Legacy Data Center Network (DCN) Reviewed: considering Big Data Stream Mobile Computing (BDSMC) applications

1Godspower I. Akawuku, 2Rapheal O. Okonkwo, 3Keneddy C. Okofor
1,2Nnamdi Azikiwe University, Awka (NAU), Nigeria.
3Federal University of Technology, Owerri (FUTO), Nigeria.

ABSTRACT

Recently, Enterprises that operate over vast geographical areas uses multiple data centers to collect, store and process data in real time via energy-efficient acquisition, wirelessly transport clients or users data to the cloud. This paper seeks to technically review existing Data Center Networks (DCN) considering BDSMC applications. In this research, related works on Distributed Data Center Networks will be presented. Within the stream computing ecosystems, there are various network models but the pool of possible DCN topologies/architectures to adopt appears little and unfit for the purpose of BDSMC optimization. However, most related Data center architecture will be reviewed. Investigate efforts on both server centric and switch-centric models adaptable to BDSMC network layer. The extent of work done on distributed spine-leaf re-designed server-centric network construction so as to automatically harvest network interconnection into a ‘stellar’ dual-port server-centric SG network; how classical graph-based interconnection network translate network performance similar to generic works for BDSMC ecosystems. Review stellar transformation using the well-studied generalized hypercube family of interconnection networks for BDSMC ecosystems. The literature was searched from the databases: IEEE Xplore Digital Library, Springer Link Digital Library, and Google Scholar, IET Digital Library, Frontiers Library, ACM Digital Library repositories resulting in 98 papers after several eliminations ranging from year 2000-2022. In conclusion, state-of-the-art dual-port server-centric DCNs (FiConn, DCell, DPillar), etc, while looking at possible architectures with excellent comparative performance for BDSMC ecosystems. Research gaps are revealed for further study.
Keywords: Data, Center, Network, Data Stream, Mobile Computing and applications.

INTRODUCTION

Large scale organizations that operate over vast geographical areas uses multiple data centers to collect and store data from their clients or users by [1] For example, most transport companies collect data from different departments that operate in remote locations. The streamed data is usually stored in multiple data centers, and its volume grows rapidly, often exceeding terabytes in each data center in the works of [2, 3]. In many cases, these organizations analyzes their big data in multiple data centers in a batch for analytics and business decisions widely discussed in [4]. When taking data in multiple data centers into consideration as a whole, analyzing such big data efficiently and effectively becomes a big challenge for service offloading. Apache [5, 6, 7, 8, 9] are the most popular frameworks for parallel and distributed big data stream processing and analysis says [10]. However, these frameworks are designed to process data locally within the same data center. Hence, they are in need to copy all data to a single data center before processing a locally distributed computation in [11]. BDSMC explains a new generation of mobile or wireless integrated computational networking infrastructures designed to extract hidden value from an ever-increasing volume of space-time
correlated heterogeneous data streams in [12]. This is enabled in real time via energy-efficient acquisition, wireless transport, and processing. The characterization of the BDSMC paradigm expresses (i.e., variety (data heterogeneity), volume (ever increasing amount of data to be processed), velocity (data generation at fast and unpredictable rates), value (huge value but hidden in massive datasets at very low density), and volatility (the acquired data streams must be transported and processed in real time). While the first four Vs are common to all big data applications, another last V (i.e., volatility) is introduced for featuring big data stream applications. In general, the value of a stream of data is closely related to both its space and time coordinates, and hence, after acquisition, this value quickly decreases if the computing-plus-communication delay is larger than a suitable quality of service (QoS)-dictated hard threshold.

**Non-BDSMC Fat-Tree Architecture**

Figure 2.0 illustrates a typical Fat-tree architecture widely used in most enterprise networks. In [13] the authors presented how to leverage largely commodity Ethernet switches to support the full aggregate bandwidth of clusters consisting of tens of thousands of elements. A simple topology was studied for streams propagation in Figure 1.0 which is not useful in BDSMC ecosystems depending on the scale. Most legacy networks use Fat-tree data structure in which top branches have stronger/thicker branch than others in the design hierarchy. These data links branches vary in thickness (bandwidth) while allowing the links for more efficient and technology-specific use. Mesh and hypercube topologies uses common design requirements based on rigid algorithm.

[14] a special case of Clos network, also referred to as a folded Clos network was depicted as fat-tree data center topology. Originally, the fat-tree topology principle was suggested for supercomputing and was modified for data center networks. In terms of the number of pods, a fat-tree topology is usually referred. This is numerated from Pod-0 to Pod-(k-1) from left to right. The topology consists of k pods with three switch layers: edge switches, switches for aggregation, and switches for the center. Thus, there are k switches (each with k ports) in each pod arranged in two layers of k/2 switches in k pod fat-tree topology, one layer for edge switches and the other for aggregation switches. The k/2 aggregation switches are attached to each edge switch. (k/2)2 core switches are available, each of which connects to k pods. The topological knowledge description is shown in Table 1.0.
Table 1.0: Fat-tree Topology

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Pods</td>
<td>K</td>
</tr>
<tr>
<td>Number of middle (Core) Switches</td>
<td>(k/2)^2</td>
</tr>
<tr>
<td>Number of gateway Switches</td>
<td>K^2/2</td>
</tr>
<tr>
<td>Number of Edge end Switches</td>
<td>K^2/2</td>
</tr>
<tr>
<td>Number of rack Switches, N (all types)</td>
<td>5K^2/4</td>
</tr>
<tr>
<td>Number of connection links, L</td>
<td>K^3/2</td>
</tr>
<tr>
<td>Number of Supported Hosts</td>
<td>K^3/2</td>
</tr>
</tbody>
</table>

For \( k = 4 \), Figure 2 shows the fat-tree topology. Here, the bottom 8 edge switches (nodes), numbered 1 to 8, have 16 supported physical hosts. This is known as the outcome of \( K + L \). Leading to multipath measure (MPM) in Equ 1 [15].

\[
MPM = \text{Min} \left\{ \frac{L-1}{K}, 1.0 \right\}
\]

In Figure 2, the topology is shown for 6 pods.

There are some advantages of fat-tree topologies. With the same number of ports, all switches are of the same form, with each port usually having the same speed; the end hosts often support the same speed. Between any two hosts, there are several routes. It is important to link each host to an edge switch first. There are two paths between two edge switches within a
pod (intra-pod) in a 4-pod fat-tree topology and there are four paths between any two edge switches that are across pods. This model is not scalable for BDSMC ecosystem. Various efforts in literature seem to isolate BDSMC application. For instance, the work of [16] focused on active job placement and routing algorithms in the Fat-Tree Topology. Similarly, the works [17, 18, 19, 20] explored the various deployments of fat-tree without any application in BDSMC system.

Non-BDSMC Dahu Datacenter Architecture

Dahu DCN was described in [21] as an improvement on networks powered by commodity Ethernet switches which supports direct link networks. By dynamically distributing traffic equally across links, this model eradicates congestion points. When performing load balancing using local data, it does traffic forwarding over non-minimal routes. HyperX topology, a direct connection network for detailing how Dahu’s hardware primitives are used and tested has been studied. A related work on HyperX topology was addressed in [22], the architecture of which is shown in Figure 3. HyperX is an L-dimensional direct network between any server pair with several paths of varying length. It can be interpreted as a Hyper-Cube topology generalization. The Hypergraph architecture is scalable but not optimized for BDSMC application.

Figure 4: HyperX topology (L=2, S=3)
Dahu new switch DCN mechanism for dynamic traffic hashing allows various network paths to be hashed and exploits non-shortest path forwarding to minimize congestion while using novel hardware switch primitives and control software to avoid persistent forwarding loops. However, the decentralized load balancing heuristic is not scalable for BDSMC environments (i.e., for rapid local decisions to alleviate congestion. BDSMC data center implementations would obviously be primarily restricted to multi-rooted tree switch-centered topologies such as [24] as illustrated in Figure 5. Intelligent switches are used in Switch-Centric category to conduct smart packet routing in a Data centers. In this category, some switch-centric data center topologies include: Clos-Network in the works of [25, 26, 27, 28, 29, 30], among others. These will be further reviewed in this Section.

Figure 6: VL2 Architecture [31].
Non-BDSMC Clos Network

The authors [32], addressed the design of a three-stage buffered Clos-network switch (TSBCS) along with a new mechanism for batch scheduling (BS). The TSBCS/BS is mapped to a "fat" combined input-cross-point queuing (CICQ) switch with a directly implemented CICQ scheduling algorithm in TSBCSS.

Similarly, in [33], a parallel Wavelength Fault Tolerant Clos network, PW-FTC, is addressed. This was used to achieve Fault Tolerant Clos-network (FTC) planes, as shown in Figure 8, a wavelength switch is performed.

Jelly Fish Architecture

In the top-of-rack (ToR) switch layer, the Jellyfish DCN schemes are based on the construction of a random graph. In this case, there are a number of ports for each ToR switch that connect to other ToR switches, while the remaining server ports are used by [35].
In this section, another class of sophisticated DC models will be reviewed. The authors [37], presented the Datacenter Optical Switch (DOS) designed for scalable and high-throughput data center interconnections, but not for the BDSMC device. The architecture is based on an arrayed waveguide grating router (AWGR) that enables the wavelength domain to overcome contention. The likelihood of lower latency and higher throughput even at high input loads has been illustrated by its switching architecture. DOS has however, not been used in any known BDSMC systems. To date, this has never been extended to BDSMC data center networks.

**Datacenter Optical Switch (DOS) Architecture**

Hybrid Packet/Circuit (HYPAC) switched DCNs were introduced optical Fibre communication DCN owing to its merits of bandwidth capacity and power efficiency says [39]. Figure 2.10 uses their proposed Collaborative Bandwidth Allocation (CBA), which optimally sets the network up. A rack-to-rack optical circuit switched (OCS) network complements the electric packet switched (EPS) network.
The authors [41], presented a flattened butterfly for high-radix networks as a cost-effective topology in Figure 12. This is useful in load-balanced traffic where, relative to the Clos network, its efficiency is very optimum. The advantage over the Clos is achieved by removing redundant hops when they are not required for load balancing. It uses high radix flattened butterfly topology that offers greater route diversity than a traditional butterfly and has around half the cost on balanced traffic of a comparable Clos efficiency network. This network is used when the level or radius of interconnection networks is increased by an increase in integrated-circuit pin bandwidth. Again, the network integration of BDSMC systems may encounter computational overhead here.

**FRINGE Architecture**

The research in [11], used Software-Defined Networking (SDN) as a smart tool, called FRINGE, to derive an effective software-define edge control system. The concept is to boost the Ethernet DCN's scalability. SDN domains on commodity access switches at the edge of the DCN are derived from the FRINGE architecture i.e.: Top-of-Rack (ToR) switches. The value is that it can be extended to DCN topologies that are arbitrary. It has access switches, recognized OpenFlow-enabled ToR switches (OFToR) at rest, or installed at the edge of DCN. The other portion of FRINGE
is the upper DCN for legacy switches/routers. The logical centralized SDN controller is used in the design to manage DCN states, such as mapping connections between OFToRs and hosts.

![Diagram of FRINGE framework](image)

**Figure 13:** FRINGE framework [43].

**Layered Scalable Data Center Architecture**

The authors introduced a data center topology in [44] called LaScaDa (Layered Scalable Data center) for the construction of scalable and cost-effective networking infrastructures for data centers. Their architecture configures nodes in uniform clusters and then interconnects the clusters in an ordered manner with a system of coordinates for nodes just to reduce the number of redundant links between clusters, thus optimizing communication. The LaScaDa forwarding packets between nodes are shown in Figure 14 using a new hierarchical row-based routing algorithm. A LaScaDa network constructed from n-port switches is a layered and recursive topology that uses n-port switches to create a k-layer LaScaDa network ($k > 1$) by interconnecting $n^{32(k-1)}$-Layer LaScaDa networks. These switches are designated as internal switches asserted [45]. The architecture uses its algorithm to construct the route of the source on the bases on the differences between the coordinates for the source and destination. In addition, a large number of nodes are connected using a small node degree by the proposed topology.

![Diagram of LaScaDa network Architecture](image)

**Figure 14:** LaScaDa network Architecture [46].
The authors [48], considered low-delay switches with end-to-end latency in large-scale High-Performance Computing (HPC) interconnect with cable delays. Skywalk was the new architecture created to satisfy the deployment (HPC systems). Skywalk uses randomness to optimize low latency which is very necessary in BDSMC network designs. This is done in a manner that accounts for the physical layout of the topology in order to cascade further cable length with latency reductions as shown in Figure 17.

Clearly, vast numbers of servers must be interconnected by the BDSMC network architecture. But in meeting the requirements of BDSMC Data Center Networks, conventional tree-based architectures seem unstable and inefficient (DCNs). These architectures fall into two groups, namely: switch-centric designs and server-centric designs, taking into account whether the interconnection intelligence is provided on the switches or on the servers (on-going). So far, it is evident that the switch functionality must be expanded to meet the need for interconnection in the switch-centric designs highlighted above, although servers do not need to be modified for interconnection purposes. As such, it will need high-end smart switches in a reliable BDSMC network, though this will significantly increase the interconnection cost.
Server-centric Datacenter Architectures

Interestingly, BDSMC systems will rely on server-centric designs to ensure robust server data stream integration. The main server-centric models in literature include [50]: DCell in Chuanxiong [51, 52, 53, 54]. General Hypercube in [55], HSDC [56], and Stellar Transformation in [56]. These have been studied and discussed in this Section as part of related works.

DCell Architecture

A clear challenge in DC networking boarders on efficient interconnection of the number of servers is exponentially rising especially in the case of BDSMC. The authors [3] discussed the DCell as a recursive structure with a high-level DCell that is being built in a completely interconnected fashion from several low-level DCells linking other DCells at the same level. As the node degree increases, their work scales doubly exponentially. The research clarified that since it has no single point of failure, DCell is fault tolerant and may be useful in BDSMC application. Except in the case of critical extreme connection or node failures, its distributed fault-tolerant routing protocol typically performs similar to shortest-path routing. Illustration 18 demonstrates the DCell architecture showing the network capacity for different types of services that appears better than the conventional tree-based framework. To build its recursively specified architecture, DCell uses servers equipped with multiple network ports and mini-switches. A server is linked to several other servers in DCell and a mini-switch through communication links, which are bidirectional by design. A more robust algorithm is needed for BDSMC application traffic density in DCell. This is missing from DCell legacy designs.

The authors [14, 17], discussed the BCube network architecture typical built and customized Figure 19 shows modular data centers for shipping-container-based data centers. The architecture has a server-centric network structure, central or at the heart. This is where the servers connect to several layers of Commercial-off-the-shelf (CoTS) commodity mini-switches with multiple network ports. The servers act both as hosts and as relay nodes for each other in their architecture. By accelerating one-to-one, one-to-many and one-to-all traffic patterns, and by providing high network capacity for all-to-all traffic, their architecture is assisted by several bandwidth-intensive applications. The main problem with BCube, as the server and/or switch failure rate rises, is that it demonstrates incremental performance deterioration. For BDSMC application, this is inappropriate. In fact due to the sealed and operational status that makes it extremely difficult to fix or replace its components, this behavior is important for shipping container data centers. It’s metrics includes seamless integrated with the TCP/IP protocol stack, efficient packet forwarding in both hardware and software, fault tolerance, load balancing and bandwidth-intensive application supports.
The authors presented FiConn network architecture in [25], which uses both ports and low-end commodity switches to derive a highly efficient and scalable structure. The server node degree is two in its layout in the design, but the design is scalable to include hundreds of thousands of low diameter and high bisection width servers. Many server machines are interconnected by FiConn DCN with low equipment cost, high and balanced network power, and link/server fault robustness. Their framework appears to deliver low-overhead traffic-aware routing mechanisms based on dynamic traffic state to enhance efficient connection utilization. Both the FiConn2 recursive and the extended recursive architectures. It can be inferred that the computational overhead in BDSMC active design is a major challenge.

Figure 20a: FiConn, Recursive Architecture (Liu et al., 2018).
An interesting server-centric data center network was found in [15] where many servers were interconnected based on optimized cost devices. Though expandable with a smaller level of regularity and symmetry, centered on the 2nd degree server model, their model leveraged high-end switching systems called hierarchical irregular compound network (HCN) and bi-dimensional compound networks (BCN). Low overhead and robust routing schemes for large server deployment were considered in the design. In the modular DCN architectures of data centers, the HCN offers great potential for a high degree of regularity, scalability and symmetry. The BCN provides the most scalable network structure with characteristics such as low diameter, high width of bisection, large number of one-to-one traffic node-disjoint paths, and strong fault-tolerant capacity. These architectures are ideal with desirable properties, but for smart grid structural integrations, this represents great computational overhead.

The research of [17] discussed DPillar as a massively scalable datacenter interconnection architecture that leverages low-end off-the-shelf commodity PC servers and switches. The DPillar uses minimal resource based on low-cost plug-and-play layer-2 nodes whose servers use dual-port commodity PCs. Its major attribute is that DPillar scales to limitless number of servers without needing physical server upgrades. The structural layout is shown in Figure 22. The challenge with DPillar is the computational workload concerns with the deployed setup. This is
not efficient for Smart grid integration model.

Figure 22: Two dimensional and vertical DPillar Architecture [10]
Switch-nodes are represented by Independent Squares; server-nodes are represented by dots. NB: the left-most & right-most server columns are the same (server-column 0))

**MCube Datacenter Architecture**
The work in [14] discussed MCube model which is a server-centric network architecture primarily designed for data centers. MCube uses commodity mini switches in offering high aggregate bandwidth while delivering cost driven performance that is fully backward compatible with Ethernet, IP, and TCP. It features fault tolerance since it does not have single point of failure and its source routing protocol performs near shortest-path routing.

![MCube Architecture](image-url)

**Figure 23: MCube Architecture [14].**

**BCDC Data Center Architecture**
In [11] BCDC network depicted in Figure 23 was designed as server-centric data center network feasible experimental testbed. Though their work highlighted superiority over DCell and FatTree implementation, fault-tolerant routing, and node-disjoint paths BCDC performance did not show significant improvements over DCell and the tree structure for scalability. The design Fat Tree. The major issue is the inability of the work to highlight scalability overhead instantiation minimized inter-connection cost management especially in smart grid applications.
Figure 23: BCDC Architecture [10]

**Generalized Hypercube and HSDC Architecture**

Figure 23 reveals a low-cost interconnection architecture that relies on a generic hypercube, while for greater incremental scalability, Figure 25 portrays a highly scalable data center network architecture. This design would require complex reductions to make it lightweight for smart grid applications, despite the current low-cost interconnection architecture called the Exchanged Generalized Hypercube (EGH) and the High Scalability DCN architecture.

Figure 24: Generalized Hypercube Architecture [11]

Figure 25: HSDC Architecture

**Stellar DCN Architecture**

In this section, the authors discussed a new generic construction for DCNs. The work presented a method of transforming the interconnection networks into potential dual-port server-centric DCNs. A description of the networking properties for stellar DCN such as routing algorithms, interconnection network, and the stellar transformation was discussed. The key aspects of the stellar construction are its topological simplicity, and its universal applicability.

**Summary of Related Works**

So far, this work has reviewed the generic DCNs that lacks the capacity to run microservices in Cloud at scale. Big data center analytics using open-source tools like Hadoop, Apache Storm, Kafka and NoSQL Cassandra was discussed in [6]. The authors [13], presented a packet-switched optical network (PSON) architecture with
centralized control for intra-data-center connectivity. Though the work classified traffic flows with different characteristics, light weighted computation was not addressed. This is because, scalability, low-latency, high-speed, and energy-efficient data center network remains very critical for BDSMC deployment in future large-scale data centers. In [20], the authors focused on hybrid-stream big data analytics model for to performing multimedia big data analysis. Their model contains four procedures, i.e., data pre-processing, data classification, data recognition and data load reduction. In this regard, Multi-dimensional Convolution Neural Network (CNN) was proposed and evaluated. In [21], DC traffic classification was investigated for time-sensitive and data-intensive service platforms. In [22], the authors proposed an edge processing unit that comprises two main parts: data classification model that classifies IoT data into maintenance-critical data (MCD) and maintenance-non-critical data (MNCD) and a data transmission unit that, based on the class of data, employs appropriate communication methods to transmit data to railway control centers. In [24], the work looked at the accurate estimation of data center resource utilization for multi-tenant co-hosted applications having dynamic and time-varying workloads. Their model adaptively and automatically identifies the most appropriate model to estimate DC resource utilization. The work in [25] proposed FlowSeer as a fast, low-overhead elephant flow detection and scheduling system using data stream mining. Their major idea is to leverage packets flows to train the streaming classification models for accurately and quick prediction the rate and duration of any initiated flow. In [27], the work proposed an online parameter-tuning method for the energy-efficient DCN named high-speed optical layer 1 switch system for time-slot-switching-based optical DCNs). The DCN comprises optical circuit switching network, optical slot switching network, and electrical packet switching network for the spine layer. Also, a procedure for reconfiguring flow classification function and a method for online parameter tuning classification was discussed. In [12], the work discussed the processing of big data streams generated by the Industrial Internet of Things (IIoT). It also looked at the edge and cloud DCs. The work then introduced the Network Elephants Learner and analYzer (NELLY) as a novel and efficient method for applying incremental learning at the server side of software-defined data center networks (SDDCNs). The idea is to accurately and timely identify elephant flows with low traffic overhead. The work in [12] proposed autoencoder (AE) network based on the distribution of polarimetric synthetic aperture radar (POL-SAR) data matrix, called a mixture autoencoder (MAE). Through a detailed analysis of the data distribution POL-SAR data matrix, a normalization method was presented. In terms of micro-services in the cloud domain, little efforts has been made as well. For instance, a representee sample of literature were captured in FacGraph Micro-service architecture in [22]. Computation offloading in [7], Containerisation in [3].

**CONCLUSION**

The development of DC network for efficient data stream offloading and Microservices in Cloud Computing Environments entails lots of efforts and contributions from data streams generation, to transmission, and storage for orchestration in the cloud.

**Gaps in literature for further study**

So far, existing works on DC network for efficient data stream offloading and Microservices orchestration within the cloud environments has some gaps. As a result of data collection with IoTs, stream computing, storage and DCN topologies present conspicuous gaps as observed from reviewed literature as follows:

i. Absence of optimal Internet of Things (IoT) node partitioning in Clusters thereby affecting Quality of Service (QoS) management.
ii. Network collapse resulting from point of failure topological layouts which affects workload scalability.

iii. Partial service accessibility as well as non-availability of recursive chain for sustained traffic propagation.

iv. Absence of Bayesian machine learning technique for QoS metric in BDMSC systems.

v. Existing works have not developed optimization schemes for data in motion (BDMSC).

vi. Little work has been done on connection availability model for stream generation.

vii. Most works have not explored the collision domain mapping of IoT clustered subnet work for QoS optimization.

viii. Most works lack optimal workload coordination for reduced resource drain.

ix. Existing approaches offers heavy traffic overhead, lower scalability, lower accuracy, and high detection time.

x. High performance Computing (HPC) workload managers lack micro-services support and deeply integrated container management, as opposed to container orchestrators (e.g. Kubernetes).

REFERENCES


