," *In Proceedings of the Conference on Design, Automation and Test in Europe* pp. 295-298, 2010.

[7.43] J. Moore, J. Chase, and P. Ranganathan, "Weatherman: Auto-mated, online and predictive thermal mapping and management for data centers," *IEEE ICAC*, pp. 155-164, 2006.

[7.44] C. Bash, C. Patel, and R. Sharma, "Dynamic thermal management of air cooled data centers," *Thermal and Thermomechanical Phenomena in Electronics Systems*, pp. 445–452, 2006.

[7.45] Q. Tang, S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Transactions on Parallel and Distributed System*s, vol. 19, no. 11, 2008, pp. 1458–1472.

[7.46] M. Anderson, M. Buehner, P. Young, D. Hittle, C. Anderson, J. Tu, and D. Hodgson, "MIMO robust control for HVAC systems," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, 2008, pp. 475– 483.

[7.47] M. Toulouse, G. Doljac, V. Carey, and C. Bash, "Exploration of a potential-flow-based compact model of air-flow transport in data centers," *American Society Of Mechanical Engineers ASME Conference*, pp. 41–50, 2009.

[7.48] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricitymarket environment," *International Conference on Computer Communications (INFOCOM)*, pp. 1–9, 2010.

[7.49] L. Bressler, Data Summarization Methods in Base SAS Procedures, http://www.lexjansen.com/mwsug/1999/ paper52.pdf, accessed Feb. 17, 2014.

[7.50] SAS, Step-by-StepProgramming, http://support.sas.com/documentaion/cdl/en/basess/58133/HTML/default/ viewer.htm#a001360661.htm, accessed Jan. 28, 2014.

[7.51] H. Lutkepohl, *New introduction to multiple time series analysis*, Springer, 2005.

[7.52] VARMAX Model SAS, http://support.sas.com/documentation/cdl/en/etsug/60372/HTML/default/viewer.htm#etsug_varmax_sect025.htm.

[7.53] Q. Tang, T. Mukherjee, S.K.S. Gupta, and P. Cayton, "Sensor-Based Fast Thermal Evaluation Model for Energy Efficient High-Performance Datacenters," *ICISIP*, Dec. 2006.

[7.54] W. C. Richard, L. M. William and W. M. Louis, *Theory of Scheduling*, Addison-Wesley Publishing Company.

[7.55] M. Harchol-Balter, K. Sigman, and A. Wierman, "Asymptotic convergence of scheduling policies with respect to slowdown," *Performance Evaluation*, vol. 49, no. 1, 2002, pp. 241-256.

8. CONCLUSIONS

Cloud computing has been a mainstream of research over the last decade. Cloud computing promises reliable services that are delivered through the next-generation DCs. Today, the contemporary society relies more than ever on the Internet and cloud computing. Cloud computing has been adopted and is being used in almost every domain of human life. However, the advent and enormous adoption of the cloud paradigm also brings numerous challenges to cloud providers and research community. Data Centers (DCs) constitute the structural and operational foundations of the cloud computing platforms. The legacy DC architectures are inadequate to accommodate the enormous adoption and increasing resource demands of the clouds. The scalability, high cross-section bandwidth, Quality of Service (QoS) guarantees, energy efficiency, and Service Level Agreement (SLA) assurance are some of the major challenges faced by today's cloud DC architectures. Multiple tenants with diverse resource and QoS requirements share the same physical infrastructure offered by cloud providers. Similarly, reliability and robustness are among the mandatory features of cloud paradigm to handle the workload perturbations, hardware failures, and intentional (or malicious) attacks. Cloud infrastructure must ensure robust behavior to deliver the anticipated services and QoS.

In Chapter 3, we studied the structural robustness of the state-of-the-art data center network (DCN) architectures. Our results revealed that the DCell architecture degrades gracefully under all of the failure types as compared to the FatTree and ThreeTier architecture. Because of the connectivity pattern, layered architecture, and heterogeneous nature of the network, the results demonstrated that the classical robustness metrics are insufficient to quantify the DCN robustness appropriately. Henceforth, signifying and igniting the need for new

robustness metrics for the DCN robustness quantification. We proposed deterioration metric to quantify the DCN robustness. The deterioration metric evaluates the network robustness based on the percentage change in the graph structure. The results of the deterioration metric illustrated that the DCell is the most robust architecture among all of the considered DCNs. The DCN robustness analysis revealed the inadequacy of the classical robustness measures for the DCN architectures.

Chapter 4 presented a comparison of the network features of the three well-known DCN architectures namely: (a) Three-Tier , (b) FatTree, and (c) DCell. Moreover, we conducted a connectivity analysis of the considered DCNs. Finally, we proposed $\mu - A2TR$, a novel robustness metric, which is able to characterize network connectivity. It has been observed that, based on several classical robustness features such as density, average nodal degree, spectral radius, algebraic connectivity, average shortest path length, and diameter, the FatTree architecture is the most robust and connected network. However, the connectivity analysis of the DCNs based on the $A2TR$ values in response to three types of node removals (random, nodal degree, and betweenness centrality) demonstrated that the DCell and FatTree are similar in terms of network connectivity in the case of random removals. Nevertheless, as regards to the targeted removals, the FatTree and ThreeTier depicted low network connectivity. From the connectivity analysis, it can be inferred that, although the traditional network features are useful in determining network robustness and connectivity, there is a need for an appropriate connectivity metric. We presented $\mu - A2TR$ and demonstrated its ability to characterize network connectivity. We beleive that the $\mu - A2TR$ metric will help the engineers and research community to design more robust DCNs.

Chapter 5 presented a study of the relationship between the network hierarchy and robustness. Our study revealed a strong correlation between the network hierarchy and robustness. In most of the cases, the higher the network hierarchy, the more vulnerable is the network to targeted attacks. Moreover, it has been observed that the initial robustness estimation of hierarchical networks using the classical robustness metrics may be misleading. The results unveiled that based on the initial network analysis without failure using he classical robustness metrics, the high GRC valued networks may be considered as more robust than the low GRC valued networks. However, when targeted failures are instigated in hierarchical networks, the high GRC valued networks depict more deterioration and very small portion of the nodes remain connected in the largest connected cluster of the network. One the contrary, the low GRC valued robustness retained the network connectivity and undergo low deterioration in case of targeted attacks. Therefore, one may infer that the robustness estimation of hierarchical networks based on the classical robustness metrics may be misleading and inappropriate.

In Chapter 6 we presented a comparison of the major DCN architectures that addressed the issues of network scalability and oversubscription. We simulated the performance of the major DCN architectures in various realistic scenarios under different network configurations. The simulation results showed that the fat-tree based DCN architecture outperformed the DCell and three-tier DCN architectures in terms of average network throughput and packet delay.

Finally, we proposed a Thermal Aware Resource Allocation (TARA) strategy that aims to stabilize the thermal dynamics of the DCs. The job allocation decisions in TARA were made by considering the ambient effect that occurred as a result of job allocations. By doing the aforesaid, the possibility of hotspots were removed that may cause servers to throttle down,

increasing the possibility of failure. Moreover, our proposed strategy was based on the findings of a real DC workload analysis that was performed using three statistical procedures: (a) Mean Procedure, (b) Correlation Procedure, and (c) Vector Autoregressive Moving-average processes with Exogenous Regressors (VARMAX) model. The aforesaid statistical inferences were used to investigate the thermal behavior, relationship, and patterns exhibit by the servers and pods within the DC. To demonstrate the improvements achieved using TARA, we perform a comparative analysis with three classical and two thermal aware scheduling approaches. The comparison results revealed that TARA performs better, in terms of maintaining thermal balance within the pods of DCs.