POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.gov.tr/politeknik

Study on data center network topologies for monitoring data using edge/fog computing

Uç/sis bilişim kullanarak verileri izlemeye yönelik veri merkezi ağ topolojileri üzerine çalışma

Yazar(lar) (Author(s)): *Pedro Juan ROIG[1], Salvador ALCARAZ[2], Katja GILLY[3], Cristina BERNAD[4], Carlos JUIZ[5]*

ORCID[1]: 0000-0002-8391-8946

ORCID[2]: 0000-0003-3701-5583

ORCID[3]: 0000-0002-8985-0639

ORCID[4]: 0000-0001-9537-415X

ORCID[5]: 0000-0001-6517-5395

To cite to this article: Roig P.J. et al., "Study on Data Center Network Topologies for Monitoring Data using Edge/Fog Computing", ***Journal of Polytechnic***, 27(5): 1859-1874, (2024).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Roig P.J. et al., "Study on Data Center Network Topologies for Monitoring Data using Edge/Fog Computing", ***Politeknik Dergisi***, 27(5): 1859-1874, (2024).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1327987

Study on Data Center Network Topologies for Monitoring Data using Edge/Fog Computing

Highlights

- *Edge/Fog Computing*
- *Fault Diagnosis and Prediction*
- *Industry 4.0/5.0*
- *Internet of Things*
- *Sensor Data Fusion*

Graphical Abstract

Data center network topologies are classified into switch-centric and server-centric, where Fat Tree and Leaf & Spine are common instances of the former and BCube, DCell, and FiConn are common examples of the latter. In this paper, the performance of such topologies have been measured in terms of number of hops away between a given pair of hosts, where a statistical point of view has been taken in order to obtain centralization and dispersion measurements for each design.

Table. Statistics related to hop count for commonly used data center network topologies

Topology	Average (μ)	Mode	Median	Variance (σ^2)	Stand.Dev. (σ)	Coeff.Variation
Fat Tree (*k=4*)	5.47	6	6	1.32	1.15	0.21
Leaf & Spine (*8-2*)	3.87	4	4	0.25	0.50	0.13
Bcube_1 (n=4)	3.20	4	4	0.96	0.98	0.31
DCell_1 (n=4)	3.53	3-5	3	1.52	1.23	0.35
FiConn_2 (n=4)	7.32	8	8	8.09	2.84	0.39

Aim

The focus is to find out the statistical measurements with respect to the minimum number of hops to go from one particular host to any of its peer hosts within each of the five topologies exposed. This way, the results obtained may lead to choose a more convenient topology to better fit the needs of real-time applications, which prefer lower average rates, or otherwise, the needs of streaming traffic, which prefer lower coefficients of variation.

Design & Methodology

The steps of this study are the following: first, to introduce the context, then, to present the topologies selected, after that, to calculate the statistics related to each design, and eventually, to compare the outcome among all layouts.

Originality

To the best of our knowledge, there is no performance study based on statistics on data center designs and focused on minimizing the number of hops away between any pair of hosts within a series of data center network topologies.

Findings

Results show that server-centric instances offer lower average values, whereas switch-centric ones display lower coefficient of variation, considering a similar number of end hosts for the topologies compared.

Conclusion

The outcome obtained for topologies with similar amounts of end hosts displays lower average values for switch-centric designs, namely Fat Tree and Leaf & Spine, whist showing lower coefficients of variation for server-centric designs, namely $BCube_1$ and $DCell_1$. According to this, the values achieved for switch-centric topologies fit better for real-time applications, whereas the values achieved for server-centric topologies fit better for streaming traffic.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Study on Data Center Network Topologies for Monitoring Data using Edge/Fog Computing

Araştırma Makalesi / Research Article

Pedro Juan ROIG[1,2*], Salvador ALCARAZ[1], Katja GILLY[1], Cristina BERNAD[1], Carlos JUIZ[2]
[1]Department of Computer Engineering, Miguel Hernández University, Elche, Spain
[2]Department of Mathematics and Computer Science, University of the Balearic Islands, Palma de Mallorca, Spain
(Geliş/Received : 19.07.2023 ; Kabul/Accepted : 24.10.2023 ; Erken Görünüm/Early View : 06.12.2023)

ABSTRACT

The election of an appropriate data center network topology is key when dealing with surveillance and monitoring processes, such as those devoted to obtaining relevant data for sensor data fusion in any type of remote computing environment so as to perform fault diagnosis and prevention. In this paper, some of the most commonly used topologies to interconnect nodes within a data center bound to edge/fog computing, representing either switch-centric ones or server-centric ones, are reviewed and analyzed from a statistical point of view in order to measure their performance, resulting in server-centric ones doing it better.
Keywords: Edge/Fog Computing, fault diagnosis and prediction, industry 4.0/5.0, internet of things, sensor data fusion.

Uç/Sis Bilişim Kullanarak Verileri İzlemeye Yönelik Veri Merkezi Ağ Topolojileri Üzerine Çalışma

ÖZ

Uygun bir veri merkezi ağ topolojisinin seçilmesi, sorun teşhisi ve önlenmesi gerçekleştirmek amacıyla herhangi bir uzaktan bilişim ortamından elde edilen sensör verilerinin birleştirilmesi için gözetim ve izleme süreçleriyle uğraşırken çok önemlidir. Bu çalışmada, Switch merkezli veya sunucu merkezli olanları temsil eden uç/sis bilişime bağlı bir veri merkezi içindeki bileşenleri birbirine bağlamak için en yaygın kullanılan topolojilerden bazılarının performanslarını ölçerek gözden geçirilmiş ve istatistiksel bir bakış açısıyla analiz edilmiştir ve sonuç olarak sunucu merkezli olanların daha iyi performans gösterdiği bulunmuştur.
Anahtar Kelimeler: Uç/Sis bilişim, sorun teşhisi ve tahmini, endüstri 4.0/5.0, nesnelerin interneti, sensör veri füzyonu.

1. INTRODUCTION

The arrival of the diverse industrial revolutions opened up disruptive transformations in multiple fields, not only from a technological perspective, but also from an economic and social points of view [1]. These are the major milestones:

- Industry 1.0 took place in the late 18th century. It brought mechanization through the use of water and steam to enable mass production.
- Industry 2.0 occurred in the late 19th century. It brought electrification through the use of electric power and assemble line production.
- Industry 3.0 happened in the 1970s. It brought automation through the advances in electronics and IT, setting the focus on automating processes [2].
- Industry 4.0 arose in the early 21st century. It brought digitalization powered on information and communication technologies through the advances in networking. This is the current stage we are in and some of its main features are smart manufacturing, cyber-physical systems and Internet of Things, as well as predictive maintenance [3].
- Industry 5.0 is expected to come in the 2020s. It will bring personalization through mass customization, as well as the collaboration between humans and machines thanks to artificial intelligence [4].
- Industry 6.0 is still in its first stages and it is expected to be ready by the middle of the 21st century. It will bring hyperconnectivity and sustainability by means of antifragile manufacturing and the use of quantum technology [5].

Focusing on the current paradigm, which is Industry 4.0, nine pillars were defined at the beginning, although some of those have been redefined [6]:

- Cyber-Physical Systems (CPS).
- Internet of Things (IoT).

** Corresponding Author*
e-mail : proig@umh.es

- Big Data and Data Analytics, thanks to the use of Machine Learning techniques.
- Cloud and Information Technology, bringing together IP and IT.
- Robots and Automated Machinery.
- 3D Printing, also known as Additive Manufacturing.
- Simulation, which is a combination of Isolation and Recreation, bringing about Digital Twins.
- Portable Devices, which allow the development of Augmented / Virtual / Mixed Reality (AR/VR/MR).
- Cybersecurity.

Furthermore, the goals of Industry 4.0 are tied to the concepts of resilience and sustainability as a founding philosophy, where the former stands for fault-tolerant solutions, while the latter does for suitable environmental-friendly solutions, such as waste control, carbon neutrality or resource optimization [7].

In this paper, surveillance and monitoring of data using edge/fog computing is going to be confronted from the point of view of data center network architectures. This way, some typical data center topologies fit for edge/fog computing are going to be compared in order to find the best features related to delay and jitter.

With respect to the contribution to science of this paper, the focus is set on analyzing some data center network topologies so as to measure the performance achieved from a statistical viewpoint. It is to be noted that there are many studies published in the literature regarding data center network architectures, where the most typical features used to confront them are throughout and latency among links connecting end hosts.

However, the approach taken herein lays on finding out the statistical measurements related to both centralization and dispersion with respect to the minimum number of hops to get from a given host to any other host within a particular topology, considering all links with the same features. This way, the values obtained represent an alternative way to measure performance of a data center, which may gear the decision making process when it comes to choosing a network topology for a data center design.

Regarding the experimental results obtained, it is to be considered that server-centric topologies present better results compared to their switch-centric counterparts when the number of end hosts is similar among them. However, the implementation of the former is more complex as the forwarding process requires further resources because each host has more than one network, whilst the latter only have connection per host [8].

There are different papers in the literature within this field devoted to compare features of different network architectures in data centers. In this sense, Couto et al. [9] carry out an analysis of Fat Tree, BCube and DCell from the point of view of reliability and survivability, concluding that BCube gives the best results when it comes to link failures, whereas DCell gives the best outcome when it comes to switch failures.

On the other hand, Cortes et al. [10] compare different features like throughput, latency, scalability or reliability referred to Fat Tree and BCube, getting to the conclusion that the second one achieves better performance with respect to the first one. In this sense, other studies go the same way, such as Alqahtani et al. [11] and Han et al. [12], although both manuscripts propose design modifications in order to boost outcome.

Furthermore, Negara et al. [13] confront BCube and DCell, finding out that the latter presents higher speed for data transmission, whilst the former achieve higher security and integrity in data transmissions. Some attempts have been made to enhance both designs, such as Lin et al. [14] proposes an improvement on the former and Ahmed et al. [15] presents a variation of the latter.

Nonetheless, an alternative approach has been undertaken in this paper compared to those cited above, as it is based on statistical operations which are carried out by undertaking arithmetic operations applied to the number of hops away among hosts, thus accounting for an alternative manner to measure performance.

The rest of the paper is organized as follows: to start with, Section 2 is devoted to operational technology, then, Section 3 is dedicated to the application of blockchain technologies to industry 4.0 and beyond, after that, Section 4 introduces fault diagnosis and detection (FDD), next, Section 5 proposes some data center network architectures fit for surveillance and monitoring of data using Edge/Fog Computing, in turn, Section 6 studies performance in each particular case, and eventually, Section 7 draws some final conclusions.

2. OPERATIONAL TECHNOLOGY

Industry 4.0 deployments facilitate the integration of information technology (IT) and operational technology (OT), thus bringing about several benefits of this convergence such as decision-support, cost optimization or process improvement [16]. Historically, OT standards were defined by the International Electrical Commission (IEC), whilst their IT counterparts were done by different entities, such as the International Telecommunication Union (ITU), the International Standard Organization (ISO) or the IEC itself.

However, in order to achieve convergence in the IT/OT management, different features need to be faced [17]. For instance, cybersecurity on OT needs to be strengthen as it was not conceived to be connected outwards, data handling in OT usually involves small amounts of data, end-to-end management processes must be harmonized, real-time ethernet solutions on OT are needed to meet time constraints and the coexistence of compatible wireless networks for both IT/OT.

Hence, integrating IT and OT in manufacturing industries brings clear advantages to the factory performance and management, where the former is responsible for the creation, storage and secure data, whilst the latter basically focuses on processes happening in the physical world, thus resulting in automated factory floor, automated workflow, predictive analytics or machine vision [18]. Such IT/OT convergence leads to the concept of Industrial Internet of Things (IIoT), which refers to the use of IoT technology in industries to enhance productivity and efficiency in manufacturing and industrial processes [19].

The security aspect of converging IT/OT is a key point as it is to be noted that the hardware and software related to automation and control systems such as Supervisory Control and Data Acquisition (SCADA), Distributed Control Systems (DCS) or Industrial Control Systems (ICS) are traditionally included into OT, which are often devoted to manage critical infrastructures [20]. Hence, the IT/OT integration makes them exposed to typical cyberattacks being perpetrated on IT infrastructures, thus they must be duly protected.

Besides, IT/OT convergence may be applied to multiple fields where a set of systems is uniformly managed, such as smart buildings, where a fully automated environment is composed of energy consumption, access control and HVAC [21], or the digital transformation in the construction industry, where humans and automatic machines work together [22].

An important concept of Industry 4.0 is that of cyber-physical systems (CPS), where a seamless automatic connection is established between the physical world and some digital components, which are able to interact and control the material world through the use of networking. The construction of CPS systems may be divided into 5 levels, that being known as a 5C architecture for implementation purposes [23]:

- Connection, such as plug & play connectivity or sensor networks.
- Conversion, such as data analytics, predictions or correlations.
- Cyber, such as digital twins, model-based reasoning or data mining.
- Cognition, such as decision-making, diagnostics or machine learning.
- Configuration, such as self-adjustment and self-optimization for variation.

Eventually, the rapid transformation due to the development of AI-based solutions such as ChatGPT3 [24] or LaMDA [25], will not only result in widespread digitalized and sustainable world [26], but also in the adoption of Industry 5.0 shortly thanks to the drive of personalization. This will be featured in customized manufacturing and empowering humans in manufacturing processes in order to attain a real collaboration between humans and machines, where the focus is not set in technology but on human-centricity, environmental stewardship and social benefit [27].

3. BLOCKCHAIN IN INDUSTRY 4.0

Blockchain may be defined as a digital ledger that permits to capture transactions carried out among several parties on real-time, where at the same time it acts as a decentralized database where every member may obtain an identical copy of the whole ledger [28]. This allows blockchain technology to be employed in many fields for the purposes of verification, identification and secure storing [29].

One of the key benefits of data registered in blockchain is its immutability, taken as the ability of a ledger to remain a permanent and unalterable history of transactions, which makes the process of auditing efficient and trustworthy [30]. It boosted the rise of smart contracts, those being decentralized applications without the need of a trusted third party, which are computer algorithms designed to enforce and verify negotiation and agreement among different untrusted parties [31].

The characteristics of blockchain technology make it an interesting solution for its use as a decentralized and distributed way to guarantee security requirements in IoT and IIoT environments [32]. Blockchain may contribute to IIoT by providing enhanced interoperability, confidentiality, safety, dependability and measurability. The combination of both concepts is often known as Blockchain for Industrial Internet of Things (BIIoT). In this sense, BIIoT 1.0 is referred to cryptocurrency and expense, BIIoT 2.0 is related to automatic digital economy by means of smart contracts and BIIoT 3.0 is devoted to the requirements of the digital world, like Industry 4.0 or smart cities [33].

Blockchain technology may be implemented in different ways, such as a public blockchain, which allows an easy access to anyone with an internet connection, where cryptocurrencies like bitcoin or ethereum are the most well known examples. Besides, a private blockchain may only be accessed through an invitation, which is issued by a central entity managing all actions, where hyperledger-fabric and private ethereum are common examples. Moreover, a hybrid blockchain tries to get the best of both models, where resources are managed by a central entity and users collaborate in its maintenance and security, although transactions are visible for public users, meaning that ledgers may be examined block by block [34].

It may be said that blockchain eliminates the need of a trusted third party, allowing for transactions to directly flow between a sender and a receiver. A bunch of transactions are included in a block and all blocks forming the blockchain are linked together by means of a cryptographic function, which avoids the forge of previous blocks [35]. Hence, blockchain may be applied to industrial processes to convey both a secure environment and high reliability [36].

Regarding Industry 4.0, one of the main advantages of blockchain technology is to deal with traceability and quality control [37], as it allows to achieve a secure supply chain, which is key in some business as those related to food, pharmacy or any kind of exports. This way, it is always available where a product comes from, where it is located or remove it upon request [38]. Besides, the authenticity of a product is easily verified, thus avoiding forgery. Moreover, it also helps reduce bureaucracy, thus improving the management of certificates and the fulfillment of standards, whilst also increasing the accuracy on inventory management [39].

Another positive aspect is related to payment management, as it permits to automate invoices and supplier payments [40]. Also, it is useful when it comes to assure buying and selling processes, which is done by attaching a GPS with of the blockchain to the information of the purchasing contract, whereas an automatic payment is made to the vendor and to the carrier on delivery [41]. Additionally, as blockchain provides each person, product or process with a unique digital identity, it simplifies any operation as identification errors are avoided.

Also, another advantage is referred to store and distribute in a secure way all data generated in real time thanks to the machine to machine (M2M) connectivity. This communication does not rely on third parties, thus enhancing synchronization and expedition of processes in a secure environment [42]. Furthermore, the proliferation of smart contracts is ever growing, where the parties involved define the object of the contract and its clauses, which will be activated if, and only if, all prerequisites are met, as they are auto executable [43]. This way, smart contracts may be seen as a connector among IoT, AI and blockchain [44].

Furthermore, it is to be noted that blockchain is operationally resilient and has low risk of hacking, due to the combination of cryptography and the consensus mechanisms used, such as proof-of-stake or proof-of-work, thus increasing data security [45]. However, this high level of privacy may impact in the lack of control of illicit activities, which may be significantly alleviated by the use of AI technology to catch them [46]. Hence, blockchain is a promising technology regarding efficiency and scalability [47], although some drawbacks may still be found related to data protection regulations or its integration with legacy systems [48].

4. FAULT DIAGNOSIS AND DETECTION

Regarding Industry 3.0, predictive maintenance was done through some given procedural steps, although the upcoming Industry 4.0 offers a new paradigm related to self-induce maintenance by means of employing smart predictive maintenance systems based on data analytics, where data digitalization is key in order to implement an appropriate fault diagnosis scheme [49].

Hence, fault diagnosis and detection (FDD) got a lot of attention since the popularization of big data. There are different ways to implement FDD, such as by means of a mathematical model of the system based on algebra, which leads to establishing a modeling of faults, thus resulting in those being classified as additive type or multiplicative (or parametric) type [50]. Likewise, another mathematical model based on *N*-manifolds may also be considered [51]. On the other hand, such procedures may not be really accurate when some parameters are missing, hence random forests based on decision paths are also proposed, leading to the proposition of an improved random forest based on decision paths [52].

With respect to decision trees, the use of multi-sensor data fusion with different fusion layers offers interesting solutions. This is due to the fact that the combination of different sort of sensors and different measuring strategies taken in a convenient fashion improves the accuracy and robustness of the outcome obtained by each sensor on an individual basis [53]. In this sense, sensor data fusion enhances reliability, range and accuracy of measurements in order to boost performance rates [54].

One possible strategy if three fusion layers are implemented is the following: the first one converts raw data obtained by sensors into a logical value, whereas the second one establishes a fusion tree and the values of intermediate nodes are calculate according to predefined logical operations, while the third one makes the final decision by accounting for the value of the root, which is given by means of predetermined equations [55]. Anyway, the predetermined operations may be adjusted if needed so as to obtain more accurate results.

For instance, focusing on autonomous vehicles, there are 3 sorts of sensor data fusion classification [56]. The first one is sensor fusion by abstraction level, answering the question as to when the fusion should be done, which presents three different answers, such as low (fusing the raw data), middle (fusing the detections) or high (fusing the tracks). The second one is sensor fusion by centralization level, answering the question as to where the fusion is happening, which also presents three different answers, such as centralized (one central unit does the fusion), decentralized (each sensor fuses data and sends it over to the following one) or distributed (each sensor processes the data locally and forwards it to the next unit).

The third one is sensor fusion by competition level, answering the question as to what the fusion should do, which presents three different answers as well, such as competitive (all sensors have the same purpose), complementary (different sensors offer different scenes) or coordinated (different sensors produce a brand new scene). Out of these three families of sensor fusion classification, there are up to nine different sorts of sensor fusion algorithms, where the fusion is often performed by Bayesian algorithms such as Kalman filters. Another instance of sensor data fusion is related to intelligent transportation systems (ITS) where real-time and multi-sensor traffic flow analysis is possible through data fusion methods, hence achieving acceptable prediction levels by means of big data analytics and machine learning techniques [57].

It is to be said that sensor data fusion combines the benefits of diverse types of sensors and different measuring principles in the most convenient fashion [58], hence improving the outcome obtained by individual sensors. Data fusion of multiple sensors enhance the reliability of measurements, along with its range and accuracy, thus attaining a better performance of assistance and safety functions [59]. It is to be noted that the process of fusing data coming from multiple sensors is mandatory in fields where a high level of precision is a must, which is the case of advanced robotics [60], because it improves reliability, redundancy and safety [61].

Regarding predictive maintenance (PdM) in Industry 4.0, it is to be noted that it contributes to enhance machine downtime, costs, control and quality of production [62]. There are different approaches in the literature, where the more popular ones are those addressing data analytics and machine learning tools [63]. Anyway, predictive maintenance consists of data mining to formulate machine learning models in order to gain knowledge with the aim of predicting the state of health of machinery [64]. Some popular methods of predictive maintenance are condition-based maintenance (CBM), prognosis and health management (PHM) and remaining useful life (RUL) [65].

5. SURVEILLANCE AND MONITORING OF DATA USING EDGE/FOG COMPUTING

Cloud computing has a lot of advantages when it comes to dealing with devices with constraint capabilities, such as IoT devices furnished with sensors and actuators, as it allows to defer their computing resources to remote facilities whilst keeping its own processing and power necessities to a minimum. However, network connectivity related to the cloud uses WAN technologies due to the long distances between the location of the users and that of their computing resources, which brings about a challenge when it comes to latency and bandwidth [66].

Hence, in order to keep the remote computing frameset for limited-resource devices whilst trying to address the issues related to latency and bandwidth, an interesting solution is to bring the remote computing resources closer to the end users, which is achieved by using the edge/fog computing paradigm [67]. It is to be said that the difference between both is a bit blurred, as the former considers that such resources are located within the network and the latter contemplates those resources around the network, resulting that both may claim the case where resources are situated on the edge of the network [68].

Anyway, a paradigm change between cloud computing towards edge/fog computing implies to the use of LAN technologies when it comes to network connectivity due to the shorter distances between the end users and the computing resources. This fact leads to lower values of latency and jitter, along with higher values of bandwidth, which alleviates the aforementioned issues related to cloud computing [69]. In spite of this, computing capabilities are obviously lower in edge/fog deployments that in the cloud due to the local scope of the former compared to the global scope of the latter. Hence, this feature makes cloud computing as the preferred backup solution for offloading both processing and storage capabilities from edge/fog environments [70].

Therefore, it seems clear that a convenient solution to implement Industry 4.0 capabilities in an IIoT environment is the deployment of edge/fog computing domains where cloud servers play the role of backup facilities. In this context, it is to be taken into account the restricted number of end users due to the local scope associated to edge/fog environments, which implies a limited number of server nodes to deal with all those end users. This way, the network topologies linking together those nodes do not need to be overly complicated, but just simple topologies providing redundant paths and fast connections among nodes [71].

Some of the most commonly used network architectures for data centers are going to be studied herein in order to compare its performance between any given pair of nodes, taking a number of around 16 to 32 nodes as a tradeoff between larger and smaller data centers. It is to be noted that performance in data centers is typically measured in time units. However, if all physical links within a data center have the same features, including speed and length, then the mathematical reasoning exposed in (1) and (2) applies relating time units and distance units, where speed may be viewed as a constant value a.

$$speed = \frac{distance}{time} \rightarrow time = \frac{distance}{speed} \quad (1)$$

$$distance = speed \times time = a \times time \quad (2)$$

This fact eventually expresses distance in terms of time multiplied by a constant of proportionality, which in this case is the speed of the wire. This leads to the conclusion that distance and time are directly proportional variables, which opens up the way to express performance in terms of distance. Moreover, if all links have the same length, then such a distance may be substituted with the number of links as such distance must be a multiple of the length of a single link, which may be seen as a constant value b, as shown in (3).

$$Number_{Links} = \frac{distance}{length_{link}} = \frac{distance}{b} \quad (3)$$

Therefore, (2) allows to change performance measurements from time units to distance units, whilst (3) permits to substitute distance units with the number of links between a source node and a destination node. Furthermore, if performance among different nodes is to be compared, then the constants of proportionality defined above will cancel each other out, then resulting that performance outcome expressed in either time units, distance units or number of links are equivalent in this particular context [72].

Additionally, data center network architectures may be classified into two broad categories, labeled as switch-centric and server-centric, where the former is characterized by switches being the main responsible for traffic forwarding, whilst the latter is done by servers [73]. This way, switches in the former are key players, thus they need to meet high standard requirements. On the other hand, switches on the latter play a secondary role, thus commodity switches may be employed as servers are play the main role regarding traffic forwarding [74].

Some typical examples commonly used in switch-centric data center topologies are Fat Tree and Leaf & Spine, whereas some instances typically employed in server-centric data center architectures are BCube, DCell and FiConn [75]. Therefore, those five topologies are going to be studied in order to calculate de average number of links between any given pair of nodes within each topology [76], along with some typical centralization and dispersion statistics related to them, in a way to look into the performance offered by each of those instances [77] .

5.1. Fat Tree

Fat Tree is a switch-centric data center topology built upon 3 hierarchical layers of switches, where nodes are connected only to one switch within the lower layer [78]. This topology is heavily determined by parameter k, which basically sets up the number of switches and hosts within each layer, as well as the number of links per switch. Moreover, some switches in the lower and middle layer are included into a Pod, where full mesh connectivity is established among devices in both layers, whilst there is also full mesh connectivity among switches belonging to a single Pod and all switches located in the upper layer. However, there are partial mesh connectivity between each switch in the middle layer and all those in the upper layer.

Figure 1 exhibits a topology where parameter $k = 4$, where all switches have 4 links and each of the 4 pods are composed by 4 switches. It is to be noted that switches in the lower and middle layers, those being $k^2/2$, have half of their links looking downwards and the other half look upwards, whilst switches in the upper layer, those being $k^2/4$, have all their links looking downwards. Besides, the oversubscription rate is $1{:}1$, meaning that all possible links in the design are available, thus no one is missing.

Furthermore, it is to be said that the number of hosts included in the Fat Tree topology with $k = 4$ is $k^3/4 = 4^3/4 = 16$ hosts. Alternatively, this may be calculated by assigning $k/2 = 4/2 = 2$ hosts to each end switch, those being the ones located in the lower layer, which accounts $k^2/2 = 4^2/2 = 8$ lower level switches. Anyway, the devices on each layer are identified sequentially from left to right, thus starting at 0 and growing according to the ordering of the natural numbers.

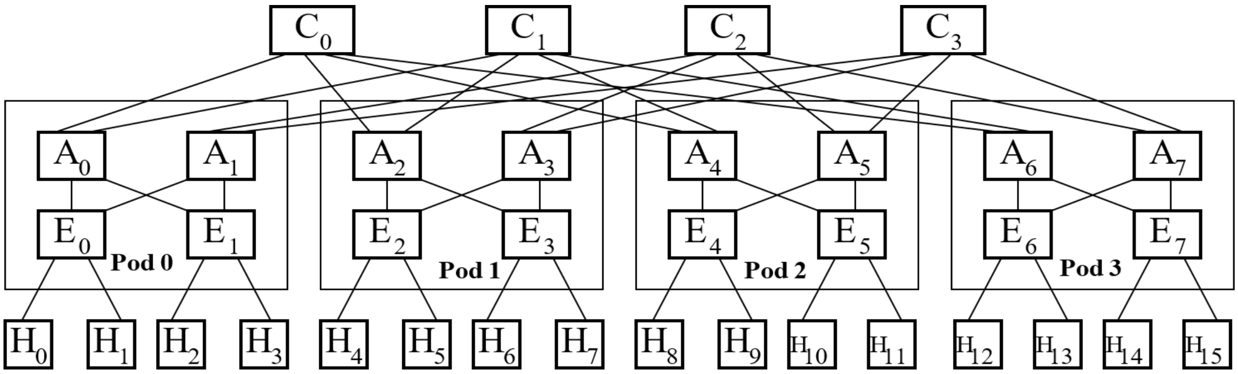

Figure 1. Fat Tree with $k = 4$ and an oversubscription rate $1{:}1$

5.2. Leaf And Spine

Leaf & Spine is also a switch-centric data center architecture organized upon 2 hierarchical layers of switches, where nodes are connected just to one switch in the lower layer [79]. Unlike fat tree, there is no parameter fixing the number of devices per layer, hence it offers a degree of freedom when it comes to design. The main feature of this topology is the establishment of a full mesh connectivity among switches in both layers, which offers an advantage in connectivity over fat tree, although this also portrays a drawback when it comes to scalability.

Figure 2 exposes a topology with 8 leaves and 2 spines, where switches in the upper layer have 8 links looking downwards towards each lower layer switch, and switches in the lower layer have 4 links, such that have half of those look downwards towards their connected hosts and the other half look upwards towards each upper layer switch. Moreover, the oversubscription rate is $1{:}1$, which means that all possible links in the design are available, thus there is no one link missing. Besides, the number of hosts included in the topology is $8 \times 2 = 16$, as each of the 8 leaves have 2 links going downwards. Anyway, the device identification on each layer are according to the sorting of natural numbers, starting at 0 and going rightwards.

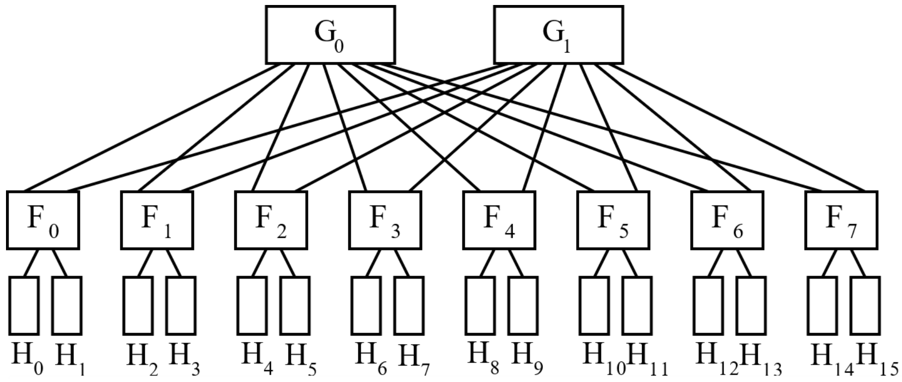

Figure 2. Leaf & Spine with 8 leaves and 4 spines, along with an oversubscription rate 1: 1

5.3. BCube

BCube is a server-centric data center topology, considered as a recursively defined structure, each one of those being labeled as a level [80]. There is a parameter n establishing the number of nodes connected to each level-0 switch. The first step to build up a BCube is to construct BCube*0* which includes only n nodes, furnished with just one port and interconnected by a switch with n ports. Then, BCube*1* is constructed out of n BCube*0* formed by nodes containing two ports and n switches with n ports. From that point on, higher levels of BCubes, such as BCube$_k$, where $k \geq 1$, are formed by n BCube$_{k-1}$ and n^k switches with n ports, where each nodes presents $k + 1$ ports.

Figure 3 depicts a network topology of a BCube_1 with $n = 4$, where there are 4 BCube*0*, composed of one switch each. Besides, each node connected to one switch within a BCube*0* is also connected to a switch located within BCube*1*. Regarding the nomenclature of the switches, they are labeled by two digits, where the first one states its level and the second one does its location within that level, going from left to right, starting at 0 and going rightwards. With respect to the nodes, they are also branded by two digits, where the first one is referred to the BCube*0* it belongs to and the second one identifies the location within that BCube*0*, beginning at 0 and going rightwards. Moreover, the number of hosts included in the topology is 16, as each of the 4 BCube*0* just has 4 nodes. Besides, each node has 2 links, where one is going to its level-*0* switch and another one to its corresponding level-*1* switch.

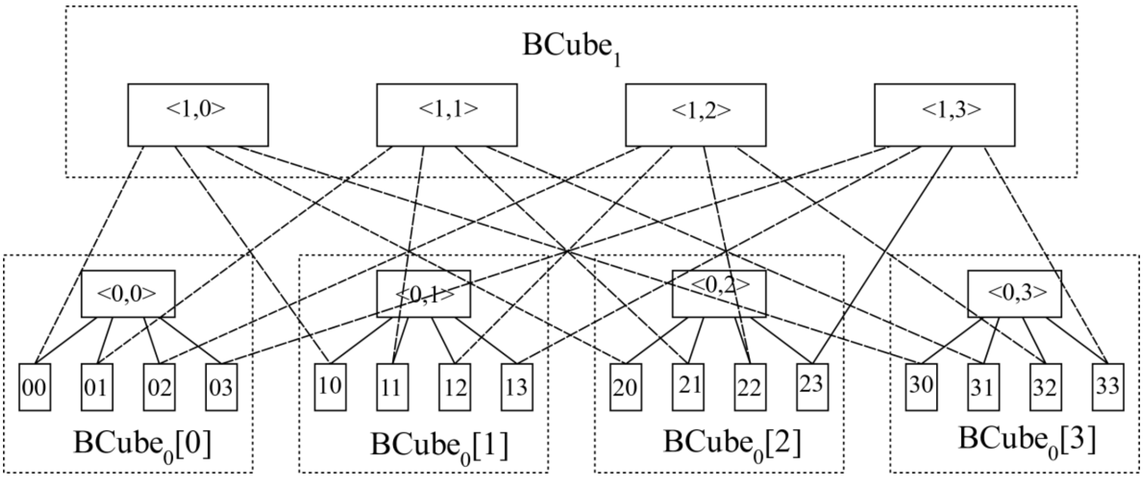

Figure 3. BCube_1 with $n = 4$ and an oversubscription rate 1: 1

5.4. DCell

DCell is a server-centric data center topology, considered as a recursively defined structure as well [81]. There is also a parameter n fixing the amount of nodes connected to each level-0 switch. The first step to construct a DCell is to build up a DCell*0*, which is composed of n nodes with exactly one port and connected through a single switch of n ports. In addition to it, DCell*1* is composed of $n + 1$ DCell*0*, where each of those is connected to all of the others by just one link.

In other words, if each node is assigned a 2-tuple $[a\_1, a\_0]$, where the subindexes stand for the level identifier where each node is located in, then an $[i, j - 1]$ node and another $[j, i]$ node are connected by a link for each i and each $j > i$. This way, if every DCell*0* within a DCell*1* is considered as a virtual node, there is a full mesh connectivity among all those DCell*0*. For higher levels of

DCell, such as DCell$_k$, where $k \geq 0$, it is to be said that the number of DCell$_{k-1}$ within a DCell$_k$ is given by the expression $g_k = t_{k-1} + 1$, whereas the number of nodes in a DCell$_k$ is stated by $t_k = g_k \times t_{k-1}$, considering that DCell$_0$ represents the initial values of $g_0 = 1$ and $t_0 = n$.

Figure 4 shows a network topology of a DCell$_1$ with $n = 4$, where there are 5 DCell$_0$, composed of one switch each. Regarding the nomenclature of the nodes, they are branded with 2 identifiers, where the first one is related to the DCell$_0$ is belongs to and the last one is assigned sequentially within that DCell$_0$ in a clockwise fashion. In addition to it, the number of hosts included in the topology is 20, as each of the 5 DCell$_0$ within the DCell$_1$ just has 4 nodes. Moreover, every node has 2 links, where one leads to its own DCell$_0$ switch and another one heads for another DCell$_0$ according to the rules exposed above.

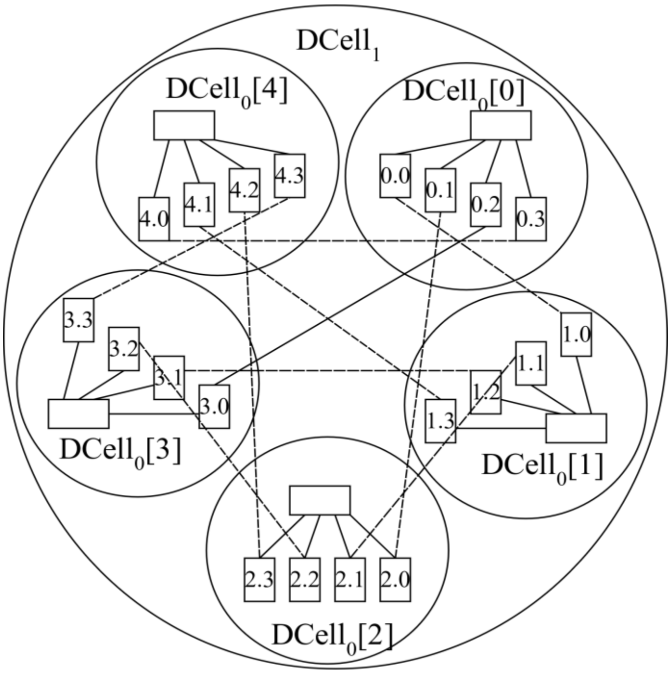

Figure 4. DCell$_1$ with $n = 4$ and an oversubscription rate $1:1$

5.5. FiConn

FiConn is also a server-centric data center topology, considered as a recursively defined structure as well [82]. As in the previous cases, there is also a parameter n stating the amount of nodes connected to each level-0 switch. However, a key point in FiConn is that nodes always have 2 ports, regardless the level of FiConn. The first step to get a FiConn is to create a FiConn$_0$, which is made of n nodes and a switch with n ports interconnecting them. In this case, the port connecting each node with its FiConn$_0$ switch is called level-0 port, whereas the backup port in each server is not being used and that is why it is called available backup port.

Focusing on the construction of a topology FiConn$_k$ for $k > 0$ upon the proper number of FiConn$_{k-1}$, it is to be considered that if there are b servers having available backup ports in a FiConn$_{k-1}$, then the amount of FiConn$_{k-1}$ within a FiConn$_k$ is $b/2 + 1$, which is called g_k. Hence, in each FiConn$_{k-1}$, $b/2$ servers from those b servers are chosen to connect the other $b/2$ FiConn$_{k-1}$ by means of their backup ports, each for a single FiConn$_{k-1}$.

Therefore, in order to build up a FiConn$_k$ upon g_k instances of FiConn$_{k-1}$, a number of servers in each FiConn$_{k-1}$ must be selected as level-k servers, which are the ones meeting the following expression: $(u_{k-1} - 2^{k-1} + 1)\ mod\ 2^k = 0$. In this sense, the value of μ_k identifies a server in FiConn$_k$, whose value ranges from 0 to $N_k - 1$, where N_k is the overall amount of servers in FiConn$_k$. Anyway, the construction of FiConn$_k$ upon g_k instances of FiConn$_{k-1}$ is easily obtained by means of the appropriate algorithm.

Sticking to the construction of FiConn$_1$, the first thing need is to establish the number of FiConn$_0$ needed, which depend on parameter n, as each FiConn$_0$ contains 4 servers and a 4-port switch. Such a number is calculated as $4/2 + 1 = 3$, because $b = 4$, meaning that each FiConn$_1$ contains 3 FiConn$_0$. Hence, the following connections are established among the level-1 servers: [0,0] and [1,0], [0,2] and [2,0], [1,2] and [2,2].

Moving to the construction of FiConn$_2$ out of four FiConn$_1$, each of those has 6 servers with backup ports free, thus it accounts for $6/2 + 1 = 4$, meaning that each FiConn$_2$ contains 4 FiConn$_1$. Hence, the following connections are established among the level-2 servers: [0,0,1] and [1,0,1], [0,1,1] and [2,0,1], [0,2,1] and [3,0,1], [1,1,1] and [2,1,1], [1,2,1] and [3,1,1], [2,2,1] and [3,2,1].

Figure 5 shows a network topology of a FiConn$_2$ with $n = 4$ (and obviously $k = 2$). One FiConn$_2$ is composed of 4 FiConn$_1$, whilst each of those is formed by 3

FiConn*0*. Eventually, as each FiConn*0* contains $n = 4$ hosts, then the overall amount of hosts in FiConn*2* is $N_2 = 4 \times 3 \times 4 = 48$, whereas $N_1 = 3 \times 4 = 12$ hosts, whilst $N_0 = 4$ hosts.

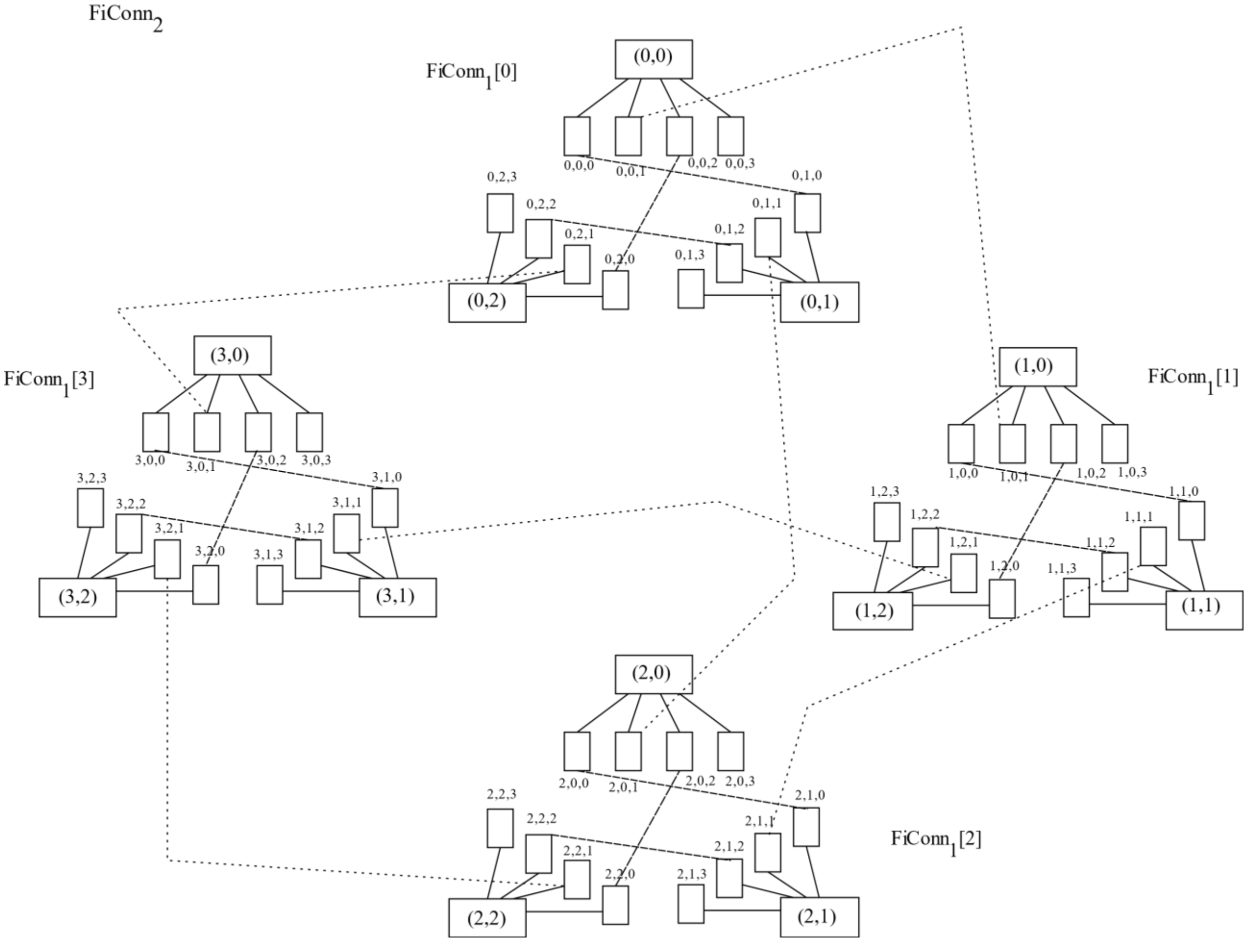

Figure 5. FiConn2 with $n = 4$ and an oversubscription rate $1{:}\,1$

6. PERFORMANCE EXPRESSED IN AVERAGE NUMBER OF LINKS

After having presented the some of the most commonly used topologies in data centers, it is going to be calculated the average number of links between any pair of nodes for each of those topologies by taking one given node as a reference to get the distance to any other node within the data center design [83]. In order to clarify the calculation, the amount of nodes at a particular number of hops away will be multiplied by this number of hops, and all those values will be summed up. Afterwards, the outcome obtained will be divided by the count of all nodes within the topology, except obviously the node taken as a reference, whose distance to itself is zero.

After that, other measurements of central tendency are also presented, such as mode and median, along with some measurements of dispersion measurements, such as variance, standard deviation and coefficient of variation [84]. All those measurements attained for each topology will be presented in a table in order to compare the results and discuss about them.

6.1. Fat Tree (Ft) With *K*=4

It is to be reminded that the Fat Tree topology considered has a value of $k = 4$ and oversubscription rate $1{:}\,1$. This results in 16 hosts, where each end switch has 2 hosts connected and each pod has 4 hosts hanging on. With regards to the bandwidth of the links, those connecting hosts and lower layer switches may be 10 Gb, whilst those between lower layer and middle layer switches may be 25 Gb, as there are no 20 Gb links, whereas those between middle layer and upper layer switches may also be 40 Gb.

With respect to the central tendency measurements, (4) shows the average number of links among nodes in Fat Tree with $k = 4$:

$$\mu_{FT} = \frac{1 \cdot 2 + 2 \cdot 4 + 12 \cdot 6}{16 - 1} = \frac{82}{15} = 5.47 \qquad (4)$$

By looking at the operation in the numerator, it is clear that mode and median are obviously 6, as it happens to be the most repeated value by far and it is the value located right in the middle of all the values sequentially ordered, respectively.

Regarding the dispersion measurements, (5) exposes the variance:

$$\sigma_{FT}^2 == \frac{1 \cdot (2-5.47)^2 + 2 \cdot (4-5.47)^2 + \\ +12 \cdot (6-5.47)^2}{16-1} = \frac{19.73}{15} = 1.32 \quad (5)$$

The standard deviation is exhibited in (6):

$$\sigma_{FT} = (1.32)^{0.5} = 1.15 \quad (6)$$

The coefficient of variation is exposed in (7):

$$CV_{FT} = \frac{1.15}{5.47} = 0.21 \quad (7)$$

6.2. Leaf and Spine (LS) With 8 Leaves and 2 Spines

It is to be noted that the Leaf & Spine topology considered is composed of 8 leaf switches and 2 spine switches, with an oversubscription rate $1{:}\,1$. This results in 16 hosts, where each end switch has 2 host connected. With regards to the bandwidth of the links, these interconnecting hosts and leaf switches may be 10 Gb, whereas these between leaf and spine switches may be 25 Gb, because there are no 20 Gb links.

Regarding the central tendency measurements, (8) exposes the average number of links among nodes in Leaf & Spine with 8 leaves and 2 spines:

$$\mu_{LS} = \frac{1 \cdot 2 + 14 \cdot 4}{16-1} = \frac{58}{15} = 3.87 \quad (8)$$

By looking at the operation in the numerator, it is clear that mode and median are just 4, because it occurs to be the most repeated value and it is the value located just in the middle of all those values sequentially ordered, respectively.

Regarding the dispersion measurements, (9) exposes the variance:

$$\sigma_{LS}^2 = \frac{1 \cdot (2-3.87)^2 + 14 \cdot (4-3.87)^2}{16-1} = \frac{3.73}{15} = 0.25 \quad (9)$$

The standard deviation is shown in (10):

$$\sigma_{LS} = (0.25)^{0.5} = 0.50 \quad (10)$$

The coefficient of variation if displayed in (11):

$$CV_{LS} = \frac{0.50}{3.87} = 0.13 \quad (11)$$

6.3. BCube (BC) With *N*=4

It is to be remarked that the BCube topology considered is a BCube*1* with a value of $n = 4$ and oversubscription rate $1{:}\,1$. This is formed by 4 BCube*0*, where each of these is composed of 1 switch with 4 hosts connected to it, thus resulting in an overall amount 16 hosts within a BCube*1*.

Furthermore, each host has 2 ports, where one is connected to its own level-0 switch, meaning the switch within its own BCube*0*, and the other one to a level-1 switch, meaning the corresponding switch in BCube*1*. With regards to the bandwidth of the links, all links may be 10 Gb as switches do not aggregate any traffic because traffic forwarding is carried out by nodes, leaving switches with a testimonial role.

With respect to the central tendency measurements, (12) shows the average number of links among nodes in BCube*1* with $n = 4$:

$$\mu_{BC} = \frac{6 \cdot 2 + 9 \cdot 4}{16-1} = \frac{48}{15} = 3.20 \quad (12)$$

By looking at the operation in the numerator, it is clear that mode and median are both 4.

Regarding the dispersion measurements, (13) exhibits the variance:

$$\sigma_{BC}^2 = \frac{6 \cdot (2-3.20)^2 + 9 \cdot (4-3.20)^2}{16-1} = \frac{14.40}{15} = 0.96 \quad (13)$$

The standard deviation is seen in (14):

$$\sigma_{LS} = (0.96)^{0.5} = 0.98 \quad (14)$$

The coefficient of variation if viewed in (15):

$$CV_{LS} = \frac{0.98}{3.20} = 0.31 \quad (15)$$

6.4. DCell (DC) With *n*=4

It is to be mentioned that the DCell*1* topology considered is a DCell*1* with a value of $n = 4$ and oversubscription rate $1{:}\,1$. This is formed by 5 DCell*0*, each of them being composed of 1 switch with 4 hosts connected to it, thus accounting for 20 hosts overall within a DCell*1*.

Moreover, each host has 2 ports, where one is connected to its own level-0 switch, meaning the switch within its own DCell$_0$, and the other one to another level-0 switch belonging to the same DCell$_1$. It is to be stated that each host within a DCell$_0$ is connected to a different DCell$_0$ within the same DCell$_1$. Additionally, it is to be reminded that there are no level-1 switches in a DCell$_1$, as it is composed of only level-0 switches, and the same happens in higher instances of DCell$_k$.

Regarding the central tendency measurements, (16) exposes the average number of links among nodes in DCell$_1$ with $n = 4$:

$$\mu_{DC} = \frac{1 \cdot 1 + 3 \cdot 2 + 6 \cdot 3 + 3 \cdot 4 + 6 \cdot 5}{20 - 1} = \frac{67}{19} = 3.53 \tag{16}$$

By looking at the operation in the numerator, the sample of values is bimodal, as there are two peaks located at 3 and 5, respectively. On the other hand, the median value is 3.

Regarding the dispersion measurements, (17) exposes the variance:

$$\sigma_{DC}^2 = \frac{\begin{matrix} 1 \cdot (1 - 3.53)^2 + 3 \cdot (2 - 3.53)^2 + \\ +6 \cdot (3 - 3.53)^2 + 4 \cdot (4 - 3.53)^2 + \\ +6 \cdot (5 - 3.53)^2 \end{matrix}}{20 - 1} = \frac{28.96}{19} = 1.52 \tag{17}$$

The standard deviation is calculated in (18):

$$\sigma_{DC} = (1.52)^{0.5} = 1.23 \tag{18}$$

The coefficient of variation if shown in (19):

$$CV_{LS} = \frac{1.23}{3.53} = 0.35 \tag{19}$$

6.5. FiConn (FC) With n=4

It is to be noted that the FiConn topology considered is a FiConn$_2$ with a value of $n = 4$ and oversubscription rate $1{:}\,1$. This is formed by 4 FiConn$_1$, where each of them contain 3 FiConn$_1$, which in turn comprise 1 switch and 4 hosts connected to it, hence yielding 48 hosts overall within a FiConn$_2$.

It is to be reminded that all hosts within a FiConn architecture are furnished with 2 ports, where one is connected to its FiConn$_0$ switch and the other one plays a backup role, which may be used to connect with a host located in another FiConn$_k$ ($k > 0$) according to some preestablished rules described earlier on. Hence, some hosts might have two ports available, whereas some other might have just one. Additionally, all switches are level-0, as there are no level-1 switches in a FiConn$_1$ and the same happens to FiConn$_2$ or any other higher FiConn$_k$.

When it comes to measure the distance among hosts in number of links away, it may depend on the source host selected, because sticking to the case of FiConn$_2$ with $n = 4$, in each FiConn$_0$ there is one host with its backup port unused, two hosts have the backup ports connected to a host within another FiConn$_0$ inside its FiConn$_1$, and another host has its backup port tied to a host within a FiConn$_0$ inside another FiConn$_1$.

The most straightforward way to deal with this scenario is by taking the host $(0,0,0)$ as the source host, and from there on, start taking all measurements. If this is the case, then it results that FiConn$_1$*[0]*, where $(0,0,0)$ is located, follows this pattern: 1—3—4—1—0—2, where the first value is one hop away, the second one is two hops away, and so on. On the other hand, the rest of FiConn$_1$ follow the general pattern: 1—0—3—2—0—6, which is the one applying to the rest of FiConn$_1$. However, the entry host in each FiConn$_1$ is reached at different hops away, such that FiConn$_1$*[1]* is reached in three hops, FiConn$_1$*[2]* is obtained in four hops, and FiConn$_1$*[3]* is attained in seven hops. Hence, if such an entry host is considered as the first 1 in the general pattern, the number of hops away to reach the rest of hosts within a given FiConn$_1$ grows sequentially according to the general pattern.

It is to be noted that after reaching a host in a new FiConn$_1$, then next hop is to get to the FiConn$_0$ switch attached to that host, and in the next hop all hosts within that FiConn$_0$ are reached. Afterwards, in the next hop, a host of each of the rest of FiConn$_0$ within that new FiConn$_1$ is reached, followed by getting to switches linked to those hosts in the next hop, and eventually, arriving to the rest of the hosts in each particular FiConn$_0$ in the next hop.

Hence, it is possible to copy this scheme in the initial FiConn$_1$ as well by choosing a host with a backup port unused as the source host, and this way, all FiConn$_1$ will follow the same pattern, including the initial one. For instance, this is achieved if host $(0,0,3)$ may be taken as the source host, because this particular host has its backup port unused.

This way, we find the general pattern in FiConn$_1$*[0]* and FiConn$_1$*[1]*: 1—0—3—2—0—6, whilst we find a modified pattern in FiConn$_1$*[2]* and FiConn$_1$*[3]*: 1—0—3—3—0—5. The reason for this is because the entry host of FiConn$_1$*[1]* is reached in three hops, whereas his takes place in six hops in both FiConn$_1$*[2]* and FiConn$_1$*[3]*, so the direct links between FiConn$_1$*[1]* with FiConn$_1$*[2]* and FiConn$_1$*[3]* facilitate to reach the hosts directly linked in those FiConn$_1$ two hops before it is expected to by the general pattern.

Putting all together, Table 1 summarizes how many hosts are at each distance in a FiConn$_2$ network topology, starting at a host with its backup port unused, such as $(0,0,3)$, which obviously would account for 0 hops away from itself and it is not included in the table.

Table 1. Number of hops away in $FiConn_2$

$FiConn_1$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
$FiConn_1$[0]	0	3	2	0	6						
$FiConn_1$[1]			1	0	3	2	0	6			
$FiConn_1$[2]						1	0	3	3	0	5
$FiConn_1$[3]						1	0	3	3	0	5
COUNT	0	3	3	0	9	4	0	12	6	0	10

Therefore, with regards to the central tendency measurements, (20) displays the average number of links among nodes in $FiConn_2$ with $n = 4$:

$$\mu_{FC} = \frac{3 \cdot 2 + 3 \cdot 3 + 9 \cdot 5 + 4 \cdot 6 + 12 \cdot 8 + 6 \cdot 9 + 10 \cdot 11}{48 - 1} = \frac{344}{47} = 7.32 \quad (20)$$

By looking at the operation in the numerator, it is clear that mode and median are both 8 as it is the most repeated value and it is the one located right in the middle when all those values are organized in a sequential fashion.

Regarding the dispersion measurements, (21) depicts the variance:

$$\sigma_{FC}^2 = \frac{3 \cdot (2 - 7.32)^2 + 3 \cdot (3 - 7.32)^2 + 9 \cdot (5 - 7.32)^2 + 4 \cdot (6 - 7.32)^2 + 12 \cdot (8 - 7.32)^2 + 6 \cdot (10 - 7.32)^2 + 10 \cdot (11 - 7.32)^2}{48 - 1} = \frac{380{,}37}{47} = 8.09 \quad (21)$$

The standard deviation is calculated in (22):

$$\sigma_{FC} = (8.09)^{0.5} = 2.84 \quad (22)$$

The coefficient of variation if shown in (23):

$$CV_{FC} = \frac{2.84}{7.32} = 0.39 \quad (23)$$

6.6. DISCUSSION

First of all, Table 2 summarizes all statistical data obtained for the aforementioned data center network topologies.

Table 2. Statistics related to hop count for commonly used data center network topologies

Topology	μ	mode	median	σ^2	σ	CV
Fat Tree ($k = 4$)	5.47	6	6	1.32	1.15	0. 21
Leaf & Spine (8 leaves, 2 spines)	3.87	4	4	0.25	0.50	0.13
$BCube_1$ ($n = 4$)	3.20	4	4	0.96	0.98	0.31
$DCell_1$ ($n = 4$)	3.53	3-5	3	1.52	1.23	0.35
$FiConn_2$ ($n = 4$)	7.32	8	8	8.09	2.84	0.39

The first column within this table exposes that switch-centric data center network topologies offer higher values of average number of hops among any given pair of nodes than their server-centric counterparts, with the exception of FiConn, where a network topology with more hosts have been chosen. This fact allows to say that performance in server-centric schemes is higher than in their switch-centric counterparts, according to the conditions exposed at the beginning of this section.

Focusing on switch-centric architectures, Leaf & Spine designs offer better outcome than Fat Tree, which may be explained because the latter has an extra layer of hierarchical switches, thus making it ready to a higher degree of escalation, as it may be viewed in both mode and median values. It permits Fat Tree doing it better for higher numbers of nodes, whereas Leaf & Spine does it better for lower numbers, whilst getting worse as the amount of nodes grow. Anyway, both designs presents quite uniform values, represented by lower values in the coefficient of variation, which may indicate a good point when working with streaming data.

Centering on server-centric topologies, $BCube_1$ designs present a slightly better outcome than $DCell_1$, although the latter might achieve better values where the amount of nodes rises, according to the values of mode and median, even though differences may not be really significant among them both. Anyway, as stated above, both designs achieve better values for average number of hops among nodes, which may indicate a good point for any type of data flows. With regards to $FiConn_2$, the design proposed has a much higher number of nodes, and that is why the results obtained are not comparable with the rest of architectures presented.

Additionally, the values obtained for the dispersion measurements, as well as the coefficient of variation, are within reasonable values, meaning that the number of hops between pairs of hops does not present high volatility, but on the contrary, it represents that the difference among values is not high, which accounts for relatively stable values in the number of hops, thus offering low variations of latency and jitter, whilst keeping good levels of bandwidth.

In summary, when it comes to centralization measurements in statistics, such the average number of hops, it results that server-centric layouts, such as BCube$_1$ $(n = 4)$ and DCell$_1$ $(n = 4)$, present better results than their switch-centric counterparts, such as Fat Tree $(k = 4)$ and Leaf & Spine (8 leaves and 2 spines). On the other hand, when it comes to dispersion measurements in statistics, such as the coefficient of variation derived from dividing the standard deviation by the average, it results the other way around, as switch-centric layouts present better outcome than their server-centric counterparts. In addition to it, FiConn$_2$ $(n = 4)$, it conveys a higher number of hosts than the other topologies proposed, and that is why results are not comparable with them. Also, the coefficient of variation does not reach 0.4 in any case, which means it is within an acceptable rate.

Eventually, the outcome achieved display lower average rates for the switch-centric topologies chosen with the same number of end hosts, namely Fat Tree and Leaf & Spine, whereas server-centric topologies with a similar amount of end hosts offer lower rates for the coefficient of variation. Therefore, the results achieved may lead to select the former for real-time applications, which prefer lower average rates, or otherwise, to choose the latter for streaming traffic as they prefer lower CV values.

7. CONCLUSION

In this paper, fault diagnosis and prediction has been reviewed from the point of view of surveillance and monitoring of data using Edge/Fog Computing. To start with, a small introduction about the different industrial revolutions happening from the 18th century up to now have been presented, along with a brief outline of what the next stages are supposed to be.

Afterwards, a review on operational technology has been presented, as this is supposed to be one of the key points in the current paradigm of Industry 4.0. After that, the role of blockchain has also been exposed within such a paradigm, with the goal of keeping the integrity of all transactions being undertaken, as well as offering an efficient traceability. Then, it is described the concept of fault diagnosis and detection as one of the main innovations of Industry 4.0 related to smart maintenance with the use of sensor data fusion from multiple sources, which is mandatory in robotics so as to enhance reliability, redundancy and safety.

At that point, the focus has been set on the logistics to carry out all of the above, which are edge/fog computing environments. In this sense, some of the most commonly used network topologies in data centers have been presented, including two instances of switch-centric designs, represented by Fat Tree and Leaf & Spine, followed by other three instances of server-centric designs, portrayed by BCube and DCell, as well as FiConn.

A test scenario for each of those designs have been set up with edge/fog computing needs in mind and some statistics have been calculated in order to evaluate performance. The results of such statistical measurements have shown that server-centric topologies perform better than switch-centric topologies due to shorter average number of hops between any given pair of nodes in the former, which makes them more convenient for real time traffic. On the other hand, the latter have presented more uniform values due to their lower amounts for coefficient of variation, which make them more interesting for streaming traffic.

ACKNOWLEDGEMENT

I would like to thank Orhan Aslan, who helped me with his valuable assistance and contributions in the creation of this study.

DECLARATION OF ETHICAL STANDARDS

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

AUTHORS' CONTRIBUTIONS

Pedro Juan ROİG: Wrote the manuscript, performed the calculations and analyzed the results.
Salvador ALCARAZ: Analyzed the results.
Katja GILLY: Analyzed the results.
Cristina BERNAD: Analyzed the results.
Carlos JUIZ: Analyzed the results.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] Sharma A., Singh B.J. Evolution of Industrial Revolutions: A Review. ***International Journal of Innovative Technology and Exploring Engineering***, 9(11): 66–73, (2020).

[2] Bula P., Niedzielski B. Industrial revolution – from Industry 1.0 to Industry 4.0. ***Management, Organisations and Artificial Intelligence***, chapter 1, Routledge, London, UK, (2021).

[3] Kjelsrud, J. A Secure Transition from Industry 3.0 to Industry 4.0 for Manufactures. Recommendations from a security perspective. ***Master's Degree Thesis in Informatics***, University of Oslo, Norway, (2022).

[4] Xu X., Lu Y., Vogel-Heuser B, Wang L. Industry 4.0 and Industry 5.0 - Inception, conception and perception. ***Journal of Manufacturing Systems***, 61: 530–535, (2021).

[5] Chourasia S., Tyagi A., Pandey S.M., Walia R.S., Murtaza Q. Sustainability of Industry 6.0 in Global Perspective: Benefits and Challenges. ***Journal of Metrology Society of India***, 37(2): 443–452, (2022).

[6] Yang F, Gu S. Industry 4.0, a revolution that requires technology and national strategies. ***Complex & Intelligent Systems***, 7: 1311–1325, (2021).

[7] Javaid M., Haleem A., Singh R.P., Suman R, Santibáñez-González E. Understanding the adoption of Industry 4.0 technologies in improving environmental sustainability.

Sustainable Operations and Computers, 3: 203–217, (2022).

[8] Alberro L., Castro A., Grampin E. Experimentation Environments for Data Center Routing Protocols: A Comprehensive Review. ***Future Internet***, 14(1): 0029, (2022).

[9] Couto R.S. et al. Reliability and Survivability Analysis of Data Center Network Topologies. ***Journal of Network and Systems Management***, 24: 346–392, (2016).

[10] Cortés-Castillo A. Various Network Topologies and an Analysis Comparative Between Fat-Tree and BCube for a Data Center Network: An Overview. ***Proceedings of the IEEE Cloud Summit***, Fairfax, VA, USA, (2022).

[11] Alqahtani J., Hamdaoui B. Rethinking Fat-Tree Topology Design for Cloud Data Centers. ***Proceedings of IEEE Global Communications Conference (GLOBECOM)***, Abu Dhabi, United Arab Emirates, (2018).

[12] Han X. et al. Study of data center communication network topologies using complex network propagation model. ***Frontiers in Physics***, 11: 1174099, (2023).

[13] Negara E.S., Keni K., Andryni R. BCube and DCell Topology Data Center Infrastructures Performance. ***IOP Conference Series: Materials Science and Engineering***, 852: 012129, (2019).

[14] Lin W., Li X.Y., Chang J.M., Jia X. Constructing Multiple CISTs on BCube-Based Data Center Networks in the Occurrence of Switch Failures. ***IEEE Transactions on Computers***, 72: 1971–1984, (2023).

[15] Ahmed R.E, Helal H. New Fault-Tolerant Datacenter Network Topologies. ***Journal of Communications***, 13(6): 259–265, (2018).

[16] Shilenge M.C., Telukdarie A. Optimization of Operational and Information Technology Integration Towards Industry 4.0. ***Proceedings of IEEE 31st International Symposium on Industrial Electronics (ISIE)***, Anchorage, AK, USA, (2022).

[17] Felser M., Rentschler M., Kleineberg O. Coexistence Standardization of Operation Technology and Information Technology. ***Proceedings of the IEEE***, 107(6): 962–976, (2019).

[18] Kuppusamy E., Mariappan K. Integration of Operation Technology (OT) and Information Technology (IT) Through Intelligent Automation in Manufacturing Industries. ***Advances in Manufacturing Technology XXXIV***, chapter 42, IOS Press, Amsterdam, The Netherlands, (2021).

[19] Kebande V.R. Industrial internet of things (IIoT) forensics: The forgotten concept in the race towards industry 4.0. ***Forensic Science International: Reports***, 5: 100257, (2022).

[20] Murray G., Johnstone M.N., Valli C. The convergence of IT and OT in critical infrastructure. ***Proceedings of 15th Australian Information Security Management Conference***, 2017, Perth, Western Australia, (2017).

[21] Dos Santos D.R., Dagrada M., Costante E. Leveraging operational technology and the Internet of things to attack smart buildings. ***Journal of Computer Virology and Hacking Techniques***, 17: 1–20, (2021).

[22] Sonkor M.S., García de Soto B. Operational Technology on Construction Sites: A Review from the Cybersecurity Perspective. ***Journal of Construction Engineering and Management***, 147(12): 04021172, (2021).

[23] Klinc R., Turk Z. Construction 4.0 - Digital Transformation of One of the Oldest Industries. ***Economic and Business Review***, 21(3): 0092 (2019).

[24] Vogel F., Hamann, H. Legal Linguistics in Times of Language Models and Text Automation. ***International Journal of Language & Law***, 12: 1–7, (2023).

[25] Thoppilan R. et al. LaMDA: Language Models for Dialog Applications. ***arXiv***, 2201.08239v3: 1–47, (2022).

[26] Darici M.B., "Performance Analysis of Combination of CNN-based Models with Adaboost Algorithm to Diagnose Covid-19 Disease", ***Journal of Polytechnic***, 26(1): 179–190, (2023).

[27] Akundi A. et al. State of Industry 5.0 - Analysis and Identification of Current Research Trends. ***Applied System Innovation***, 5(1): 0027, (2022).

[28] Bonyuet D. Overview and Impact of Blockchain on Auditing. ***The International Journal of Digital Accounting Research***: 20, 31-43, (2020).

[29] Ferrer-Sapena A., Sánchez-Pérez E.A. Applications of blockchain technology in scientific documentation: current situation and perspectives. ***El profesional de la información***, 28(2): e280210, (2019).

[30] Bahga A., Madisetti V.K. Blockchain Platform for Industrial Internet of Things. ***Journal of Software Engineering and Applications***, 9(10): 910036, (2016).

[31] Khan S.N., Loukil F., Ghedira-Guegan C., Benkhelifa E., Bani-Hani A. Blockchain smart contracts: Applications, challenges, and future trends. ***Peer-to-Peer Networking and Applications***, 14: 2901–2925, (2021).

[32] Wang Q., Zhu X., Ni Y., Gu L., Zhu H. Blockchain for the IoT and industrial IoT: A review. ***Internet of Things***, 10: 100081, (2020).

[33] Kumar R.L., Khan F., Kadry F., Rho S. A Survey on blockchain for industrial Internet of Things. ***Alexandria Engineering Journal***, 61(8): 6001–6022, (2022).

[34] Lucio Y.LL., Villalba K.M., Donado S.A. Adaptive Blockchain Technology for a Cybersecurity Framework in IIoT. ***IEEE Revista Iberoamericana de Tecnologías del Aprendizaje***, 17(2): 178 –184, (2022).

[35] Siegfried N., Rosenthal T., Benlian A. Blockchain and the Industrial Internet of Things: A requirement taxonomy and systematic fit analysis. ***Journal of Enterprise Information Management***, 35(6): 1454–1476, (2022).

[36] Latif S., Idrees Z., Huma Z., Ahmad J. Blockchain technology for the industrial Internet of Things: A comprehensive survey on security challenges, architectures, applications, and future research directions. ***Transactions on Emerging Telecommunications Technologies***, 32(11): e4337, (2021).

[37] Dwivedi S.K., Roy P., Karda C., Agrawal S., Amin R. Blockchain-Based Internet of Things and Industrial IoT: A Comprehensive Survey. ***Security and Communication Networks***, 2021: 7142048, (2021).

[38] Wang G., Shi Z., Nixon M., Han S. ChainSplitter: Towards Blockchain-Based Industrial IoT Architecture for Supporting Hierarchical Storage. ***Proceedings of IEEE International Conference on Blockchain***, Atlanta, GA, USA, (2019).

[39] Xu X, Zheng Z, Yang S, Shao H. A Novel Blockchain Framework for Industrial IoT Edge Computing. ***Sensors***, 20(7): 2061, (2020).

[40] Singh A., Raghav S., Senger P.K., Kumar A. Blockchain and Edge computing for Industrial Internet of Things (IIoT) Applications. ***Asian Journal of Convergence in Technology***, 8(1): 0016, (2022).

[41] Greeshma P.P, Rajashekar J.S. Blockchain for Industrial Internet of Things. ***International Journal for Research in Applied Science & Engineering Technology (IJRASET)***, 10(IX): 0147, (2022).

[42] Kumar K.S., Kumar V.K., Ilamparithi T., Prabhu S.R.B., Kumar R.D. Emerging Trends and Research Issues on Blockchain Technology for 5G-Enabled Industrial IoT. ***Blockchain Technology***, chapter 13, 1st edition, CRC Press, Boca Raton, FL, USA, (2020).

[43] Sharma R. Blockchain for Industrial Internet of Things (IIoT). ***Blockchain and AI Technology in the Industrial Internet of Things***, chapter 3, 1st edition, IGI Global, Hershey, PA, USA, (2021).

[44] Adhikari N., Ramkumar M. IoT and Blockchain Integration: Applications, Opportunities, and Challenges. ***Network***, 3(1): 115–141, (2023).

[45] Sandner P., Gross J., Richter R. Convergence of Blockchain, IoT, and AI. ***Frontiers in Blockchain***, 3, 522600, (2020).

[46] Zhao S., Li S., Yao Y. Blockchain Enabled Industrial Internet of Things Technology. ***IEEE Transactions on Computational Social Systems***, 6(6): 1442–1453, (2019).

[47] Baalamurugan K.M. et al. Blockchain-enabled K-harmonic framework for industrial IoT-based systems. ***Scientific Reports***, 13: 1004, (2023).

[48] Alladi T., Chamola V., Parizi R.M., Choo K.K. Blockchain Applications for Industry 4.0 and Industrial IoT: A Review. ***IEEE Access***, 7: 176935–176951, (2019).

[49] Adeyeri M.K. From Industry 3.0 to Industry 4.0: Smart Predictive Maintenance System as Platform for Leveraging. ***Arctic Journal***, 71(11): 64–81, (2018).

[50] Doraiswami R., Cheded L. Fault Detection and Isolation. ***Fault Diagnosis and Detection (FDD)***, 1st edition, IntechOpen, London, UK, (2017).

[51] Bagchi S., Cheng Q.S. Computational modeling of consistent observation of asynchronous distributed computation on N–manifold. ***Cogent Engineering***, 5(1): 1528029, (2018).

[52] Zhang Y., Luo L., Ji X., Dai Y. Improved Random Forest Algorithm Based on Decision Paths for Fault Diagnosis of Chemical Process with Incomplete Data. ***Sensors***, 21(20): 6715, (2021).

[53] Durrant-Whyte H., Henderson T.C. Multisensor Data Fusion. ***Handbook of Robotics***. Springer, Berlin, Heidelberg, 585–610, (2008).

[54] Naqvi R.A., Arsalan M., Qaiser T., Khan T.M., Razzak I. Sensor Data Fusion Based on Deep Learning for Computer Vision Applications and Medical Applications. ***Sensors***, 22(20): 8058, (2022).

[55] Lee S.K., Hong S.H., Jun W.H., Hong Y.S. Multi-Sensor Data Fusion with a Reconfigurable Module and Its Application to Unmanned Storage Boxes. ***Sensors***, 22(14): 5388, (2022).

[56] **https://www.thinkautonomous.ai/blog/9-types-of-sensor-fusion-algorithms/**, “9 types of sensor fusion algorithms” (2021), accessed on July 7th, 2023.

[57] Yadav P., Mishra A., Kim S. A Comprehensive Survey on Multi-Agent Reinforcement Learning for Connected and Automated Vehicles. ***Sensors***, 23(10): 4710 (2023).

[58] Castanedo F. A Review of Data Fusion Techniques. ***The Scientific World Journal***, 2013: 704504, (2013).

[59] **https://www.bosch-mobility-solutions.com/en/solutions/sensors/sensor-data-fusion/**, “Sensor data fusion” (2023), accessed on July 7th, 2023.

[60] Aroulanandam V.V. et al. Sensor data fusion for optimal robotic navigation using regression based on an IOT system. ***Measurement: Sensors***, 24: 100598, (2022).

[61] Varghese J.P., Sundaramoorthy K., Sankaran A. Development and Validation of a Load Flow Based Scheme for Optimum Placing and Quantifying of Distributed Generation for Alleviation of Congestion in Interconnected Power Systems. ***Energies***, 16(6): 2536, (2023).

[62] Zonta T. et al. Predictive maintenance in the Industry 4.0: A systematic literature review. ***Computers & Industrial Engineering***, 150: 106889, (2020).

[63] Kashinath S.A. et al. Review of Data Fusion Methods for Real-Time and Multi-Sensor Traffic Flow Analysis. ***IEEE Access***, 9: 51258–51276, (2021).

[64] Karaahmetoglu E., Ersöz S., Türker A.K., Ates V. and Inal A.F., “Evaluation of Profession Predictions for Today and the Future with Machine Learning Methods: Emperical Evidence From Turkey”, ***Journal of Polytechnic***, 26(1): 107–124, (2023).

[65] Achouch M. et al. On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. ***Applied Sciences***, 12(16): 8081, (2022).

[66] Roig P.J., Alcaraz S., Gilly K., Bernad C., Juiz C. Modeling an edge computing arithmetic framework for IoT. ***Sensors***, 22(3): 1084 (2022).

[67] Roig P.J., Alcaraz S., Gilly K., Bernad C., Juiz C. Arithmetic Framework to Optimize Packet Forwarding among End Devices in Generic Edge. ***Sensors***, 22(2): 421 (2022).

[68] Sunyaev A. Fog and Edge Computing. ***Internet Computing***, chapter 8, Springer, Cham, Switzerland, (2020).

[69] Roig P.J., Alcaraz S., Gilly K., Bernad C., Juiz C. Modeling of a Generic Edge Computing Application Design. ***Sensors***, 21(21): 7276 (2021).

[70] Firouzi F., Farahani B., Panahi E., Barzegari M. Task Offloading for Edge-Fog-Cloud Interplay in the Healthcare Internet of Things (IoT). ***Proceedings of IEEE International Conference on Omni-Layer Intelligent Systems (COINS)***, Barcelona, Spain, (2021).

[71] Roig P.J., Alcaraz S., Gilly K., Bernad C., Juiz C. Formal Algebraic Model of an Edge Data Center with a Redundant Ring Topology. ***Network***, 3(1): 142–157 (2023).

[72] Roig P.J. Formal Algebraic Modelling of a Fog Computer Network Architecture. ***PhD Thesis in Information and Communication Technologies***, University of the Balearic Islands, Spain, (2022).

[73] Al-Makhlafi M., Gu H., Yu X., Lu Y. P-Cube: A New Two-Layer Topology for Data Center Networks

Exploiting Dual-Port Servers. ***IEICE Transactions on Communications***, E103-B(9): 940–950, (2020).

[74] Cortés-Castillo, A. Various Network Topologies and an Analysis Comparative Between Fat-Tree and BCube for a Data Center Network: An Overview. ***Proceeding of IEEE Cloud Summit, Fairfax***, VA, USA, (2022).

[75] Roig P.J., Alcaraz S., Gilly K., Bernad C., Juiz C. Arithmetic Study about Efficiency in Network Topologies for Data Centers. ***Network***, 3(3): 298–325, (2023).

[76] Roig P.J., Alcaraz S., Gilly K., Juiz C. Arithmetic Study about Energy Save in Switches for some Data Centre Topologies. ***Journal of Polytechnic***, 25(2): 785–797 (2022).

[77] Roig P.J., Alcaraz S., Gilly K., Bernad C., Juiz C. Edge Data Center Organization and Optimization by Using Cage Graphs. ***Network***, 3(1): 93–114, (2023).

[78] Al-Fares M., Loukissas A., Vahdat A. A Scalable, Commodity Data Center Network Architecture. ***Proceedings of SIGCOMM 2008***, Seattle, WA, USA, (2008).

[79] Alizadeh M., Edsall T. On the Data Path Performance of Leaf-Spine Datacenter Fabrics. ***Proceedings of IEEE 21st Annual Symposium on High-Performance Interconnects***, San Jose, CA, USA, (2013).

[80] Guo C. et al. BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers. ***Proceedings of SIGCOMM 2009***, Barcelona, Spain, (2009).

[81] Guo C. et al. DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers. ***Proceedings of SIGCOMM 2008***, Seattle, WA, USA, (2008).

[82] Li D., Guo C., Wu H., Tan K., Zhang Y., Lu S. FiConn: Using Backup Port for Server Interconnection in Data Centers. ***Proceedings of INFOCOM 2009***, Rio de Janeiro, Brazil, (2009).

[83] Roig P.J., Alcaraz S., Gilly K., Bernad C., Juiz C. Features of data center network topologies fit for IIoT deployments. ***Advances and Challenges in Science and Technology***, 9: 29–48 (2023).

[84] Bornholdt H., Röbert K., Breitbach M., Fischer M., Edinger J. Measuring the Edge: A Performance Evaluation of Edge Offloading. ***Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)***, Atlanta, GA, USA, (2023).