Scafida: A Scale-Free Network Inspired Data Center Architecture

László Gyarmati, Tuan Anh Trinh
Network Economics Group
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics, Hungary
{gyarmati,trinh}@tmit.bme.hu

ABSTRACT

Data centers have a crucial role in current Internet architecture supporting content-centric networking. State-of-the-art data centers have different architectures like fat-tree [16, 10], DCell [12], or BCube [11]. However, their architectures share a common property: symmetry. Due to their symmetric nature, a tricky point with these architectures is that they are hard to be extended in small quantities. Contrary to state-of-the-art data center architectures, we propose an asymmetric data center topology generation method called Scafida inspired by scale-free networks; these data centers have not only small diameters and high fault tolerance, inherited by scale-free networks, but can also be scaled in smaller and less homogenous increments. We extend the original scale-free network generation algorithm of Barabási and Albert [5] to meet the physical constraints of switches and routers. Despite the fact that our method artificially limits the node degrees in the network, our data center architectures keep the preferable properties of scale-free networks. Based on extensive simulations we present preliminary results that are promising regarding the error tolerance, scalability, and flexibility of the architecture.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications; C.4 [**Performance of Systems**]: Design studies

General Terms

Design, Reliability, Management

Keywords

data center, scale-free network

1. INTRODUCTION

Data centers provide diverse services—like cloud services and large-scale computations—to their customers. The properties of data centers can be described based on capabilities like computation, storage, and Internet access. In addition, the network topology of the data center has a crucial impact on the performance as huge amount of data have to be transmitted intra data center.

Data center networking has attracted the attention of the research community recently. Novel architectures are proposed continuously with different network topologies, including BCube [11, 18], DCell [12], and fat-tree [2, 16, 10]—just to mention a few of them. Although these proposals present diverse structures, they also share two common basic properties: symmetric design and homogenous equipments. Due to their symmetric structures the size of these data centers can be altered only in large quantities; accordingly, it is hard to scale the network with the flexibility of having heterogeneous deployments.

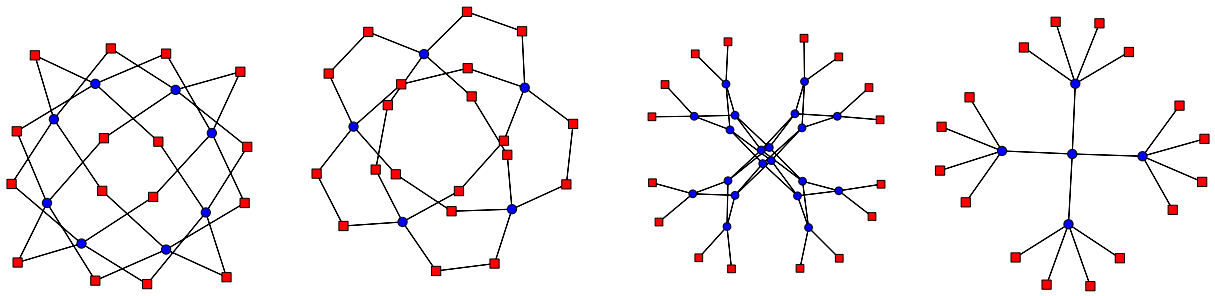
However, networks can also have asymmetric structures not only symmetric ones. The existence of biological networks, whose structures are mostly asymmetric, proves that these structures have preferable properties as they survived the evolutionary competition. Numerous biological networks share a common characteristic; i.e., the distribution of their node degrees follows power-law distribution [3]; these networks are called scale-free networks. Scale-free networks have two important aspects, namely ultra-small diameter [6] and high error tolerance [4], which would be favorable in case of data center networks too.

Therefore, in this paper, we propose a scale-free network inspired data center architecture generation algorithm called Scafida and show that our method retains the preferable properties of scale-free network. Scale-free networks can have large node degrees due to the power-law distribution; therefore, to meet the physical properties of available network equipments, namely the number of ports, we artificially constrain the degree of network's nodes. We limit the node degrees by extending the original scale-free network generation algorithm of Barabási and Albert [5].

The main contribution of our work is threefold:

- we propose a scale-free network inspired data center architecture generating algorithm, which considers the physical constraints of network equipments,
- we show that limiting the degree of the nodes does not ruin the small diameter and error tolerance of scale-free networks,
- we illustrate that our design is highly scalable and flexible using simulation results.

The structure of the paper is as follows. We motivate our scale-free network inspired data center architecture in Section 2. The topology generation algorithm as well as the impacts of constraining degree are presented in Section 3 along with an illustration of the scalability and flexibility property. We discuss our proposal in Section 4, review the related work in Section 5 while Section 6 concludes the paper.



(a) BCube (b) DCell (c) Fat tree (d) Balanced tree

Figure 1: Illustration of state-of-the-art data center topologies; all of them are symmetric

2. MOTIVATION

The idea of connecting remote locations with communication systems, like telephone and computer networks, has basically changed the way how people live their lives. Although the technological designs of the most influential systems of the last century from POTS to mobile telephony are diverse they share a common characteristic: symmetry dominates their design. State-of-the-art data center architectures have also precisely designed, symmetric structures made out of homogenous equipments. As an illustration, we show four data center topologies in Figure 1, namely BCube, DCell, fat tree, and balanced tree, which all have symmetric designs.

However, innumerable examples exist in the nature where networks have asymmetric structure. From the protein of the cells throughout the cells and organs to the organisms and beyond asymmetric networks are formed. These networks facilitate favorable operation to fulfill their required behavior as they all survived the natural selection.

Surprisingly, some of the biological networks share a common structural characteristic: the degree distribution of the networks' vertices follows power-law distribution. These networks are called scale-free networks. Scale-free networks have two principle properties that may play an important role in their evolutional success. On the one hand, the diameter of scale-free networks is extremely small, namely it scales with $\ln \ln N$, where N denotes the number of nodes of the network [6]. Accordingly, signals traverse such networks quickly in average; therefore, it cannot happen that crucial information propagates slowly in the network [17]. On the other hand, scale-free networks are highly resistant to random failures, i.e. the diameter of the network does not increase until significant number of the nodes fails [4].

The required properties of data center networks are in accordance with the key characteristics of scale-free networks. Due to economics of scale operating data center networks is cost effective; therefore, several companies deployed their own architectures including Microsoft, Google, Yahoo, Facebook and Amazon [9]. Data center network architectures consist of tens of thousands or even more servers in order to provide services to their customers. On the one hand, a data center architecture has to be resilient to hardware failures as well as network outages in order to provide reliable services. On the other hand, data centers with shorter paths have fewer allocated resources as the data traverse fewer links and equipments; accordingly, the throughput capabilities of the system are enhanced.

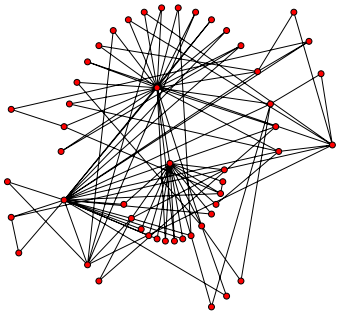
Therefore, an obvious question arises: is there any objection to design a data center network whose structure is asymmetric based on scale-free network topology? We believe that making reasonable modification on the original scale-free network paradigm, i.e. limiting the degrees of the nodes, it is possible to create a scale-free network inspired data center architecture. Contrary to the state-of-the-art data center architectures, these networks have asymmetric topologies with the properties of scale-free networks. In the following sections, we reveal and explain the details of this architecture and discuss its properties.

3. THE SCALE-FREE NETWORK INSPIRED DATA CENTER TOPOLOGY

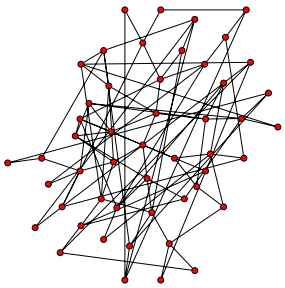
As we mentioned above, the properties of scale-free networks are favorable in case of data center design. Therefore, in this section we propose and analyze a data center topology generation algorithm called Scafida by modifying the scale-free network creation algorithm of Barabási and Albert [5]. Since the discovery that the World Wide Web has a scale-free network topology [3], several algorithms have been proposed that can generate scale-free network topologies [5, 1, 14]. One of them is the method of Barabási and Albert, which is also known as preferential attachment [5]. The network structure is generated iteratively, i.e. the nodes are added one by one; a new node is attached probabilistically to an existing node proportional to the node's degree. This phenomenon is also known as the richer gets richer principle.

Our method artificially constrains the number of links that a node can have, i.e. the maximal degree of the nodes, in order to meet the port number of network routers and switches. We note that node is used as a common term for servers and switches throughout the paper. One, who is familiar with the order of magnitude of scale-free networks' node degrees, probably sees immediately a contradiction between the limited number of switch ports in data centers and the unconstrained degree of the nodes in scale-free networks. However, surprisingly, although we constrain the maximal degree of the network, we are able to sustain the preferable properties of scale-free networks, namely short distances and high error tolerance! Moreover, the degree limitation does not affect significantly the bisection bandwidth of the topologies.

In order to visualize our data center architecture concept, in Figure 2 we show two topologies generated with our proposed method. The topology in Figure 2(a) presents an ordinary Barabási–Albert scale-free network, where the maximal degree of the nodes is not limited. Accordingly, some



(a) Original SFN (without degree limitations)



(b) Maximal degree: 5

Figure 2: Illustration of scale-free network inspired topologies

nodes have much more edges than the others; e.g., the three vertices in the middle have more than 10 links. In contrast, Figure 2(b) presents a topology where the maximal degree that a node may have is limited (5 links). The two topologies differ in their node degree distributions; however, their properties—as we will see shortly—are similar.

Before presenting qualitative proofs for this phenomenon, we first introduce our Scafida algorithm. Afterwards, we present the impact of constraining the degrees of the nodes on the paths' lengths and on the bisection bandwidths. Finally, we illustrate the flexibility of the proposed method based on extensive simulations; in particular, we show that the performance of Scafida networks tends to be similar as of the state-of-the-art data centers.

3.1 The Scafida Algorithm

To illustrate the 'unlimited' nature of the maximal node degrees in case of scale-free networks, we generated in total 300 scale-free network topologies, consisting of 1000, 10000, and 100000 nodes. In the experiment, the means of the maximal node degrees of the separate networks were 80.45, 257.5, and 816.35, respectively. However, switches and routers of network architectures have a quantified number of ports; the state-of-the-art commodity switches have 4, 8, 16, 24, or 48 ports, while the servers may have a few links. Accordingly, the original scale-free network generation algorithm has to be modified in order to meet the properties of the network equipments, i.e. the degrees of the network's nodes have to be constrained.

Our proposed algorithm generates a data center topology, which has determined number of servers and made out of the specified number of different switches. We apply the notations of the Barabási–Albert model, thus n_{t_0} denotes the number of servers. In the Barabási–Albert algorithm the network is growing iteratively, i.e. the nodes are added one by one to the network. Every newly added node has m links, whose target is selected proportionally to the degree of the nodes. In addition to the original model, we introduce new parameters; a t_i type switch has p_{t_i} ports, the number of t_i type switches is n_{t_i} $(i = 1, \ldots, k)$, while p_{t_0} denotes the number of ports a server has. We assume without loss of generality that the switches are ordered based on the numbers of their ports, i.e. if $t_i < t_j$ then $p_{t_i} < p_{t_j}$. We note that we use the term node in order to emphasize that only at the end of the algorithm turns out which nodes will be servers and which will be switches. In other words, the algorithm creates a logical network that can be physically realized at the end of the algorithm.

The pseudocode of Scafida is presented in Figure 3; in the followings we go through each line and explain the algorithm in detail. First, an empty graph is created that will store the topology of the data center (line 1) and m initial nodes are added to the graph (line 2). In order to match the number and type of the available network equipments, the values of a_{t_i}s are set to zero (line 3); i.e., at the beginning there do not exist any allocated equipment. Similarly, as an initialization, R is set to be an empty list (line 4). The preferential attachment principle is implemented using the R list, which stores the indices of the nodes in the network. R may store an index multiple times; e.g., if node v has d_v links it is stored d_v times in the list. Next, similar to the original Barabási–Albert algorithm, a new node is added to the network (line 5), denoted by index m; this node has m links whose targets are the initially added vertices. Except the initial nodes ($\{0, \ldots, m-1\}$), every newly added node has m links. Usually, the value of m is 2 or 3 in nature. These values are applicable in case of data centers as well; however, the operator is able to set the m parameter to influence the performance of the topology. We assume that $m \leq p_{t_0}$ holds, this assumption does not constrain the algorithm in general. Because R stores the ids of the nodes proportional to their degrees, the ids of the initial nodes are added to R once (line 6) while the id of node m is added to R m-times (line 7).

Afterwards, the nodes are added to the graph one-by-one, the id of the next vertex is denoted by b (line 8), until the network has the required number of network equipments (line 9), computed by summing the number of servers (n_{t_0}) and switches (n_{t_i}). As we have mentioned above, every newly added node (line 10) receives m links whose targets are selected based on the preferential attachment principle. T is used to store the ids of target nodes; the T list is initially empty (line 11). The target node selection procedure is executed until m different node is selected (line 12). Accordingly, a possible target node is selected from R (line 14); thus, proportionally to the degrees of the nodes. The id of the target node is picked continuously until a vertex is found that has not been picked previously; i.e., it is not included in T (line 15).

Next, we have to check whether the picked target node has

empty ports (line 16). If the degree of the selected v_t node does not equal to the port number of the network equipments (line 17), the id of the target node is added to T and a new edge is created between the actual b node and the picked v_t node (line 18). Otherwise, let us assume that $d_{v_t} = p_{t_i}$ for easier notation (line 19); this can be done without loss of generality. Next, it has to be determined whether the number of ports of v_t can be increased. This is done by using two variables, one for the number of allocated switches that have more ports (line 20), and one for all the larger switches (line 21). Only those switches are summarized that have more ports than v_t (line 23). If the port number of v_t can be increased (line 26), first the values of allocated switches are updated (line 27), then the id of the target node is stored in T (line 29) and the new edge is added to the network (line 30). Note that the R list is not updated yet, it will be only updated after all the target nodes are determined. On the other hand, if the number of ports cannot be increased (line 32), the id of the picket target node is removed from R as it cannot have additional links.

Finally, after m target node is picked R has to be updated before adding the next node to the network. The ids of the target nodes are added once to R (line 33) while the id of the new node b is added m-times (line 35). Afterwards, the id of the new node is incremented for the next round (line 36).

3.2 Impact of constraining degree

Next, we show that constraining the maximal degree of the nodes does not impact significantly the properties of scale-free networks. As the servers of a data center communicate with each other, the average length of the paths between the nodes fundamentally impacts the performance of the data center network. In Figure 4(a), we present the impact of limiting node degrees on the average shortest path lengths, computed by dividing the sum of the shortest path lengths with the number of the paths. We note that the servers may be involved in the routing process if their degrees are larger than one. The presented values are averaged over 50 topologies for each case, where the value of m was set to 2; as the deviations of the results are negligible they are not shown in the figure for better understanding. We used port numbers of commodity switches to constrain the maximal node degrees; NL denotes the topologies without degree limitation. Irrespective of the size of the networks, the average lengths of the paths increase moderately due to the constrained degrees; in most cases the increment of the lengths is less than an additional hop.

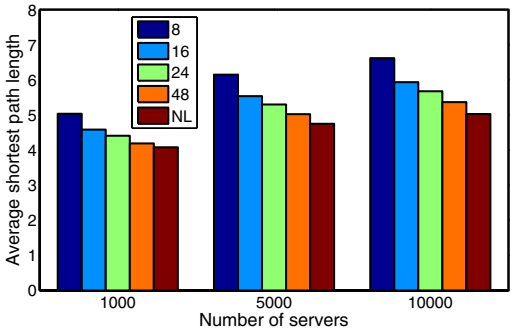
Another crucial aspect of a data center architecture is its throughput capability. Some applications (e.g., MapReduce [8]) require intensive communication among the servers of the data center; bottlenecks in the topology would cause performance degradation. The throughput capability of a data center can be measured with bisection bandwidth. The servers are divided into two groups, in total 200 times in our analyses, afterwards the maximal flow between the two parts is computed; the capacity of the links in the network were set evenly. Figure 4(b) shows the impact of degree limitation on the distribution of bisection bandwidth in case of data centers with 5000 nodes. The distribution of bisection bandwidth is almost irrespective of the degree limitation. In the analysis we assume that the capacities of the links are equal and the servers and switches have enough throughput

Input:
n_{t_0} — number of servers
p_{t_0} — number of servers' ports
$n_{t_1}, \ldots, n_{t_k}$ — number of t_i type switches
$p_{t_1}, \ldots, p_{t_k}$ — number of ports of t_i type switches
a_{t_i} — number of t_i type switches already allocated
d_v — degree of the node $v \in V$
m — number of links a newly added node has

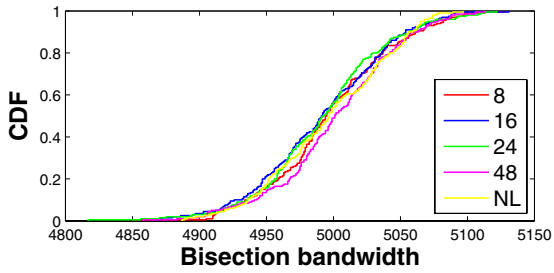
Algorithm

1 $G = (V, E)$ // an empty graph
2 $V = V \cup \{0, 1, \ldots, m-1\}$ // add initial nodes
3 $a_{t_i} = 0; \quad i = 0, \ldots, k$ // initialization
4 $R = \{\}$ // used for preferential attachment
// add initial links to the network
5 $E = E \cup \{(m, 0), (m, 1), \ldots, (m, m-1)\}$
6 $R = R \cup \{0, \ldots, m-1\}$ // update the index list
7 $R = R \cup \{m, \ldots, m\}$ // m times
8 $b = m + 1$ // the index of the next vertex
9 **while** $b < \sum_{i=0}^{k} n_{t_i}$ **do**
10 $V = V \cup \{b\}$ // add the node to the graph
11 T = {} // store the selected target nodes
12 **while** $|T| < m$ **do**
13 **repeat**
14 $v_t = random(R)$ // a random item of R
15 **until** $v_t \notin T$
16 **if** $d_{v_t} \notin \{p_{t_0}, \ldots, p_{t_k}\}$ **then**
17 $T = T \cup \{v_t\}$
18 $E = E \cup \{(b, v_t)\}$ // add the edge
19 **else**
// let $d_{v_t} = p_{t_i}$ w.l.o.g.
// determine whether the switch can be extended
20 $nasw = 0$ // number of allocated larger switches
21 $ntsw = 0$ // total number of larger switches
22 **for** $j = 0, \ldots, k$ **do**
23 **if** $p_{t_j} > p_{t_i}$ **then**
24 $nasw+ = a_{t_j}$
25 $ntsw+ = n_{t_j}$
26 **if** $nasw < ntsw$ **then**
// the target switch can have more ports
27 $a_{t_i} = a_{t_i} - 1$
28 $a_{t_{i+1}} = a_{t_{i+1}} + 1$
29 $T = T \cup \{v_t\}$
30 $E = E \cup \{(b, v_t)\}$
31 **else**
32 $R = R \setminus \{v_t\}$
// update the index list
33 $R = R \cup T$
34 **for** $i = 1, \ldots, m$ **do**
35 $R = R \cup \{b\}$
36 $b = b + 1$

Figure 3: The pseudocode of Scafida algorithm



(a) Path lengths



(b) CDF of bisection bandwidths (5000 nodes)

Figure 4: The impact of constraining the degree of the nodes

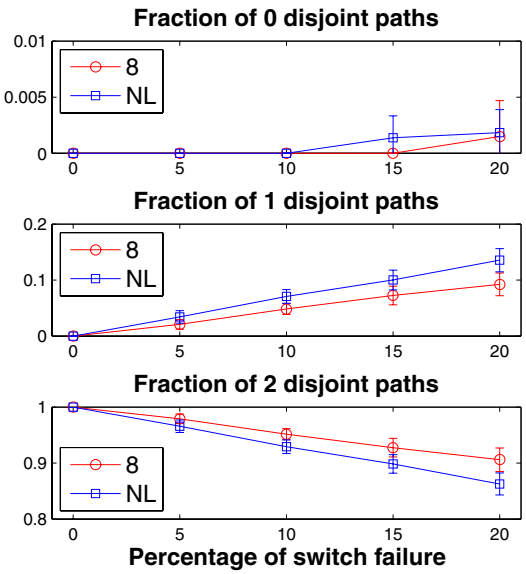


Figure 5: Scafida tolerates the failure of switches well; the impact of node failures on the number of disjoint paths is shown

capabilities, thus they do not constrain the performance of the topology.

Based on the presented results we state that our data center architecture generation method, which constrains the degrees of network's nodes in order to meet the sizes of commodity switches, has preferable properties although the degrees of the nodes are artificially limited.

3.3 Error tolerance

Due to their specific structure scale-free networks tolerate efficiently the random failure of nodes [4]. Stochastic failure of network equipments is ordinary in data centers because of their size; therefore, if the constraining of the node degrees alters this desirable property the applicability of Scafida would be limited. Fortunately, the error tolerance of our data center architectures is similar to the original scale-free networks.

We quantify the impact of constraining degree on the resilience of the architectures by investigating the distribution of the number of disjoint paths between every pair of servers. A fraction of the switches fails in the networks; the percentage of the failed switches is scaled from 0 to 20 in steps of 5. 1000-node Scafida topologies are analyzed using several error scenarios; the results are averaged over these simulations. As we assumed that the servers have two ports, the number of disjoint paths between two servers can be at most two. Accordingly, instead of showing the cumulative distribution function of the disjoint paths, in Figure 5 the ratio of server pairs is plotted; the different figures present different number of disjoint paths. Two extreme cases are shown, one where the degrees are not constrained (i.e., an original scale-free network) and one where the maximal degree is 8. The error tolerance of Scafida is presented best on the last figure, where the results for two disjoint paths are shown. Even if 20 percent of the switches fail more than 90 percent of the server pairs still have two disjoint paths. We note that the fraction of cases where there does not exist a connection between two servers is quite low; the scales of the figures are not even in order to visualize all the results. The error tolerance of Scafida is even better than that of the original scale-free networks because the impact of a failure can be significant in case of a failure of a switch with large degree.

The servers connect only to switches due to the way how the topology is generated. When the first server is added to the network, it connects to the network with two links as $m = 2$. Both the links connect to switches because this is the first server to be added. As the number of server's ports is two ($t_{p_0} = 2$), this server does not have any unused ports. Therefore, it will not have any additional links. Thus, if we add the next server to the topology, it can only connect to switches, etc. Due to the large number of servers and knowing that the servers connect only to switches, the possibility that a server gets disconnected although it is not failed is extremely small; in addition, the probability of this event gets more and more lower as the size of the network increases.

3.4 Flexibility

The Scafida topology generation method is extremely scalable and flexible. On scalability we mean that using the proposed method any size of data center can be created; it can generate a topology with 1000 servers but also with 50000 servers. In addition, the size of the data center can be set on a fine-grained scale. State-of-the-art data center architectures can be created using only few parameters; therefore, the number of servers in the structure can be set only on a course-grained scale. On the other hand, the Scafida algorithm is extremely flexible in terms of the type and the number of network equipments (e.g. servers and switches).

As Scafida is able to produce data centers out of different types of switches, in the followings we illustrate the flexibil-

ity of the proposed algorithm. Due to its flexibility, Scafida networks can be created out of the parts of state-of-the-art data centers. Table 1 shows some promising preliminary results, namely the performance of our data center structures, including the average shortest path lengths, diameters, and bisection bandwidths, is comparable with the performance of state-of-the-art structures. Pairwise, the topologies have the same number of servers and switches. Therefore, the proposed method is able to generate data center structures flexibly without significantly degrading the performance of the network. We note that in order to meet the actual degree distribution of the state-of-the-art data centers, after the end of the proposed algorithm we created additional links between the network's nodes using the preferential attachment principle, in order to use all the ports of the network equipments.

The performance of Scafida is comparable with the state-of-the-art architectures based on the distribution of bisection bandwidths and shortest path lengths as well. In case of bisection bandwidths, the servers are partitioned 200 times into two groups, then the bandwidth between the two groups is determined. Based on the simulation results, we compute the cumulative distribution function of the bisection bandwidth values; the results are shown in Figure 6. In all cases, the two topology generation methods provide similar bisection bandwidths; the largest difference is in case of BCube but it is still marginal (5 percent). The distribution of the paths is computed using the shortest paths between every pair of servers. The plots in Figure 7 show similar characteristics for the methods again. Despite Scafida's asymmetric nature, the path lengths in Scafida architectures are quite comparable to their symmetric counterpart.

4. DISCUSSION

After presenting the generation method and the properties of Scafida, we discuss the implications of the results and the limitations of the structure. In terms of implications, first we want to emphasize that the properties inherited by scale-free networks are crucial in data center networking. On the one hand, short paths results shorter latency and may enhances the throughput performance of the system. On the other hand, due to the large number of equipments, failures happen round the clock; accordingly, the high error tolerance of the proposed method could motivate its application in practice.

Besides these aspects, another significant implication of the structure is that data centers can be built out of any number of network equipments; i.e., the structure is highly scalable and flexible. The presented results are based on the properties of currently available commodity switches; however, as the size of these equipments increases the properties of our data center structure enhance because it will approximate more and more the unconstrained scale-free networks. Due to the fact that preferable properties of scale-free networks holds in case of low node degrees too, Scafida can also be applied in data centers, where the servers are interconnected with each others [7]; the usage of our method may extremely reduce the diameter of these structures.

In terms of limitations, the wiring of the proposed architecture can be a complex task, especially if the number of ports of network elements is large. Scale-free networks tolerate random failures well; however, they are vulnerable to attacks when nodes with the highest degrees are damaged. Therefore, operators of the data centers have to pay special attention to protect their architectures; e.g., by deploying effective firewalls.

Not only the structures of the data centers but also the routing algorithms affect the performance of the systems. Therefore—as a future work—we plan to design an appropriate routing algorithm that exploits the benefits of the architecture and evaluate a prototype of the whole system.

5. RELATED WORK

Several data center architectures have been proposed recently; all of them are based on symmetric structures. Data centers based on fat-tree topologies [2, 16, 10] are built using commodity switches arranged in three layers, namely core, medium, and server layers. The structure is also known as Clos topology. The fat-tree topology can be built using commodity Ethernet switches [2]; where the flows are leveraged using multiple paths. Portland [16] is a scalable, fault tolerant layer-2 data center built on multi-rooted tree topologies (including fat tree as well) that supports easy migration of virtual machines. VL2 [10] is an agile data center architecture with properties like performance isolation between services, load balancing, and flat addressing.

The symmetric structure of the BCube [11] data center architecture is designed using a recursive algorithm. BCube is intended to be used in container based, modular data centers, with a few thousand servers. MDCube architecture proposes a method to interconnect these containers to create a mega-data center [18].

Similar to BCube, DCell is also built recursively [12]. DCell's recursive expression scales up rapidly, thus DCell can have enormous number of servers with small structural levels and switch ports. In DCell, not only the commodity switches but also the servers are responsible to route the packets in the data center.

Not only the throughput capabilities but also the power consumption of the data center architectures are diverse [13]. Power consumption of data centers is a crucial challenge that has to be addressed [9], a partial solution may be reducing the power consumption and heat dissipation of servers and switches [15]. Scale-free network inspired data centers can be energy efficient due their highly scalable and flexible design. As any size of data center can be created using Scafida, our proposed structure is energy proportional; i.e., the power consumption of the network is proportional to the number of servers in it.

The phenomenon that numerous biological networks have similar degree distribution was first identified in case of the World Wide Web [3]. The degree distribution of these networks follows power-law distribution, where the probability that a node has k links can be estimated as $P(k) \sim k^{-\gamma}$. The value of γ is usually between 2 and 3 in case of biological systems. Barabási and Albert identified two conditions that are necessary to form a scale-free network [5]. On the one hand, the nodes are added iteratively to the network, while, on the other hand, a new node picks its neighbors based on the nodes' degrees. Scale-free networks were intensively analyzed in the last decade; it has been shown analytically—among others—that if a scale-free network has N nodes its diameter is proportional to $\ln \ln N$ [6]. Furthermore, based on simulation results it was presented that scale-free networks tolerate random failures; however, they are extremely vulnerable to attacks [4].

Table 1: Scafida topologies have similar performance as state-of-the-art data centers using the same switches and servers

Topology	Number of servers	Path length, mean	Diameter	Bisection bandwidth, mean
BCube	4096	7.00	8	5953.85
Scafida BCube	4096	5.80	8	5796.45
DCell	2352	4.84	5	1628.05
Scafida DCell	2352	4.72	12	1600.02
Fat-tree	3456	5.91	6	1728.0
Scafida Fat-tree	3456	4.74	7	1718.60
Balanced tree	2304	3.96	4	1039.98
Scafida Balanced tree	2304	6.14	11	1041.12

6. CONCLUSION

In this paper, we proposed Scafida, a novel data center structure generating algorithm inspired by scale-free networks. Our method constrains the degrees of network's nodes to meet the physical capabilities of commodity switches. Based on simulation results we presented the impact of limiting the node degrees. Surprisingly, our method provides almost the same properties like scale-free networks; i.e., short distances between the nodes and high error tolerance, which are favorable in the context of data centers. Moreover, despite its asymmetric structure Scafida's properties are similar to that of state-of-the-art data centers. Although we are still working towards the first prototype of Scafida, we believe our proposed solution will be a feasible architecture, preferred by both the research community and the industry.

Acknowledgement

The authors would like to thank the anonymous reviewers for their insightful comments. This work has been partially supported by HSNLab, Budapest University of Technology and Economics, `http://www.hsnlab.hu`.

7. REFERENCES

[1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180, New York, NY, USA, 2000. ACM.

[2] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *SIGCOMM '08*, pages 63–74, 2008.

[3] R. Albert, H. Jeong, and A.-L. Barabási. Internet: Diameter of the World-Wide Web. *Nature*, 401(6749):130–131, September 1999.

[4] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–82, August 2000.

[5] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.

[6] R. Cohen and S. Havlin. Scale-free networks are ultrasmall. *Phys. Rev. Lett.*, 90(5):058701, Feb 2003.

[7] P. Costa, T. Zahn, A. Rowstron, G. O'Shea, and S. Schubert. Why should we integrate services, servers, and networking in a data center? In *WREN '09: Proceedings of the 1st ACM workshop on Research on enterprise networking*, pages 111–118, New York, NY, USA, 2009. ACM.

[8] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, San Francisco, 2004.

[9] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, 2009.

[10] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Vl2: a scalable and flexible data center network. In *SIGCOMM '09*, pages 51–62, 2009.

[11] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. Bcube: a high performance, server-centric network architecture for modular data centers. In *SIGCOMM '09*, pages 63–74, 2009.

[12] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: a scalable and fault-tolerant network structure for data centers. In *SIGCOMM '08*, pages 75–86, 2008.

[13] L. Gyarmati and T. A. Trinh. How can architecture help to reduce energy consumption in data center networking? In *e-Energy '10*, pages 183–186, New York, NY, USA, 2010. ACM.

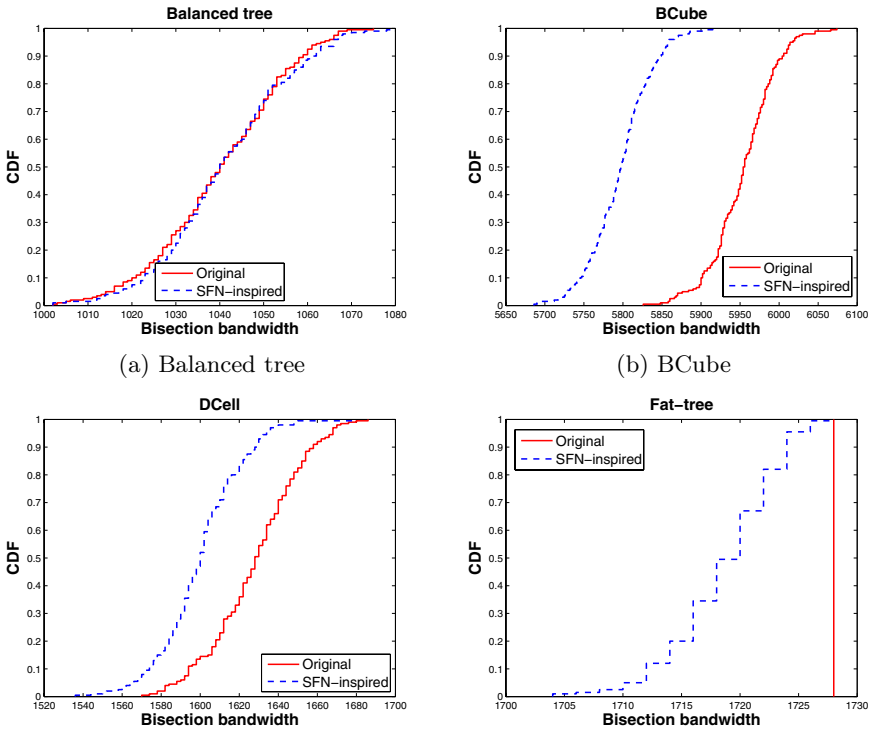
[14] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, 64(2):026118, Jul 2001.

[15] D. S. Nikolopoulos. Green building blocks - software stacks for energy-efficient clusters and data centers. *ERCIM News 79*, 2009.

[16] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. In *SIGCOMM '09*, pages 39–50, 2009.

[17] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86(14):3200–3203, Apr 2001.

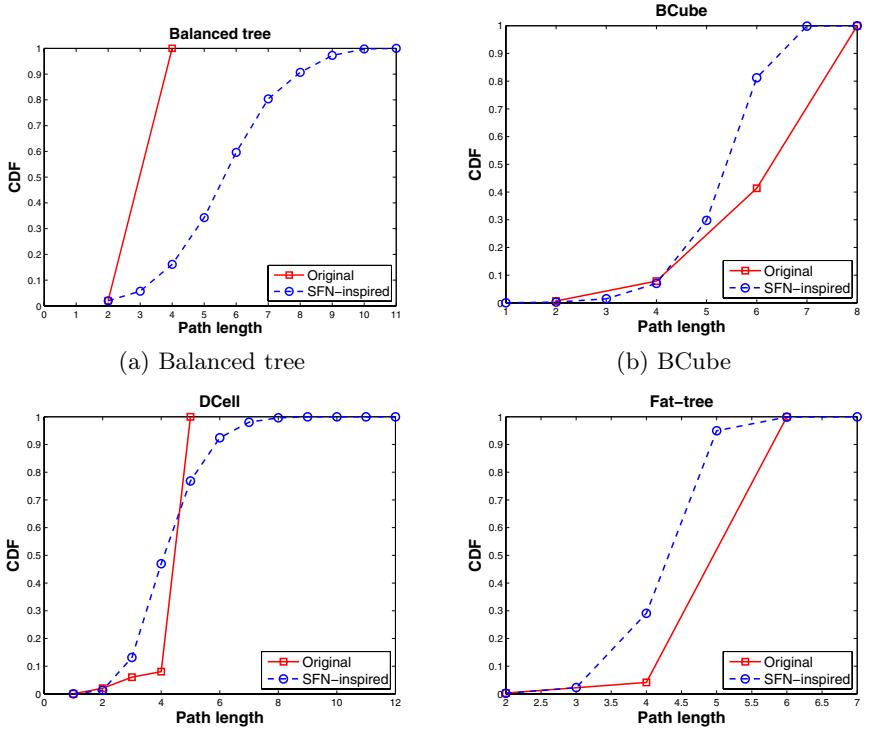
[18] H. Wu, G. Lu, D. Li, C. Guo, and Y. Zhang. Mdcube: a high performance network structure for modular data center interconnection. In *CoNEXT '09*, pages 25–36, New York, NY, USA, 2009. ACM.



(a) Balanced tree (b) BCube

(c) DCell (d) Fat tree

Figure 6: Cumulative distribution function of bisection bandwidths in case of state-of-the-art and analogous scale-free network inspired data centers



(a) Balanced tree (b) BCube

(c) DCell (d) Fat tree

Figure 7: Cumulative distribution function of path lengths in case of state-of-the-art and analogous scale-free network inspired data centers