International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-11, September 2020

Network Performance Comparison of Fat-Tree and BCube Data Center Architecture: Case Study on Government Office Network

Indra Zulardi, Nico Surantha

Abstract: ***The internet now plays crucial roles such as streaming videos, social networking, managing large scientific data. There is a need for high-speed Internet. The volume of data handled by the data centers is increasing very rapidly. Hence, the need to handle data center traffic effectively becomes inevitable. At present, the government office network uses a data center network design based on Fat-Tree that employs high-end IP switches. Unfortunately, the resulting network performance has delays of more than 0.06 ms, throughput of about 500 kbps, and packet loss rate is 2. This research introduces BCube to re-design and improve network performance (i.e., delay, throughput, and packet loss). Through this research, we propose a new data center design in the government office that improves the network performance significantly, with delays less than 0.004 ms, throughput more than 515 kbps, and packet loss rate 0, two–three percent better compared to the Fat-Tree data center network design that contributes significantly to the data center operations in the government office and its overall business performance through the data center network design.***

Keywords: ***Data Center Networks Design, Network Performance, Delay, Throughput, Packet loss***

I. INTRODUCTION

The needs of the data center are driven by the various uses of cloud computing that have led to the use of smartphones and tablets, as well as devices and sensors connected to the internet. Therefore, with more and more data being generated, a data center facility that is capable of handling large amounts of data is required. Some companies such as Facebook, Google, and Amazon already have a massive data center facility to support their services and their colocation facility stores business applications from companies around the world. It takes more than nine hectares to place several rack servers. As predicted by the Cisco Global Cloud Index [1], global data center traffic is rising at an annual rate of 27%, and by the end of 2021, it will have reached 19.5 zettabytes.

Revised Manuscript Received on September 30, 2020.
* Correspondence Author

Indra Zulardi*, Department of Computer Science, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University Jakarta, Indonesia. E-mail: indra.zulardi@binus.ac.id

Nico Surantha, Department of Computer Science, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University Jakarta, Indonesia. E-mail: nico.surantha@binus.ac.id

Current IT trends have led to the concepts of social networking, openness, sharing, collaborations, mobile, easy maintenance, one-click, distribution, scalability, concurrency, and transparency. A data center is a facility used for the placement of several physical server collections, including a virtual machine and data storage system. The development of a data center itself is very advanced, as shown by the establishment of many data center service provider companies. Data center networks are an important part of big data computing and processing systems that require high bandwidth communication and scalability between servers [2]. The government office uses a high-end IP switch/router with a Fat-Tree data center network design at present. The performance of the existing data center network design can only produce about 1.35 terabits, more than 40 terabits of available aggregate bandwidth capacity in the government office data center network, non-uniform bandwidth between data center nodes will complicate the design of the application and limit the overall capacity of the data center. However, there are two perceived shortcomings with the implemented Fat-Tree model that have not been expected by the government office. Based on the KPI performance that has been approved by the government office [3], the data center network is underperforming. Based on data from the government office data center networks that show an average delay of more than 0.006 ms, a throughput of about 500 kbps, and a packet loss rate is 2, this result necessitates research that can contribute to a re-design of the data center to improve network performance. In this research, BCube [3–5] data center network architecture is selected to solve the network performance issue. Some studies regarding data center network design develop data centers based on BCube [6–9]. BCube is a server-centric data center network design in which the server is a critical component for forwarding data, connectivity switches are elementary, servers have better forwarding functions, and bandwidth consumption does not oversubscribe. The server-centric data center topology has high scalability, symmetry, and uniformity [11]. Most server-centric data center designs are defined recursively, which means that high-level structures grow from some of the low-level structures recursively. This kind of topology has advantageous features for designing global layered indices. Hence, this research is focused on improving the network performance at the government office, so that the design of the data center network can be optimal, and the network performance will be improved for the government office's data center. In this work, the researchers will re-design the data center in the government office using a modification of the traditional data center based on BCube, with the following contributory changes:

- Designing a new data center network for the office
- Introducing a new design of the BCube data center network combined with Fat-Tree
- Improving network performance on delay, throughput, and packet loss, in particular

The above contributions will have a high impact on the data center network in the government office. They will offer an excellent network performance and implement a new design of the BCube data center.

II. RELATED WORKS

Research by [8] and [12] introduced a new method known as BCube Connected Crossbars (BCCC), incorporating the advantages of server-centric networks while removing BCube's limitations. BCCC can be built using dual-port commodity servers and switches, making it a cost-effective network solution for large-scale data centers. Another feature of BCCC that it is easy to expand from an existing data center to a larger data center without modifying the existing topology. These two research studies addressed three key points. First, they introduced how to build BCCC and provided detailed schemes for servers and switches in BCCC. Second, they proposed three efficient routing algorithms for unicast, broadcast, and multicast. Third, they provided a comprehensive comparison between BCCC and some famous structures in terms of design and capital expenditure.

Another research [10] has proposed a cost-effective data center with good scalability named Hyper-BCube by combining DCell and BCube designs to overcome the limitations of each design. A study [13] on the classic design of BCube explained that all servers are not connected directly to other servers. This is also valid for switches; that is, switches are not connected to other switches. By placing the switch in the middle of the server, communication can be achieved between the servers. Judging from the classic design of this BCube, the interconnection bandwidth (IBW) is very low, resulting in the data flow from different nodes on the data center network, such as server to server (srv-srv), switch to switch (swc-swc), and server to switch (srv-swc) and vice versa. On the other hand, the enterprise switch provides many one-gigabit ports and several 10-gigabit ports at high speed. It should be noted that the classic BCube design uses only a one-gigabit port. With the modification proposed by researchers in the classic BCube design, one-gigabit ports are replaced by 10 gigabits at high speed, which will be used for interconnection between different switch layers, modified in various horizontal, vertical, and hybrid forms. This modification only adds additional links without adding a switch or router, to increase the IBW between switches on the classic BCube design. The performance is improved with the new design (horizontal and hybrid-BCube) compared to the classic BCube design.

Many research efforts focused on evaluating DCNs architectures using IP communication. Fat-Tree and BCube architectures have been evaluated in a study [14] using the nix-vector protocol [15] with IP addresses. BCube performs better than Fat-Tree in both metrics average packet delay and average throughput due to its network architecture, characterized by the presence of several possible paths for sending traffic flows.

A method [16] is focused on aggregating an uncertain incast transfer and minimizing the amount of caused network traffic. Prior approaches, relying on deterministic incast transfers, remain inapplicable. A review of this paper takes the first step toward the study of aggregating uncertain incast transfers and proposing efficient approaches from two aspects, that is, the initialization of uncertain senders and incast tree building.

A study based on a comparison between Fat-Tree and BCube [17] conducted performance evaluations between the two using the IPv4 protocol. The results of the research indicated that BCube architecture has a better performance than the Fat-tree for throughput and packet delay.

Studies with data center architecture performance analysis using content-centric communication (CCN) [18] performed comparisons between NDN and IP-based communication and found that Fat-Tree can offer a better communication performance. Their simulation results showed that NDN outperforms IP in terms of average packet delay and throughput. This is mainly due to the in-network cache mechanism allowed by the content store (CS) structure held on each content node.

A research focused on a comparison between Fat-Tree, DCell, and BCube [19] and demonstrated that BCube recovers better if the connection fails, but DCell is better if all the switches are down. For the failure of the connection, BCube retains at least 84% of its servers when 40% of its connections are broken, compared to 74% in DCell since BCube uses a redundant interface system as a server-based network.

Throughput is an important indicator of the quality of network performance. The throughput shows the success rate of sending packets from one point on the network to another. [20].

III. DATA CENTER NETWORKS DESIGN

A. Fat-Tree Data Center Network Design

A fundamental class of universal routing network called Fat-Tree [21] allows for the interconnection of commodity switches and hosts. Several decades ago, Charles Clos designed a network topology known as the Clos network that ensures high bandwidth for end devices using smaller commodity switches that are interconnected together [22]. The Fat-Tree data center network design adopted by *Al-Fares et al*. [4] can be seen as an interplay of these two fundamental concepts. In Fat-Tree [4], network elements follow the hierarchical organization of network switches in four layers, namely, core switch, aggregation switch, edge switch, and end-host layers. In a *k-ary* Fat-Tree, there are $(k/2)^2$ *k*-port switches in the core switch layer. Below the core switch layer, there are *k* pods, where *k* = recursion depth, enumerated from pod 0 to pod *k* - 1.

Each core switch has one port that is connected to each of the *k* pods.

Within each pod, there is an aggregation switch layer and edge layer, each containing *k/2* switches.

Each *k*-port switch in the edge layer is connected to *k/2* end hosts.

The remaining *k/2* ports of each edge switch are connected to the *k/2* of the *k* ports of each aggregation switch. The number of end hosts that the architecture can support is $k^3/4$.

The Fat-Tree data center network design also incorporates several improvements to achieve better performance and fault-tolerance, such as two-level table routing, flow scheduling, flow classification, and *Bidirectional Forwarding Detection*. An illustration is depicted in Figure 1 with k = 4.

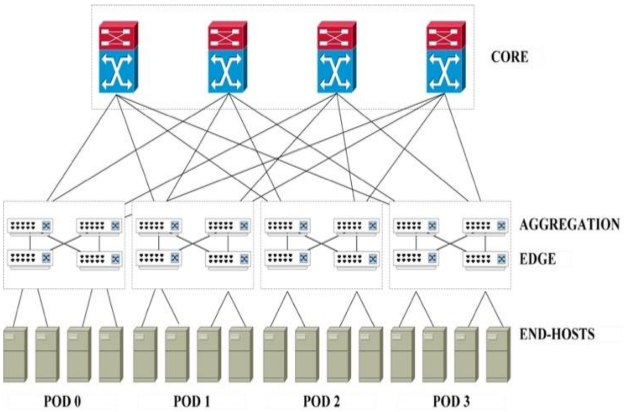

Figure 1: Fat-Tree Data Center Network Design with k = 4

B. BCube Data Center Network Design

The BCube data center network design is a recursive structure by nature [5]. For example, a $BCube_1$ is constructed from *n* $BCube_0$ and *n* n-port switches.

In a more general form, a $BCube_k$ ($k \geq 1$) is constructed from *n* $BCube_{k-1}{}^S$ and n^k n-port switches. A BCube0 contains *n* servers that are connected to an *n*-port switch.

For each server in a $BCube_k$, there are $k + 1$ ports, enumerated from level-0 to level-*k*.

A $BCube_k$ has $N = n^k + 1$ servers and $k + 1$ level of switches. Each level contains n^k *n*-port switches. The switches are denoted in the form of $< l, s_{k-1}\, s_{k-2} \dots s_0 >$, where $l (0 \leq l \leq k)$ is the level of the switch and ($s_j \in [0, n - 1]$, j 2 [0, k - 1]) allows the enumeration of a unique address for each switch.

Each server in a $BCube_k$ is enumerated using the addressing scheme as $a_k, a_{k-1} \dots a_0$ ($a_i \in [0, n - 1], i \in [0, k]$). To construct a $BCube_k$, first, enumerate the n $BCube_{k-1}{}^S$ from 0 to $n^k - 1$.

Next, enumerate the servers in each $BCube_{k-1}$ from 0 to $n^k - 1$, as the number of servers is equal to n^k.

To connect the switches to the servers, the level-*k* port of the *i*-th server ($i \in [0, n^k - 1]$) in the *j*-th $BCube_{k-1}$ ($j \in [0, n^k - 1]$) is connected to the *j*-th port of the *i*-th level-*k* switch.

The core of the BCube data center network design is its server-centric network structure, where servers with multiple network ports connect to multiple layers of switches.

Servers act not only as end hosts but also relay nodes for each other.

In Figure 2, the BCube data center network design with n = 4 and k = 1 is illustrated.

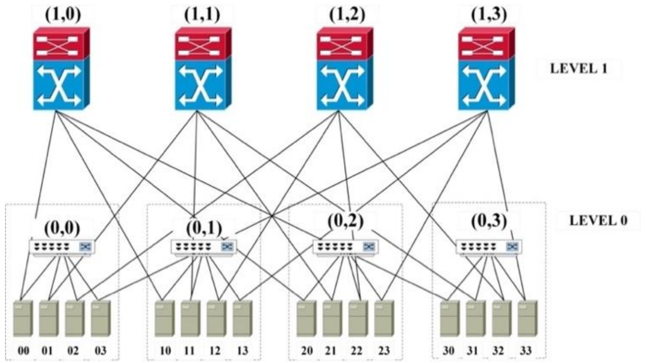

Figure 2: BCube Data Center Network Design with n = 4 and k = 4

IV. RESEARCH METHODOLOGY

This research methodology includes the re-design and simulation of the new data center network design, consisting of four phases.

The first phase is the current data center network analysis and the second phase is the literature of studies on and data collection of the new design data center network in the government office based on BCube.

The third phase is the process of designing a new data center network based on BCube. Finally, the last phase is the simulation and evaluation of the data center network in the government office based on BCube. The diagram in Figure 3 illustrates how this research process works.

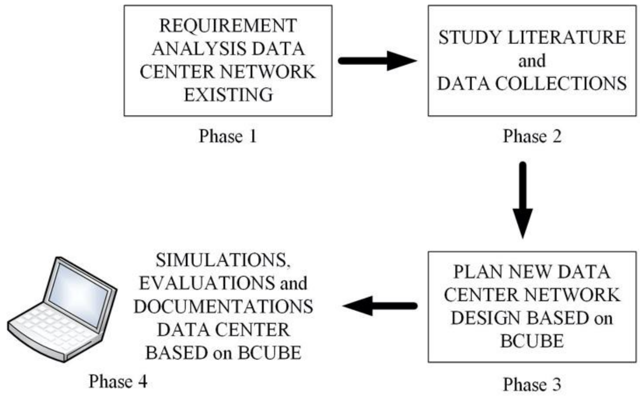

Figure 3: Research Methodology

A. Requirement Analysis Data Center Network Existing

The initial phase of this research starts with an analysis of the existing data center network requirements, improving communication between servers and switches, and vice versa, and determining the technical objectives. The reciprocity of this research is to design a data center that has high availability and improved network efficiency for the government office. This data center currently uses the Fat-Tree design.

B. Study Literature and Data Collection

The second phase in this research, literature analysis, is conducted to collect the data needed from numerous studies on the BCube data center network design.

Further, the average data delay, packet loss, and throughput from the current data center network in the government office were included in the data collection.

C. Plan of New Data Center Network Design Based on BCube for the Government Office

The third phase is the process of redesigning the current data center network in the government office that is based on Fat-Tree to replace it with a data center network design based on BCube. This phase involves selecting the flow of data traffic used during the simulation.

D. Experiment on and Evaluation of Data Center Based on BCube

The final phase of this research involves experimenting with and evaluating the new data center network based on BCube using the ns-3 [23] software. The specifications of the PC used are stated in Table 1.

Table 1: PC specification

No	Hardware	Specification
1	Memory	16 GB 2400 MHz DDR4
2	CPU	Intel Core i7-8750, 2.20 GHz Gen 8
3	Graphics	Intel UHD Graphics 630 1536 MB
4	Operating System	Ubuntu 18.04 64 bits

V. EXPERIMENTS

A. Experimental Design

The first trial in the design process for the new data center is conducted on the ns-3 simulator. It begins by setting the network data center architecture, either Fat-Tree or BCube, and further attempting to add several links to the switch from each server, and vice versa. The researchers measure the network output (delay, packet loss, and throughput). The delay is defined as the result of the difference between the time received and delivery, while packet loss is the class that counts the number of packets lost. This class records the packets lost in a sequence number using the client/server transmission. Any packets beyond a given timeframe are deemed lost. Throughput is the number of packets received at a specific time. The simulation flow of this research is shown in Figure 4.

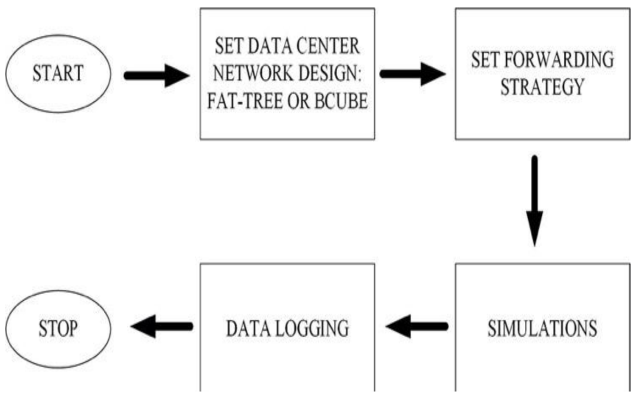

Figure 4: Simulation Flow

As shown in Figure 5, additional links from each server connect with the top-of-rack (ToR) switch. Data traffic on a data center is expected to increase the network efficiency of the data center with additional connections. In this simulation, the BCube design runs by sending information to the BCube ID. The source obtains the routing path based on the source-destination BCube IDs and writes all path information into the packet header. It is submitted based on the path information embedded in the packet header. In this scenario, the server/host in the architecture of the BCube data center becomes very dominant compared to the switch.

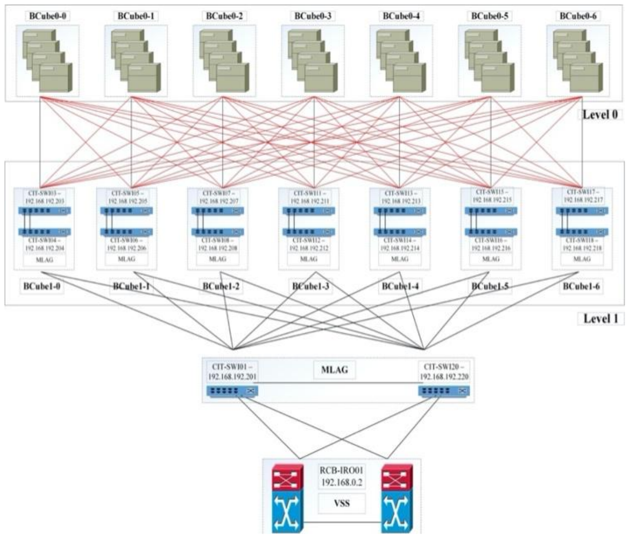

Figure 5: New Data Center Network Design Based on BCube in the Government Office

B. Experimental Simulation Results

The researchers use the ns-3 software in this simulation and run with the metrics shown in Table 2.

Table 2: Comparison of the Simulation Setting for the Fat-Tree and BCube Data Center Network Designs

Parameters	Fat-Tree	BCube
Simulation time	100 sec	100 sec
Packet size	1024 bytes	1024 bytes
Data rates between switches	1 Mbps	1 Mbps
Communication pattern	Random selection of two hosts and sending data between them	Random selection of two hosts and sending data between them
Traffic flow pattern	Exponential random	Exponential random
Routing protocol	Nix-vector protocol	Nix-vector protocol
Variable parameter	n varied (2–12)	K = 2, n varied (2–12)

The average delay simulated in the design of the Fat-Tree data center network is observed to increase the first time and the average delay is 0.00682375 ms with switches (k) = 2 and servers = 2. On the other hand, the average delay in the design of the BCube data center network is 0.00288266 ms and so on. The delay from BCube is more stable than Fat-Tree. Similar to Fat-Tree, the delay will increase at the beginning of the simulation as the host will first scan the server information. However, BCube does have a smaller delay than Fat-Tree. The average packet delay in the BCube design is 0.004 ms based on the measurement results. The simulation results using ns-3 are summarized in Table 3.

Table 3: Comparison of the Simulation Result of Delay, Packet Loss, and Throughput of the Fat-Tree and BCube Data Center Network Designs

Data Center	Switches	Servers	Average Delay (ms)	Average Packet loss rate	Average Throughput (kbps)
Fat-Tree	2	2	0.00682375	0	494.714
	4	16	0.00635411	0	503.975
	6	54	0.00669762	0	505.415
	8	128	0.00664817	0	513.236
	10	250	0.00659701	0	509.127
	12	432	0.00672581	2	514.322
BCube	2	2	0.00288266	2	509.828
	4	16	0.00373647	0	515.159
	6	54	0.0044258	0	515.487
	8	128	0.00486099	0	509.772
	10	250	0.00491856	0	510.288
	12	432	0.00499315	0	513.938

The average delays in the Fat-Tree and BCube data center network designs are shown in Figure 6.

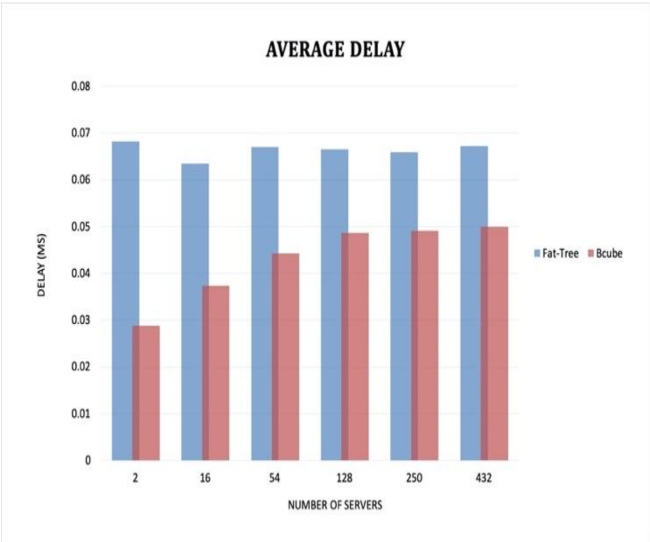

Figure 6: Comparison of the Fat-Tree and BCube Data Center Network Designs for Average Delay

BCube's average delay packet produces fewer values than the Fat-Tree, indicating that BCube's data transfer is faster compared to Fat-Tree. Meanwhile, the packet loss rate observed using Fat-Tree is 2.

It is high and increases with the number of servers. In the case of BCube, the loss of packets is 0, which is expected to enhance the performance.

This appears to be the expected behavior because each pair of nodes has a particular path in Fat-Tree, and thus, the path will be less dense.

From the simulation results, the throughput in BCube reaches 515.159 kbps with switches = 6 and servers = 54 while in Fat-Tree, it reaches 505.145 kbps.

The Fat-Tree and BCube simulations are started with switches = 2, servers = 2, level 1, and so on.

Figure 7 shows that the average throughput seems to be constant between Fat-Tree and BCube.

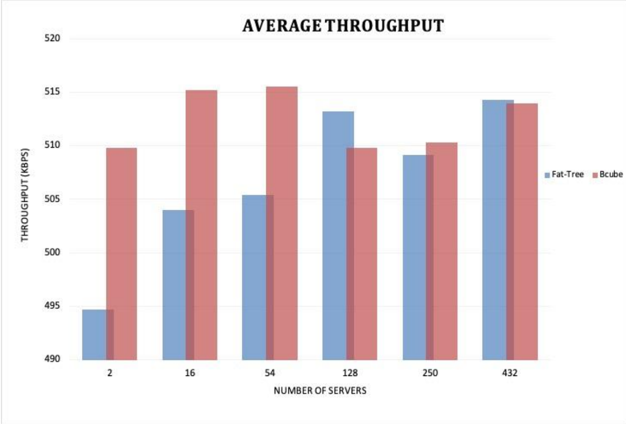

Figure 7: Comparison of the Fat-Tree and BCube Data Center Network Designs for Average Throughput

As the number of servers increases, the throughput increases. One of the reasons why BCube performs better in all the network performance parameters is because its network architecture has multiple possible paths to send a network element traffic flow. This will bring about lesser traffic congestion and make more bandwidth available. Additionally, servers in the BCube data center network design act as relay nodes and help each other to speed up packet routing and traffic flow, thus resulting in improved network performance in comparison to the Fat-Tree data center network design. As for the Fat-Tree data center network design, the consistent overall performance is due to the fundamental properties of Fat-Tree networks that were theoretically proven to be very efficient for the network interconnection. One may also argue that to a large extent, the network performance of both data center network designs is independent of the size of the data center networks. The results in the present study have shown that the BCube data center network design performs slightly better than Fat-Tree in terms of average delay and packet loss, while for the seems to be fairly constant.

VI. CONCLUSION

This research compares the current network architecture of the data center network in the government office that is based on Fat-Tree with BCube. Through this research, the comparative network performance of Fat-Tree and BCube data center network architectures is reported. The simulation result shows that the average delay and packet loss is two three percent and offers better performance. The average throughput seems to be relatively constant between BCube and Fat-Tree.

ACKNOWLEDGMENT

This research was supported by Bina Nusantara University. We would like to thank the government office for giving us the opportunity to improve the network performance of the data center network design.

REFERENCES

1. Cisco System Inc., "Global Cloud Index Projects Cloud Traffic to Represent 95 Percent of Total Data Center Traffic by 2021," *Cisco System Inc.*, p. 4, 2018.

2. D. Li, Y. Shen, and K. Li, “Length Shuffle: Achieving high performance and flexibility for data center networks design,” *Computer Communications*, vol. 111, pp. 142–152, 2017. https://doi.org/10.1016/j.comcom.2017.08.001
3. R. C. Bahrain, *KPI Performance Strategy v1.4.pdf*. Bahrain.
4. M. Al-fares, A. Loukissas, and A. V. A. Scalable, “Commodity Data Center Network Architecture,” *SIGCOMM Comput. Commun. Rev*, vol. 38, no. 4, pp. 63–74, 2008. https://doi.org/10.1145/1402946.1402967
5. C. Guo *et al.*, “BCube: A high performance, server-centric network architecture for modular data centers,” *Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009. https://doi.org/10.1145/1594977.1592577
6. Z. Han and L. Yu, “A Survey of the BCube Data Center Network Topology,” *Proceedings - 4th IEEE International Conference on Big Data Security on Cloud, BigDataSecurity 2018, 4th IEEE International Conference on High Performance and Smart Computing, HPSC 2018 and 3rd IEEE International Conference on Intelligent Data and Securit*, pp. 229–231, 2018. https://doi.org/10.1109/BDS/HPSC/IDS18.2018.00056
7. T. Pan, B. Cheng, J. Fan, C. K. Lin, and D. Zhou, “Toward the completely independent spanning trees problem on BCube,” *2017 9th IEEE International Conference on Communication Software and Networks, ICCSN 2017*, vol. 2017-Janua, pp. 1103–1106, 2017. https://doi.org/10.1109/ICCSN.2017.8230281
8. Z. Li, Z. Guo, and Y. Yang, “BCCC: An Expandable Network for Data Centers,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3740–3755, 2016. https://doi.org/10.1109/TNET.2016.2547438
9. C. Jia, H. Wang, and L. Wei, “Study of Smart Transportation Data Center Virtualization Based on VMware vSphere and Parallel Continuous Query Algorithm over Massive Data Streams,” *Procedia Engineering*, vol. 137, pp. 719–728, 2016. https://doi.org/10.1016/j.proeng.2016.01.309
10. D. Lin, Y. Liu, M. Hamdi, and J. Muppala, “Hyper-BCube: A scalable data center network,” *IEEE International Conference on Communications*, pp. 2918–2923, 2012. https://doi.org/10.1109/ICC.2012.6363759
11. X. Wang, J. X. Fan, C. K. Lin, J. Y. Zhou, and Z. Liu, “BCDC: A High-Performance, Server-Centric Data Center Network,” *Journal of Computer Science and Technology*, vol. 33, no. 2, pp. 400–416, 2018. https://doi.org/10.1007/s11390-018-1826-3
12. Z. Li and Y. Yang, “Permutation generation for routing in bcube connected crossbars,” *IEEE International Conference on Communications*, vol. 2015-Septe, pp. 5460–5465, 2015. https://doi.org/10.1109/ICC.2015.7249192
13. M. Asghari, Vahid; Farrahi, Moghaddam Reza; Cheriet, “Performance analysis of modified BCube topologies for virtualized data center networks,” *Computer Communications*, vol. 96, pp. 52–61, 2016. https://doi.org/10.1016/j.comcom.2016.10.001
14. D. Wong, K. T. Seow, C. H. Foh, and R. Kanagavelu, “Towards reproducible performance studies of datacenter network architectures using an open-source simulation approach,” *GLOBECOM - IEEE Global Telecommunications Conference*, pp. 1373–1378, 2013. https://doi.org/10.1109/GLOCOM.2013.6831265
15. NS-3, “Nix Vector Routing.” [Online]. Available: https://www.nsnam.org/docs/release/3.18/doxygen/group__nixvectorrouting.html. [Accessed: 01-Mar-2020].
16. D. Guo, “Aggregating Uncertain Incast Transfers in BCube-Like Data Centers,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 934–946, 2017. https://doi.org/10.1109/TPDS.2016.2612660
17. Mardiyansyah, Ariani, Q. Ridho, R. Harwahyu, and R. F. Sari, “Performance Comparison of BCube and Fat Tree Topology Data Center Network using Named Data Networking(NDN),” *Proceedings - 1st International Conference on Informatics, Multimedia, Cyber and Information System, ICIMCIS 2019*, pp. 73–78, 2019. https://doi.org/10.1109/ICIMCIS48181.2019.8985204
18. I. Dhiab, Y. Barouni, S. Khalfallah, and J. B. H. Slama, “Datacenter network architecture performance analysis using content centric communication,” *2016 International Symposium on Networks, Computers and Communications, ISNCC 2016*, 2016. https://doi.org/10.1109/ISNCC.2016.7746119
19. R. de S. Couto, S. Secci, M. E. M. Campista, and L. H. M. K. Costa, “Reliability and Survivability Analysis of Data Center Network Topologies,” *Journal of Network and Systems Management*, vol. 24, no. 2, pp. 346–392, 2016. https://doi.org/10.1007/s10922-015-9354-8
20. L. Rivenes, “What is Network Throughput?” 2016. [Online]. Available: https://datapath.io/resources/blog/what-is-network-throughput/. [Accessed: 07-Jun-2020].
21. C. E. Leiserson, “Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing.,” *Proceedings of the International Conference on Parallel Processing*, pp. 393–402, 1985. https://doi.org/ 10.1109/TC.1985.6312192
22. B. C. Clos, “A Study of Non-Blocking Switching,” pp. 406–424, 1952. https://doi.org/10.1002/j.1538-7305.1953.tb01433.x
23. NS-3, “NS-3.” [Online]. Available: https://www.nsnam.org/about/. [Accessed: 15-Feb-2020].

AUTHORS PROFILE

Indra Zulardi received B.Sc. degree in Industrial Engineering from Bandung Islamic University, Bandung, Indonesia in 2002. He is currently pursuing a master’s degree in Bina Nusantara University, Indonesia. Indra Zulardi has also been working at Royal Court Kingdom of Bahrain as a IT Network Specialist, Kingdom of Bahrain.

Nico Surantha currently serves as an assistant professor in Computer Science Department, Binus Graduate Program, Bina Nusantara University. His research interest includes wireless communication, health monitoring, network design, digital signal processing, system on chip design, and machine learning. He is an IEEE member. Currently, Nico Surantha and his team in Binus University are working on smart health monitoring prototype development. They developed machine learning based techniques to analyze heart signal. Then, they implement the developed algorithm into IoT prototype. Nico Surantha has also been working on SoC development for WiFi based system, especially for industrial wireless system. For this project, he collaborates with Kyushu Institute of Technology Japan.