A Data Center Interconnects Calculus

Hao Wang, Haoyun Shen, Philipp Wieder, Ramin Yahyapour
GWDG / University of Göttingen, Germany

***Abstract*—For many application scenarios, interconnected data centers provide high service flexibility, reduce response time, and facilitate timely data backup. Many data center system parameters might have variant impact on the interconnection performance. Despite many studies on data center network performance, there exist few analytical work that reveal insightful knowledge with wide range of system parameters as input, especially focusing on data center interconnects (DCI). This paper creates analytical models for representative data center network architectures and provides the performance calculus aiming to apply for data center interconnects. By parameterising the number of devices, the arriving traffics, the switch link capacities, and the traffic locality, we derive the relationship among the DCI bandwidth, inter-DC latency, and these parameters. Based on this, further discussion and numerical examples investigate and evaluate the modelling and calculus from multiple angles and show the possibility how this calculus assists DC/DCI design and operation.**

I. Introduction

Large data centers (DC) are becoming the key infrastructure of Internet nowadays, where massive data are stored, forwarded, and processed by different business and private customers. However, the data centers might be centrally located far away from the customer, use the network only for transport, and be operated mostly by non-local companies. This leads to low flexibility, long latency, and risky data storage. Geographically distributing the compute and storage functions across DCs will facilitate more efficient utilization of resources, thus satisfy strict latency, capacity, and availability requirements. In addition, more and more findings of virtualization techniques and deployment of "cloud"-based infrastructures also fuel the growth of this trend. In this way, the borders between individual data centers are becoming soft. Through cross-layer data center interconnects (DCI), we face a big virtual computer and by optimally orchestrating resources, achieve high or customized quality of service.

One major DCI challenge is to broaden the interconnect bandwidth because of the tremendous growth in inter-DC traffics. The improvement lies in the breakthroughs of digital signal processing and optical transmission technologies. While these breakthroughs pave the way for further system designs, the tradeoff between the DCI bandwidth requirements according to realistic application scenarios and the usually high expenditure of new technologies is still a main design and operation concern. Furthermore, when the data center needs to be dynamically scaled up/out, lack of overview related to DCI would make the operation complicated and even more costly than necessary. However, the question like how much DCI bandwidth can satisfy what kind of traffic load sourced from hosts and out of data center is not yet adequately investigated in the literatures and previous works, in particular when we face diverse data center network (DCN) architectures and the traffics are quite random. More precisely, despite the individual DCI bandwidth requirements from diverse applications, data center operators and designers firstly need a systematic view related to DCI bandwidth over the whole infrastructure, since this guides building or renting DCI channels, assists DC re-architecture, and avoids potential complicated operations.

978-1-5386-2542-2/18/$31.00

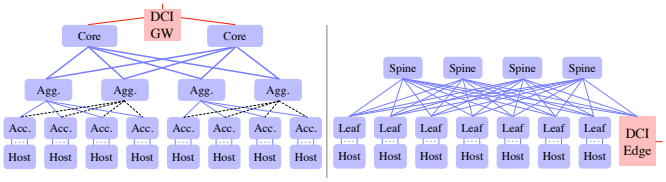
There exist many works that attempt to reveal the performance aspects as well as additional system properties of DCNs, few taking DCI into account. One set of these works follow the DCN architecture designs, e.g., three-tier Fat-Tree [3], [14], Leaf-Spine [1], DCell [16], BCube [15], etc. The authors of new designs usually provide analytical comparison with existing architectures, however, very roughly, and do measurements on a system prototype. The DC designers and operators who will establish their DCs according to these architecture designs can hardly follow the analytical principles therein as they are not customisable. And the analyses are architecture related, which is a lack of generality and restricts their scope of application onto the corresponding architectures. A further drawback, particularly from this paper's point of view, is the lack of DCI consideration. Another set of works are measurement based, that can almost accurately reflect the system reality [18], [4], [5], [8]. These works revealed many facts and insights of traffics within and inter DCs, established itself as a basis of many following designs and researches. While the drawbacks are also apparent, e.g., prior system deployment required, closely system scale and settings related. Moreover, these works mainly focus on traffics and few on relation between traffic and fabric capacity. Optimization based characterization can also expose the DC performance, e.g., [21]. Together with software-defined network (SDN) and virtualization techniques, this methodology mainly contributes to the resource orchestration with customized constraints, instead of trading off system parameters and performance metrics in particular when optimal goals do not exist. Optimization often relies on the underlying mean value assumption and observation of performance metrics, e.g., mean delay, mean rate, which however, can not very adequately reflect the capacity of systems. We frequently need to observe quantiles.

In this paper, we provide an analytical model of DCI to reveal the relationship between DC system settings and DCI bandwidth as well as latency, while avoiding aforementioned drawbacks. Technically, we apply stochastic network calculus [10], [7], [19], [17] to model the queueing and flow

demultiplexing elements in DCN. Accordingly, we show a calculus on the DCI flow bound firstly, in the way that we observe the bounds of host's arrival and link's transmission processes and derive the output flow bounds, iteratively until we obtain the bound of the flow to DCI channel. After that, we derive the stochastic delay bound in regard to this bound and DCI bandwidth. Stochastic flow and delay bounds observation widens the application scope of this model to non-Poisson traffics and beyond classical queueing analysis facilitates light-weighted DCN modelling with information like quantiles instead of mean value. The models presented in this paper applies directly for the three- and two-tier Fat-Tree architectures, also for the other architectures with iterative calculus application. We believe that our calculus with close-formed derivation will greatly benefit the DCI design and operation.

In network calculus, there exist works that model the in-tree architecture, particularly in the domain of sensor network [6]. This kind of work applies deterministic network calculus to derive the worst-case delay for the flow of interest. Since the arrival traffics are quite random, this calculus very possibly underestimates the system performance. A further limitation is that the arrivals are usually assumed to be bounded (or regulated) by token bucket, which restricts the application scope, or at least, the observation point of arrivals, i.e., the system's first service process is token bucket and we use its output as system arrivals. This calculus does not take into account the flow demultiplexing into the in-tree architecture, which is apparently the case in DCN. In [25], [13], stochastic network calculus is applied to derive the delay bounds of fork-join systems. Fork-join architecture consists of both demultiplexing (fork) and multiplexing (join) such that seems to be the fundamental structure of DCN, whereas its join synchronization is not applicable in DCI case, particularly when we need to observe the joined flow at the intermediate hops instead of the end hosts. Furthermore, observing the flow of interest weakens the application of these analytical results. Because given a DCN, i.e., given topology and link capacities, different flows of interests are sharing the network, such that the services they received are complementary, i.e., some flows receive increasing services, while the others might receive decreasing services, the viewpoints from individual flows can hardly converge to consensus and support overall design and operation decisions. [22], [24] applied stochastic network calculus to derive delay bounds considering the impact of cross flows. Therein, the cross flows share the services with the flow of interest, thus generate the leftover service. In that sense, the method can also be used to analyse DCN. But it is difficult to find a straightforward transformation to this paper as they focus on tandem network and the flow cross scenarios do not fit well into DCN.

The rest of this paper is structured as follows. In Section II we introduce the conventional DCN architectures as well as the related DCI options and merge them into one general representation. In Section III, we first introduce the fundamentals of network calculus, then apply them to commonly model



(a) Three-layer fat-tree (b) Leaf-spine

Figure 1. DC architectures and DCI options.

the general DCN architecture with DCI link. In Section IV, we provide flow bound calculus at different situations in this model, till we derive the bound of DCI flow and the stochastic delay that it will experience in DCI channel. Therefore, we establish the relationship among DCI bandwidth, delay, and system parameters. We numerically illustrate the results in Section V and conclude briefly in Section VI.

II. Data Center Network Architectures

Two data center network architectures are dominant. One is the conventional three-layer fat-tree, the other is leaf-spine architecture. Many other architectures evolve either based on these two or towards serving high performance computing. Figure 1 depicts these architectures with DCI options. DCI traffics flow in/out via either DCI gateway or an edge pod. The bandwidth in fat-tree grows bigger in the upper aggregate trunks, while the leaf-spine architecture abandons the overloaded upper trunks. The flat design of leaf-spine can therefore eliminate the cost of core switches and distribute the switching capacity to more spine switches with lower cost. On the other hand, circle detection process of spanning tree protocol chooses links only as backup. For example, with traffic flowing from an access switch, possibly dotted links shown in Figure 1.(a) will be blocked for the time being. Reducing layer will break circle, thus, avoid waste of switch ports. It is also easier to scale out for leaf-spine than fat tree. More and more emerging DCs adopt leaf-spine architecture, whereas there are still big amount of fat tree DCs running today. In this paper, we will lay our focus on both.

The anchor points to interconnect a fat tree or leaf-spine DC with remote ones are alternatively at layer 1, 2, 3. One option for fat tree DC is to connect via core switch/router and DCI gateway. Another option is to connect via dedicated edge pod and DCI gateway. See Figure 1. From the point of view of the DCI channel, whichever in (a) or (b), the traffics are aggregated to it, hence should require larger bandwidth. However, empirical findings [18], [4], [5], [8] and the forecast in [2] illustrate that only small part of traffics will go out of DCs. What bandwidth do we really need? This is the question we will answer in this paper.

Before modelling the DCN, we first merge above introduced DC architectures into one general representation with DCI considerations, which is depicted in Figure 2. This is basically the leaf-spine architecture wherein the DCI part is generalized. The DCN part consists of K spine switches denoted as S_k, where $1 \leq k \leq K$, and M leaf switches denoted as L_m, where $1 \leq m \leq M$. Each L_m connects to N_m hosts, where $N_m \geq 1$. We point out that, the three layer fat-tree can be easily obtained

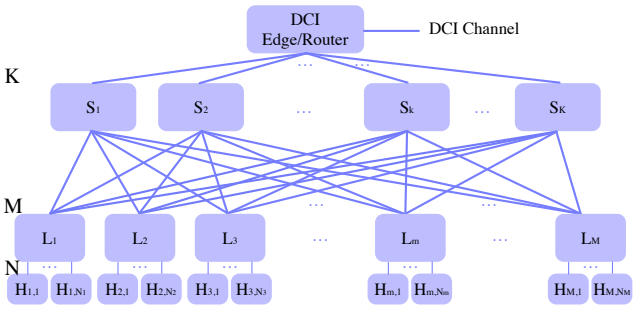


Figure 2. General DC architecture with DCI consideration.

from this architecture, by adding access switch layer and aggregate their upper links to individual aggregate switches, cluster by cluster. And we represent the leaf as aggregation, spine as core switches. From the viewpoint of DCI, these architectures exhibit certain similarity, especially when we move the DCI edge up to the spine switch layer in the leaf-spine architecture. All the inter-DC traffics aggregate to the top layer. So we have this three-layer architecture generalization for DCI. Further, multiple DCI channels are possibly used for redundancy or more bandwidth, either within one pod or in separated pods. The parallel transmission case is out of this paper's scope, other cases, are straightforward extensions of the following calculus.

III. Fundamentals of Modelling Data Center Network and Traffics in Network Calculus

Interconnecting DCs will improve the response time by moving processing local to the users and help backing up data in time for business continuity. In order to avoid that the DCI channel contributes the main part of the overall delay or mismatches the system scalability with inadequate bandwidth, we should characterize the traffics arriving at the DCI channel and derive the potential delay they will experience.

In this paper, we model data center elements still using components defined in queueing network theory, but mapping these components to the network calculus analysis [10], [7], [19], [17]. The class of tractable queueing network analysis has been consistently facing with the fundamental limitations imposed by assumptions like Poisson arrivals and exponential service time distribution. Network calculus circumvents some of the fundamental problems of the classical approach by using inequalities instead of equalities, namely considering bounds on traffic, service, delay, or buffer size, etc. Next, we first characterize the traffics, then model the transmission links in network calculus. Then, we decompose the flow transformation behaviours into separate components.

A. Network Calculus Fundamentals

Network traffic consists of data flows carried between two network nodes, which can be represented as point processes, i.e., bi-dimensional stochastic process $(t_n, s_n)_{n\geq 1}$, where t_n's denote the arrival times and s_n's denote the sizes of the flow's instantaneous arrivals, respectively. In this paper, we assume that the time model is discrete with events (e.g., flow data's

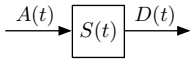


Figure 3. Dynamic server.

arrivals) occurring at time instants $t = 0, 1, 2, \ldots$ Thus, an arrival flow is described as a cumulative process $A(s,t)$ counting the number of data units (e.g., packets) arrived in the time interval $(s,t]$. The univariate representation is $A(t) = A(0,t)$, and the instantaneous arrival is $a(t) = A(t-1,t)$. A departure flow $D(t)$ is an arrival flow for the next node. In network calculus, we look for *bounds* on the flow instead of flow itself, such that a wider spectrum of traffic patterns can be covered. There are several forms of bounds, like bounding processes and envelopes [7], [20], [17], deterministic values on the long-term average rate and burstiness [10], [19], or bounds on the moments [7], [11], etc. We use moment generating function (MGF) bound in this paper to characterize the traffic bounding process or traffic itself, where the MGF of a random variable X, if exists, is denoted as $M_X(\theta) = E[e^{\theta X}]$ for some $\theta > 0$. Then the MGF bound (an upper bound) of arrivals is in form of $M_{A(s,t)}(\theta) \leq e^{\theta\rho(\theta)(t-s)+\theta\sigma(\theta)}$. It is also called $(\sigma(\theta), \rho(\theta))$ envelope [7] and can be denoted as $A \sim (\sigma(\theta), \rho(\theta))$. We use both denotation in this paper.

Queueing theory usually models a network link as a queueing system with a queue and a service process (mainly the transmission process). Network calculus substitutes the service process with a bounding process - *dynamic server* (see Figure 3), denoted by a bivariate random process $S(s,t)$, such that it satisfies the following relationship between departure process $D(t)$ and arrival process $A(t)$

$$D(t) \geq \inf_{0\leq s\leq t} \{A(s) + S(s,t)\} \ . \tag{1}$$

An example is the constant rate service $S(s,t) = C(t-s)$ with service rate $C > 0$, for which actually the equality holds. For DC network case, from the links' or aggregate flow's point of view, the transmission process belongs to this type, while from each flow's point of view, more general dynamic server applies. Because cross flows are sharing the deterministic rate link and the leftover service process might be quite dynamic due to the randomness of the cross flows. The dynamic server usually has a Laplace bound in the form of $M_{S(s,t)}(-\theta) \leq e^{-\theta\rho^S(-\theta)(t-s)-\theta\sigma^S(-\theta)}$, which for the work-conserving link case is $e^{-\theta C(t-s)}$, for some $\theta > 0$ and link capacity $C > 0$.

We model the flows fusion simply by accumulating the data from all the sub flows, i.e., the aggregation flow $A(t)$ of two flows $A_1(t)$ and $A_2(t)$ are defined as $A(t) = A_1(t) + A_2(t)$. Whereas the flow diffusion is not as straightforward as how fusion is defined. Which kind of system behaviour can be translated as diffusion? The flow demultiplexing from one leaf switch to multiple spine switches belong to this case. The traffic locality can also be a factor affecting the flow direction, e.g., some flows go across leaf-spine fabric, while the other may stay inside a rack. We model these as separate elements using the *scaling element* in network calculus [9], [27], [26], which (see Figure 4) consists of an arrival process $A(t)$, a scaling random process $\mathbf{X} = (X_i)_{i\geq 1}$ taking non-negative

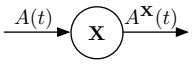


Figure 4. Scaling element.

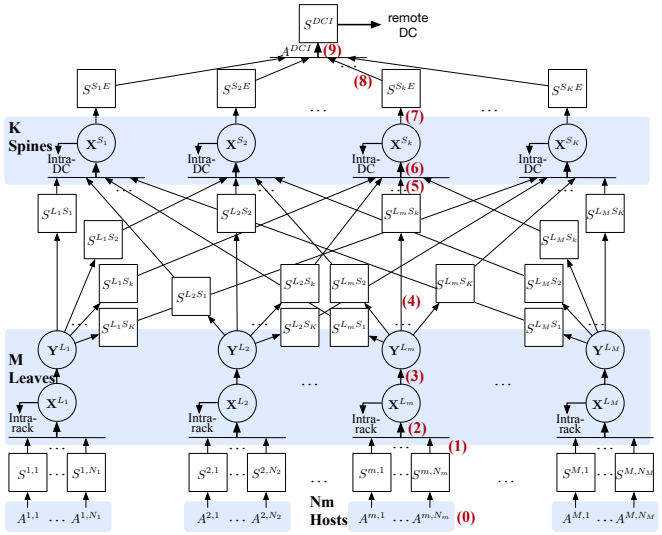


Figure 5. DC and DCI network calculus model.

integer values, and a scaled process $A^{\mathbf{X}}(t)$ defined for all $t \geq 0$ as

$$A^{\mathbf{X}}(t) = \sum_{i=1}^{A(t)} X_i \ . \tag{2}$$

Note, scaling element has no queue. In case the flows diffuse to multiple destinations, we use separate scaling element for each destination, such that we obtain a set of scaling elements. Assuming $n, n \geq 1$ destinations, we have scaling elements $\mathbf{X}^1, \cdots, \mathbf{X}^n$, such that $\mathbf{X} = \sum_{j=1}^{n} \mathbf{X}^j = \mathbf{1}$, where $\mathbf{1}$ is a process with values 1 indexed in space domain, and $\sum_{j=1}^{n} \mathbf{X}_i^j = 1$ for $i = 1, 2, \ldots$

B. Data Center Network Model

We apply the previously introduced network calculus elements to DCN, thus, model Figure 2 in Figure 5. In this figure, we simply use a horizontal line with multiple input flows to represent the flow aggregation. We use the dynamic server process of each link to represent the transmission process. S^{m,N_m} represents the link from host H_{m,N_m} to leaf switch L_m. $S^{L_m S_k}$ represents the link from leaf switch L_m to spine switch S_k. $S^{S_k E}$ represents the link from spine switch S_k to DCI edge. S^{DCI} represents the DCI channel. In addition, we assume that the sourcing flows are already aggregated to each host H_{m,N_m}, and denoted as A^{m,N_m}, for the ease of exposure. For more generality, we can repeatedly use flow aggregation component here. We denote flow locality, i.e., intra-rack, intra-/inter-DC, and the flow demultiplexing to different spine switches, as scaling elements $\mathbf{X}^{L_m}$, $\mathbf{X}^{S_k}$, and vector $\mathbf{Y}^{L_m}$ respectively. As a straightforward extension, scaling elements can also represent the flow demultiplexing to multiple DCI edge pods, if there exist. In this paper, we mainly focus on the case with one DCI pod.

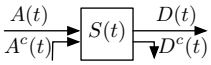


Figure 6. Dynamic server with flow of interest and cross flow.

IV. Data Center Interconnects Calculus

In this section, we first introduce several fundamental calculus on the traffics and performance indicators. Then, we derive the DCI flow bound and leverage the stochastic delay bound to guide DCI bandwidth choice.

A. Performance Calculus

In order to derive DCI flow bound, along the path from (0) to (9) in Figure 5, several explicit situations need to be dealt with. These include the bounds of flow a) aggregated by switches (position 2, 6, 9), b) regulated by the link transmission (1, 5, 8), c) transformed (filtered) by the scaling elements (3, 4, 7).

The aggregated flow has the following bound.

Lemma 1 (Bound of Aggregated Flow)**.** *Assume that flows $A_1(t), A_2(t), \ldots, A_N(t)$ respectively has MGF bound according to $(\sigma_1(\theta), \rho_1(\theta)), (\sigma_2(\theta), \rho_2(\theta)), \ldots, (\sigma_N(\theta), \rho_N(\theta))$, for any $\theta > 0$ and $N \geq 1$. The aggregated flow $A(t) = \sum_{k=1}^{N} A_k(t)$, has the following MGF bound:*

1) dependent flows,

$$M_{A(s,t)}(\theta) \leq e^{\theta \sum_{k=1}^{N} \rho_k(p_k\theta)(t-s) + \theta \sum_{k=1}^{N} \sigma_k(p_k\theta)} \ , \tag{3}$$

where $\sum_{k=1}^{N} \frac{1}{p_k} = 1, p_k > 1$ for any k.

2) independent flows,

$$M_{A(s,t)}(\theta) \leq e^{\theta \sum_{k=1}^{N} \rho_k(\theta)(t-s) + \theta \sum_{k=1}^{N} \sigma_k(\theta)} \ . \tag{4}$$

For the ease of exposure, we use the same θ for each bound. They can be different. The proof follows directly by using the definition of flow aggregation and Hölder's inequality.

Next lemma computes the output bound of a flow that traversed a link.

Lemma 2 (Output Bound)**.** *A flow of interest $A(t)$ traverses a work-conserving link with dynamic server process $S(t) = Ct$, sharing the server with a cross flow $A^c(t)$ without priority knowledge (blind multiplexing). See Figure 6. The flows and the service are independent. Assuming that the flows have MGF bounds according to $(\sigma(\theta), \rho(\theta))$ respectively $(\sigma^c(\theta), \rho^c(\theta))$, we get the following bound on the output $D(t)$*

$$M_{D(s,t)}(\theta) \leq e^{\theta\rho(\theta)(t-s)} \frac{e^{\theta(\sigma(\theta)+\sigma^c(\theta))}}{1 - e^{\theta(\rho(\theta)-(C-\rho^c(\theta)))}} \ , \tag{5}$$

under the stability condition $\rho(\theta) + \rho^c(\theta) - C < 0$.

Proof. The leftover dynamic server for flow $A(t)$ is $S^{LO}(s,t) = [S(s,t) - A^c(s,t)]^+$ in case of blind multiplexing [12]. Then for all $0 \leq s \leq t$, we have

$$\begin{aligned} D(s,t) &\leq A(t) - D(s) \\ &\leq A(t) - \inf_{0 \leq u \leq s} \{A(u) + S^{LO}(u,s)\} \\ &\leq \sup_{0 \leq u \leq s} \{A(u,t) - S(u,s) + A^c(u,s)\} \ . \end{aligned}$$

In the second line, we used the definition of dynamic server. Then, we have

$$M_{D(s,t)}(\theta) \leq \sum_{0\leq u\leq s} E\left[e^{\theta(A(u,t)+A^c(u,s)-S(u,s))}\right]$$
$$\leq e^{\theta\rho(\theta)(t-s)}e^{\theta(\sigma(\theta)+\sigma^c(\theta))}\sum_{0\leq u\leq s} e^{\theta(\rho(\theta)-(C-\rho^c(\theta)))(s-u)} .$$

In the first line, we used Union bound. In the second line, we used the assumptions. Then, using stability condition and letting $s\to\infty$ complete the proof. □

If there is no cross flow, the result holds letting $\rho^c(\theta) = \sigma^c(\theta) = 0$. Without satisfying the stability condition, the output flow will be unbounded. The lemma can be easily extended to the case with more general dynamic server.

Next lemma shows the bound on the traffics after traversing a scaling element.

Lemma 3 (MGF Bound of Scaled Flow)**.** *A flow $A(t)$ traverses an independent scaling element $\mathbf{X}$ and generates the scaled flow $A^{\mathbf{X}}(t)$. The scaled flow has MGF bound*

$$M_{A^{\mathbf{X}}(t)}(\theta) = M_{A(t)}(\log M_{\mathbf{X}}(\theta)) ,$$

if $\mathbf{X}$ is i.i.d.

The proof for the uni- / bi-variate formulation follows by using conditional expectation. For a more general scaling process, e.g., a Markov-modulated process, see [9]. This lemma can be applied recursively to derive the MGF bound of a scaled flow through a series of scaling elements. For instance, if $\mathbf{X}$ and $\mathbf{Y}$ are *i.i.d.* then

$$M_{(A^{\mathbf{X}})^{\mathbf{Y}}(t)}(\theta) = M_{A^{\mathbf{X}}(t)}(\log M_{\mathbf{Y}}(\theta)) = M_{A(t)}(\log M_{\mathbf{X}}(\log M_{\mathbf{Y}}(\theta))) . \quad (6)$$

Next, we derive the *stochastic delay bound*. Before that, we define the *delay process* as $W(t) = \inf\{d : A(t-d) \leq D(t)\}$. Implicitly, we assume first in first out (FIFO) scheduling when we observe the arrival and departure data units. Assume the arrivals independent of the service. Given the MGF bound on arrivals and Laplace bound on service regarding to $(\sigma(\theta), \rho(\theta))$ respectively $(0, -\theta C)$, we derive the stochastic delay bound d with violation probability ε^W as below. First,

$$Pr(W(t)\geq d) \leq Pr\left(\sup_{0\leq s<t}\{A(s,t) - S(s,t+d)\geq 0\}\right) \leq \frac{e^{-\theta C d}e^{\theta\sigma(\theta)}}{1-e^{-\theta(C-\rho(\theta))}} := \varepsilon^W .$$

By turn we used the definitions of delay process, dynamic server, the Union bound, Chernoff's inequality, and the assumptions. Then

$$d = \inf_{\theta>0}\left\{\frac{\theta\sigma(\theta) - \log(\varepsilon^W\theta(C-\rho(\theta)))}{\theta C}\right\} . \quad (7)$$

Similarly, defining the *backlog process* as $B(t) = A(t) - D(t)$, we get the *stochastic backlog bound* b for the violation probability ε^B as

$$b = \inf_{\theta>0}\left\{\frac{\theta\sigma(\theta) - \log(\varepsilon^B\theta(C-\rho(\theta)))}{\theta}\right\} . \quad (8)$$

Interestingly, letting $\varepsilon^W = \varepsilon^B$ implies $b = Cd$.

B. Inter-DC Flow Bound

Now, we apply the results introduced above to derive the DCI flow bound.

Theorem 1 (DCI Flow Bound)**.** *Consider the network model from Figure 5, where arrival processes at hosts to leaf m links, i.e., $A^{m,1}(t),\ldots,A^{m,N_m}(t)$, for all $1\leq m\leq M$ and host number $N_m\geq 1$, traverse the data center network with service and scaling elements that, along any path from host to DCI channel, are (mutually) independent. Assume the MGF bounds $M_{A^{m,n_m}(s,t)}(\theta)\leq e^{\theta\rho^{m,n_m}(t-s)+\theta\sigma^{m,n_m}(\theta)}$, for $1\leq n_m\leq N_m$ and some $\theta>0$. Assume the Laplace bounds $M_{S^{m,N_m}(s,t)}(-\theta)\leq e^{-\theta C^{m,n_m}(t-s)}$, $M_{S^{L_mS_k}(s,t)}(-\theta)\leq e^{-\theta C^{L_mS_k}(t-s)}$, $M_{S^{S_kE}(s,t)}(-\theta)\leq e^{-\theta C^{S_kE}(t-s)}$ for some $\theta>0$ where $1\leq k\leq K$. Moreover, denote the scaling element from leaf switch L_m to spine switch S_k as $\mathbf{Y}^{L_mS_k}$, i.e., $\mathbf{Y}^{L_m}=\sum_{k=1}^K\mathbf{Y}^{L_mS_k}=\mathbf{1}$, and assume that all $\mathbf{X}^{L_m},\mathbf{Y}^{L_mS_k},\mathbf{X}^{S_k}$ are i.i.d. processes and independent of the service elements. Under stability conditions, to be explicitly given in the proof, we have the following inter-DC arrival flow bound*

$$M_{A^{DCI}(s,t)}(\theta)\leq e^{\theta\rho(\theta)+\theta\sigma(\theta)} ,$$

where

$$\rho(\theta)=\sum_{k=1}^K\frac{1}{\theta_k}\left(\sum_{m=1}^M\left(a^{m,k}(\theta_k)\sum_{n_m=1}^{N_m}\rho^{m,n_m}\left(a^{m,k}(\theta_k)\right)\right)\right),$$
$$\sigma(\theta)=\sum_{k=1}^K\frac{1}{\theta_k}\left(\sum_{m=1}^M\left(a^{m,k}(\theta_k)\sum_{n_m=1}^{N_m}\left(\sigma^{m,n_m}(a^{m,k}(\theta_k))-\log\left(1-e^{a^{m,k}(\theta_k)(\rho^{m,n_m}(a^{m,k}(\theta_k))-C^{m,n_m})}\right)\right)-\right.\right.$$
$$\left.\log\left(1-e^{a^{m,k}(\theta_k)\sum_{n_m=1}^{N_m}\rho^{m,n_m}(a^{m,k}(\theta_k))-a^{\mathbf{X}^{S_k}}(\theta_k)C^{L_mS_k}}\right)\right)$$
$$\left.-\log\left(1-e^{\sum_{m=1}^M a^{m,k}(\theta_k)\sum_{n_m=1}^{N_m}\rho^{m,n_m}(a^{m,k}(\theta_k))-\theta_kC^{S_kE}}\right)\right),$$

denoting for brevity that

$$\theta_k{:=}p_k\theta \text{ for } k=1,2,\ldots,K,\ \sum_{k=1}^K\frac{1}{p_k}=1,\ p_k>1$$
$$a^{\mathbf{X}^{L_m}}(\theta){:=}\log M_{\mathbf{X}^{L_m}}(\theta)$$
$$a^{\mathbf{Y}^{L_mS_k}}(\theta){:=}\log M_{\mathbf{Y}^{L_mS_k}}(\theta)$$
$$a^{\mathbf{X}^{S_k}}(\theta){:=}\log M_{\mathbf{X}^{S_k}}(\theta)$$
$$a^{m,k}(\theta){:=}a^{\mathbf{X}^{L_m}}\left(a^{\mathbf{Y}^{L_mS_k}}\left(a^{\mathbf{X}^{S_k}}(\theta)\right)\right) .$$

Proof. The derivation follows the steps from (0) to (9) in Figure 5, i.e., flows source from the hosts and aggregate to DCI channel. Step (0) represents the inputs, i.e., given assumptions on arrival flows to the hosts. At (1), applying Lemma 2 (output bound) we get the MGF bound of departure flow $D^{m,n_m}(t)$

$$M_{D^{m,n_m}(s,t)}(\theta)\leq e^{\theta\rho^{m,n_m}(\theta)(t-s)+\theta\sigma_D^{m,n_m}(\theta)} ,$$

where

$$\sigma_D^{m,n_m}(\theta) := \frac{1}{\theta}\Big(\sigma^{m,n_m}(\theta) - \log\Big(1 - e^{\theta(\rho^{m,n_m}(\theta) - C^{m,n_m})}\Big)\Big)$$

and under the stability condition for some $\theta > 0$

$$stab1 : \theta\rho^{m,n_m}(\theta) - \theta C^{m,n_m} < 0 \ .$$

Then, using Eq. (4) in Lemma 1, the aggregated flow of all this kind of flows to leaf switch m at (2) has the following MGF bound

$$M_{D^m(s,t)}(\theta) \leq e^{\theta\sum_{n_m=1}^{N_m}\rho^{m,n_m}(\theta)(t-s)+\theta\sum_{n_m=1}^{N_m}\sigma_D^{m,n_m}(\theta)} \ .$$

At (3), the flow is divided by the scaling element $\mathbf{X}^{L_m}$ as some parts of the flow stay inside the rack. Although different flow sources may intrinsically exhibit diverse locality, aggregating them will smooth the variation in a smaller granularity [23], e.g., observing from packet instead of flow level. So we assume $\mathbf{X}^{L_m}$ *i.i.d.* Afterwards, the egressed flow from a leaf (rack here) is demultiplexed to different spine switches by scaling elements $\mathbf{Y}^{L_m S_k}, 1 \leq k \leq K$. See step (4). Again, we assume that each demultiplexing branch applies an *i.i.d.* scaling process, because of flow aggregation. A further reason is that load balancing also enhances the variation smoothing effect and additionally reduce the dependency. Nevertheless, scaling processes other than *i.i.d.* ones are still applicable, e.g., Markov-modulated processes, which can be found in [9]. For (3) and (4), we recursively apply Lemma 3 as shown in Eq. (6) and have the MGF bound of the flow at (4)

$$M_{((D^m)^{\mathbf{X}^{L_m}})^{\mathbf{Y}^{\mathbf{L_m S_k}}}(s,t)}(\theta) = M_{D^m(s,t)}\big(a^{\mathbf{X}^{\mathbf{L_m}}}\big(a^{\mathbf{Y}^{L_m S_k}}(\theta)\big)\big) \ .$$

(5) and (6) are similar to (1) and (2). Denoting the flow at (6) as D^k and using Lemma 2, 1, we have its MGF bound

$$\begin{aligned} M_{D^k(s,t)}(\theta) \quad \leq \quad & e^{\Bigg[\theta\Bigg(\frac{1}{\theta}\sum_{m=1}^{M}\Big(a^{\mathbf{X}^{\mathbf{L_m}}}\big(a^{\mathbf{Y}^{L_m S_k}}(\theta)\big)\cdot \\ & \sum_{n_m=1}^{N_m}\rho^{m,n_m}\big(a^{\mathbf{X}^{\mathbf{L_m}}}\big(a^{\mathbf{Y}^{L_m S_k}}(\theta)\big)\big)\Big)\Bigg)(t-s)+ \\ & \theta\Bigg(\frac{1}{\theta}\sum_{m=1}^{M}\Big[a^{\mathbf{X}^{\mathbf{L_m}}}\big(a^{\mathbf{Y}^{L_m S_k}}(\theta)\big)\sum_{n_m=1}^{N_m}\sigma_D^{m,n_m}\big(a^{\mathbf{X}^{\mathbf{L_m}}}\big(a^{\mathbf{Y}^{L_m S_k}}(\theta)\big)\big) \\ & - \log\Big(1-e^{\Big(a^{\mathbf{X}^{\mathbf{L_m}}}\left(a^{\mathbf{Y}^{L_m S_k}}(\theta)\right)\sum_{n_m=1}^{N_m}\rho^{m,n_m}\left(a^{\mathbf{X}^{\mathbf{L_m}}}\left(a^{\mathbf{Y}^{L_m S_k}}(\theta)\right)\right) \\ & \qquad -\theta C^{L_m S_k}\Big)}\Big)\Big]\Bigg)\Bigg] \ , \end{aligned}$$

under a further stability condition

$$\begin{aligned} stab2 : a^{\mathbf{X}^{\mathbf{L_m}}}\big(a^{\mathbf{Y}^{L_m S_k}}(\theta)\big)\sum_{n_m=1}^{N_m}\rho^{m,n_m}\big(a^{\mathbf{X}^{\mathbf{L_m}}}\big(a^{\mathbf{Y}^{L_m S_k}}(\theta)\big)\big) \\ -\theta C^{L_m S_k} < 0 \ . \end{aligned}$$

The packets in the aggregated flow at (6) can either go out of this DC (north-south traffic) or stay within (east-west traffic). We adopt the same arguement as for the rack and demultiplexing scalings to assume $\mathbf{X}^{S_k}$ *i.i.d.* Hence, we derive the MGF bound for (7)

$$M_{(D^k)^{\mathbf{X}^{S_k}}(s,t)}(\theta) = M_{D^k(s,t)}\big(a^{\mathbf{X}^{S_k}}(\theta)\big) \ .$$

Again, using Lemma 2, the output flow at (8), denoted as $D^{S_k E}(t)$, has the following MGF bound

$$\begin{aligned} M_{D^{S_k E}(s,t)}(\theta) \quad \leq \quad & e^{\Bigg[\theta\Big(\frac{1}{\theta}\sum_{m=1}^{M}a^{m,k}(\theta)\sum_{n_m=1}^{N_m}\rho^{m,n_m}\big(a^{m,k}(\theta)\big)\Big)(t-s)+ \\ & \theta\Bigg(\frac{1}{\theta}\sum_{m=1}^{M}\Big[a^{m,k}(\theta)\sum_{n_m=1}^{N_m}\sigma_D^{m,n_m}\big(a^{m,k}(\theta)\big)- \\ & \log\Big(1-e^{\Big(a^{m,k}(\theta)\sum_{n_m=1}^{N_m}\rho^{m,n_m}\left(a^{m,k}(\theta)\right)-a^{\mathbf{X}^{S_k}(\theta)C^{L_m S_k}}\Big)}\Big)\Big]- \\ & \log\Big(1-e^{\Big(\sum_{m=1}^{M}a^{m,k}(\theta)\sum_{n_m=1}^{N_m}\rho^{m,n_m}\left(a^{m,k}(\theta)\right)-\theta C^{S_k E}\Big)}\Big)\Bigg)\Bigg] \ , \end{aligned}$$

under stability condition

$$stab3 : \sum_{m=1}^{M}a^{m,k}(\theta)\sum_{n_m=1}^{N_m}\rho^{m,n_m}\big(a^{m,k}(\theta)\big) - \theta C^{S_k E} < 0 \ .$$

At (9), the input flows might be dependent, since they are possibly splitted at one leaf switch and merged again at the DCI edge. So, we complete the proof by applying Eq. (3) in Lemma 1 for $\sum_{k=1}^{K}\frac{1}{p_k} = 1, p_k > 1$, under stability conditions $stab1, 2, 3$. □

This calculus mainly serves the leaf-spine architecture shown in Figure 5, while it is straightforward to enlarge the scope of it to an architecture with more incremental layers, by using aforementioned lemmas. Next, we extend the architecture with one more access switch layer (see Figure 7 for illustration), and derive the inter-DC flow bound. Therein, the arrival processes $A^{m,n_m}(t)$ at the leaf/aggregate switch m are in fact departures from the n_m-th access switch connected with it. These departures are obtained by aggregating and scaling the departures from the individual host-to-access switch links. We denote the arrival processes from hosts in the new architecture as $A^{m,n_m,j}(t)$ for all $1 \leq j \leq J^{n_m}, 1 \leq m \leq M, 1 \leq n_m \leq N_m$, and the dynamic server of link from host (m, n_m, j) to n_m-th access switch of leaf/aggregate switch m as $S^{m,n_m,j}(t)$. Again, M, N_m, J^{n_m} for m and n_m are all given numbers according to the system settings. This extension fits well the conventional three layer fat-tree architecture. Next proposition shows the corresponding DCI flow bound. For more layers, we can clearly see the iteration from the result.

Proposition 1 (DCI Flow Bound with access layer)**.** *Consider the extension of Figure 5 with Figure 7, besides the assumptions in Theorem 1, we assume the MGF bounds* $M_{A^{m,n_m,j}(s,t)}(\theta) \leq e^{\theta\rho^{m,n_m,j}(\theta)(t-s)+\theta\sigma^{m,n_m,j}(\theta)}$ *for all* m, n_m, j *described above and some* $\theta > 0$*, the Laplace bounds* $M_{S^{m,n_m,j}(s,t)}(-\theta) \leq e^{-\theta C^{m,n_m,j}(t-s)}$*. Denote the i.i.d. scaling elements from the access switch to leaf/aggregate*

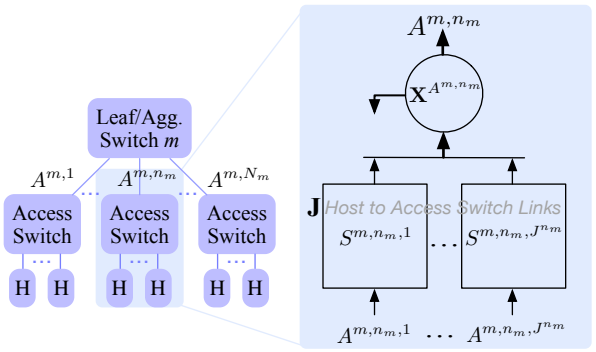


Figure 7. DC and DCI network calculus with extension to three layer architecture.

switch as $\mathbf{X}^{A^{m,n_m}}$ *and assume them independent of the dynamic servers. We have the following DCI flow bound*

$$M_{A^{DCI}(s,t)}(\theta) \leq e^{\theta\rho(\theta)+\theta\sigma(\theta)} ,$$

where

$$\rho(\theta) = \sum_{k=1}^{K} \frac{1}{\theta_k} \left(\sum_{m=1}^{M} \sum_{n_m=1}^{N_m} \sum_{j=1}^{J^{n_m}} a\rho^{m,n_m}(a) \right) ,$$

$$\sigma(\theta) = \sum_{k=1}^{K} \frac{1}{\theta_k} \Bigg(\sum_{m=1}^{M} \Bigg(\sum_{n_m=1}^{N_m} \Bigg(\sum_{j=1}^{J^{n_m}} \Big(a\sigma^{m,n_m,j}(a) - \log\left(1 - e^{a\rho^{m,n_m,j}(a) - aC^{m,n_m,j}}\right)\Big) - \log\left(1 - e^{\sum_{j=1}^{J^{n_m}} a\rho^{m,n_m,j}(a) - a^{m,k}(\theta_k)C^{m,n_m}}\right)\Bigg) - \log\left(1 - e^{\sum_{n_m=1}^{N_m}\sum_{j=1}^{J^{n_m}} a\rho^{m,n_m,j}(a) - a^{\mathbf{X}^{S_k}}(\theta_k)C^{L_m S_k}}\right)\Bigg) - \log\left(1 - e^{\sum_{m=1}^{M}\sum_{n_m=1}^{N_m}\sum_{j=1}^{J^{n_m}} a\rho^{m,n_m,j}(a) - \theta_k C^{S_k E}}\right)\Bigg) ,$$

denoting for brevity that

$$a := a^{\mathbf{X}^{A^{m,n_m}}}\left(a^{\mathbf{X}^{L_m}}\left(a^{\mathbf{Y}^{L_m S_k}}\left(a^{\mathbf{X}^{S_k}}(\theta_k)\right)\right)\right) ,$$

under the stability conditions

$$stab1: a\rho^{m,n_m,j}(a) - aC^{m,n_m,j} < 0$$

$$stab2: \sum_{j=1}^{J^{n_m}} a\rho^{m,n_m,j}(a) - a^{m,k}(\theta_k)C^{m,n_m} < 0$$

$$stab3: \sum_{n_m=1}^{N_m}\sum_{j=1}^{J^{n_m}} a\rho^{m,n_m,j}(a) - a^{\mathbf{X}^{S_k}}(\theta_k)C^{L_m S_k} < 0$$

$$stab4: \sum_{m=1}^{M}\sum_{n_m=1}^{N_m}\sum_{j=1}^{J^{n_m}} a\rho^{m,n_m,j}(a) - \theta_k C^{S_k E} < 0 .$$

The proof directly follows by using Lemma 1, 2, 3, and Theorem 1. We point out that the iteration lies in the accumulation of rates and burstiness components across layers.

C. Discussion on Inter-DC Flow Bound Calculus

Approximation. The application of Hölder's inequality facilitates the mathematical derivation, particularly separating dependent *r.v.*'s. However, such a "black-box" way may lose the gain of knowing dependency information, thus, worsen the output bound. We introduce an approximative analysis. Because the flows are splitted at leaf switches and merged again at the DCI edge, the flows of step (8) are dependent. In fact, this dependency exhibits "quasi" negative correlationship, since given an input flow $(A(t))$, increasing the amount demultiplexed to one destination $(A_1(t))$ will decrease the amount to the other destination $(A_2(t))$, where $A(t) = A_1(t) + A_2(t)$. We can thus approximately use the inequality like $E[e^{\theta(A_1(t)+A_2(t))}] \leq E[e^{\theta A_1(t)}]E[e^{\theta A_2(t)}]$ instead of Hölder's inequality at (9). The $(\sigma(\theta), \rho(\theta))$ in Theorem 1 is simplified as $\theta_k := \theta$ instead of $p_k\theta$.

Node-by-node vs. end-to-end. So far, all the derivations are based on the node by node analysis, i.e., derive the output bounds and if necessary, aggregate them at each step from (1) to (9). In fact, when we focus on any single source flow, e.g., $A^{m,1}$, parts of it reach (8) through one path. It is then possible to apply the end-to-end analysis introduced in [12] (handling cross traffic) and [9] (flow transformation) to derive the output bound at (8) for flow of interest and aggregate all these bounds once. However, the "pay-burst-only-once" (PBOO) [19] property that in principle leads to better analytical results does not apply for the output flow bound, particularly if we use the method introduced in [11]. Consider a flow $A(t)$ traverses two concatenated nodes with dynamic server $S_1(t)$ and $S_2(t)$. The processes are independent. The concatenation dynamic server for this simple network is in a convolution form $S(s,t) = \inf_{s\leq u\leq t}\{S_1(s,u) + S_2(u,t)\}$ for $0 \leq s \leq t$. Assume MGF bound $M_{A(s,t)}(\theta) \leq e^{\theta\rho(\theta)+\theta\sigma(\theta)}$ and Laplace bound $M_{S_i(s,t)}(-\theta) \leq e^{-\theta C_i(t-s)}$ for $i = 1, 2$. The node by node analysis generates a bound for the departure flow $D(t)$

$$M_{D(s,t)}(\theta) \leq e^{\theta\rho(\theta)(t-s)} e^{\theta\sigma(\theta)} \frac{1}{1 - e^{\theta\rho(\theta)-\theta C_1}} \frac{1}{1 - e^{\theta\rho(\theta)-\theta C_2}} ,$$

while the end-to-end analysis (see [11]) generates

$$M_{D(s,t)}(\theta) \leq e^{\theta\rho(\theta)(t-s)} e^{\theta\sigma(\theta)} \frac{1}{\left(1 - e^{\theta\rho(\theta)+\log b}\right)^2} ,$$

if let $b = \sup\{e^{-\theta C_1}, e^{-\theta C_2}\}$. The end-to-end calculus generates worse bound.

D. Application of the Calculus - Trading Off DCI Bandwidth and System Parameters

The proposed calculus can help to choose suitable DCI bandwidth, either in the design and deployment phases or the duration of operation. The criteria can be quite diverse and flexible. As we can see in the previous modelling, there are many parameters that can quantify the system and trade off each other. Data center designer and operator can accordingly customize these parameters as input and output under the condition that certain QoS/QoE requirements or business agreements are achievable. See Table I for the summary of the

Parameter	Description
K	number of spine/core switches
M	number of leaf/aggregate switches
N_m	number of hosts connected to leaf switch m, resp. number of access switches to aggregate switch m
J^{n_m}	number of hosts connected to n_m-th access switch of aggregate switch m
C^{DCI}	link capacity of inter-DC channel
C^{S_kE}	link capacity from spine/core switch k to DCI edge
$C^{L_mS_k}$	link capacity from leaf switch m to spine switch k, resp. from aggregate switch m to core switch k
C^{m,n_m}	link capacity from host H_{m,n_m} to leaf switch m, resp. from n_m-th access switch to aggregate switch m
$C^{m,n_m,j}$	link capacity from host $H_{m,n_m,j}$ to n_m-th access of aggregate switch m
$\mathbf{X}^{S_k}$	out-of-DC decision process at spine resp. core switch k
$\mathbf{Y}^{L_m}$	demultiplexing processes from leaf switch m to spines, resp. from aggregate switch m to cores
$\mathbf{X}^{L_m}$	egress decision process at leaf/aggregate switch m
$\mathbf{X}^{A^{m,n_m}}$	egress decision process at n_m-th access switch of aggregate switch m
A^{m,n_m}	arrivals of the link from host n_m to leaf switch m
$A^{m,n_m,j}$	arrivals of the link from host j to access switch n_m of aggregate switch m
$(d,\ \varepsilon^W)$	stochastic delay bound with violation probability
$(b,\ \varepsilon^B)$	stochastic backlog bound with violation probability

Table I
DCN PARAMETERS.

parameters used in leaf-spine respectively three layer fat-tree architecture. Small letters represent indexes. Heterogeneous arrivals and system settings imply a bigger set of parameters, while homogeneous systems hold a reduced set. $\mathbf{X}$, $\mathbf{Y}$ and A are all random processes, therefore, the detailed parameters, e.g., rate, burstiness, etc., depend on the process characteristics and realistic statistics.

Delay is a very important performance metric. Several delay related performance indicators can be used as criteria, e.g., mean delay, worst-case delay, or delay quantiles. Pursuing exact delay quantiles might be difficult, instead, we use tail delay bound. Denote the link capacity of inter-DC channel as $S^{DCI}(t)$. We can assume it as a work-conserving link, i.e., $S^{DCI}(t) = C^{DCI}t$. Eq. (7) and Theorem 1 produce a delay bound d^{DCI,ε^W} with violation probability ε^W. Under a requirement that the delay at time t exceeds the bound d^{DCI,ε^W} with probability ε^W, we can calculate a suitable C^{DCI}, if all other parameters are also given. Or, if regarding C^{DCI} as input, we can obtain the stochastic delay bound for any violation probability ε^W. In parallel, from Eq. (8), we can derive the relation of strochastic backlog bound and system parameters. In addition, every link will contribute to the overall end-to-end delay. Harmonising the DCI delay with all the link delay will be another option to optimize the DCI bandwidth. For instance, let DCI delay bound with violation probability ε as a function of maximal delay bound with ε of all DCN links.

Other important aspects that can be reflected in the calculus are scalability and architecture variation of DCs that support interconnects. Usually, a data center supports multi-tenancy. Each tenant may generate many flows, with longer or shorter duration, or varying number of parallel flows at one time. Aggregating these flows across a long term and model-fitting

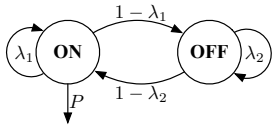


Figure 8. Markov-Modulated On-Off (MMOO) arrival process.

it is a way to characterize the flow process for individual tenant. An alternate is to apply Lemma 1 to all the flows that belong to this tenant. Each tenant may contribute one sub-flow, already aggregated for itself, to each A^{m,n_m}. Recursively using Lemma 1 for the number of tenants that reside in the host (m, n_m) will tie up this tenant scalability, traffic intensity, and DCN capacity in accordance with above calculus under customized requirements. On the other hand, numbers $K,\ M,\ N_m$, for $1 \le m \le M$, shape the DCN architecture. Together with link capacities, these numbers further shows the overall DCN capacity in an implicit way. In the calculus of stochastic output, delay and backlog bounds, we can reveal the impact of these numbers over DCI bandwidth, and potentially, further design and operation goals.

V. NUMERICAL EVALUATION AND ILLUSTRATION

In this section we numerically calculate the stochastic delay bounds of DCI channel, while drawing insights on how DCI bandwidth, as well as those parameters contained in its arrival flow bound presented in Theorem 1 and Proposition 1 affect the stochastic delay bounds. Subsequently, we are able to obtain insightful knowledge for the benefit of DC design and operation regarding to DCI requirements.

In order to compute stochastic delay bound of DCI channel, we use Eq. (7) and Theorem 1. We consider the data center network scenario from Figure 5 with two example arrival processes: Poisson with rate λ and Markov-Modulated On-Off (MMOO). The former matches the high granularity user events; the latter matches different data granularities in particular with varying burstiness. The MMOO process is represented in Figure 8 in terms of the transition probabilities λ_1 and λ_2 , and also the peak-rate P, i.e., the process transmits at rate P while in state 'on' and is idle while in state 'off'. For MMOO process $A(t)$ we have MGF bound $E\left[e^{\theta A(t)}\right] \le e^{\theta\rho^A(\theta)t+\theta\sigma^A(\theta)}$, where $\rho^A(\theta) = \frac{1}{\theta}\log\frac{\lambda_1 e^{\theta P}+\lambda_2+\sqrt{(\lambda_1 e^{\theta P}+\lambda_2)^2-4(\lambda_1+\lambda_2-1)e^{\theta P}}}{2}$ and $\sigma^A(\theta) = 0$. As discussed, we apply Bernoulli process with parameter p for the scaling elements $\mathbf{X}$'s, $\mathbf{Y}$. So, their MGFs are in form of $1 - p + pe^{\theta}$ for some $\theta > 0$ with different p.

In particular, for the convenience of illustration we consider a homogeneous system setting in the numerical evaluation, i.e., symmetric architecture, evenly assigned switch link capacities and load balancing, same arrival parameters. In that sense, we simplify the result in Theorem 1 and Eq. (7) through introducing new notations, i.e., $N_m = N$, $\rho^{m,n_m} = \rho^A$, $\sigma^{m,n_m} = \sigma^A$, $C^{m,n_m} = C^{HL}$, $C^{L_mS_k} = C^{LS}$, $C^{S_kE} = C^{SE}$, $\mathbf{X}^{L_m} = \mathbf{X}^L$, $\mathbf{Y}^{L_mS_k} = \mathbf{Y}^L$, $\mathbf{X}^{S_k} = \mathbf{X}^S$. For the heterogeneity case, the ways of parameterising every arrival, link capacity, and the number of devices in our calculus will inherently allow us to input unbalanced arrival load or set up biased architecture.

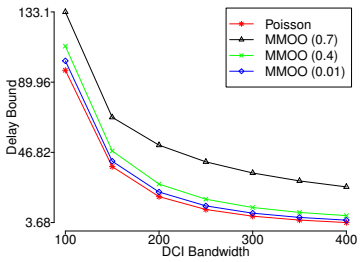


Figure 9. Delay bounds of DCI channel changing bandwidth.

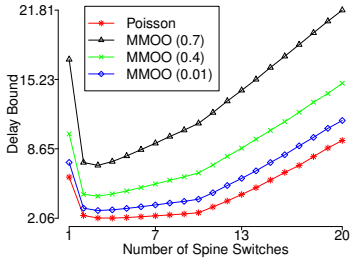


Figure 10. Delay bounds of DCI channel changing number of spine switches.

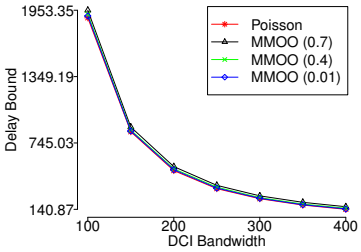


Figure 11. Delay bounds of DCI channel changing bandwidth for larger scale DC.

First, we illustrate the relation between the DCI bandwidth and the stochastic delay bound. In turn, we observe the influence of arrival burstiness from the hosts onto the stochastic delay bounds at DCI channel. We use the following numerical settings. The number of switches and hosts are $K = 12, M = 20, N = 36$. We set the link capacities as $C^{HL} = 10, C^{LS} = 40, C^{SE} = 40$. For the Poisson arrival processes from hosts, we set λ as 2. For the MMOO arrival processes, we set $P = 12$, $\lambda_1 = (0.7, 0.4, 0.01)$, and average rate $P\frac{1-\lambda_2}{2-\lambda_1-\lambda_2} = 2$. The units are $Gbps$, or normalized in certain granularity. 0.7, 0.4 and 0.01 represent from high to low traffic burstiness. We further assume the out-of-rack probability $p^{\mathbf{X}^L}$ as 0.2 (approximately the measurement results from [4], [5] and the forecast in [2]), and the probability from spine switch to DCI edge $p^{\mathbf{X}^S}$ as 0.1. The probabilities from the leaf to spine switches as well as the traffic locality between these two tiers should be evenly smoothed by load balancing and big traffic volume, such that we set $p^{\mathbf{Y}^L} = \frac{1}{K}$.

Figure 9 illustrates the inverse proportion between DCI bandwidth and delay bound by plotting the corresponding ε-quantiles (in time units) with $\varepsilon = 10^{-3}$. With the same average rate of arrival processes, we see that more burstiness (MMOO with $\lambda_1 = 0.7$) of arrivals will lead to larger delay. Besides, with the aid of this calculus, data center designer and operator can regulate the DCI bandwidth, particularly under certain delay requirement and with measured traffic load.

Figure 10 illustrates the impact of number of spine switches over the stochastic delay bounds at DCI channel. The numerical settings keep the same except that we set DCI bandwidth as 400. Interestingly, the figure indicates that we obtain the minimal stochastic delay bound at $K = 3$. Usually, each spine switch potentially produces burstiness to the DCI channel. Aggregating more these traffics should generate larger stochastic delay bound. However, fewer spine switches also mean that each spine switch will aggregate more burstiness from the layer below, such that generate more possibly bigger burstiness. On the other hand, we should note that $K = 3$ is not the optimal solution, since this is generated only when observing the DCI channel. Bigger K means that traffic load is balanced towards more spine switches such that traverse them with smaller delay. Therefore, we should consider end-to-end delay or harmonise DCI delay with other link delays in DC fabric. Since too much flow demultiplexing happens in the data center network, there will exist many end-to-end flows that contribute commonly to the overall DCI delay while each traverses heterogeneous path with different delays. The end-to-end observation is ambiguous. From Figure 10 we also see that for the arrival processes with more burstiness or for a DCI channel connected with more spine switches, the delay bounds increase faster, which implies that for these settings DCI channel becomes the bottleneck faster.

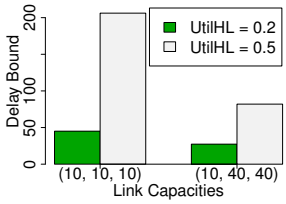


Figure 12. Delay bounds of DCI channel changing link capacities.

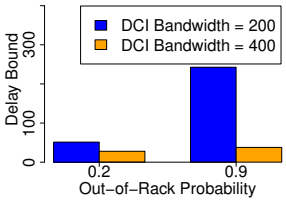


Figure 13. Delay bounds of DCI channel changing out-of-rack probability.

In Figure 11, we show the stochastic delay bounds enlarging the DC scale - 10 times of server and leaf switch amount as in Figure 9. As single DCI channel is instable, we choose to distribute the traffic to 10 DCI edge and links. Comparing with Figure 9, the delay bounds are enlarged although the utilization keeps unchanged for each DCI link. This is reasonable, since the number of potential burstiness sources are 10 times enlarged. Decreasing the number of spines can reduce the delay bounds. The proposed calculus can expose when we should deploy more DCI links to offload the traffic according to their stability and/or SLAs. The scalability problem might further closely relate to DC architecture, i.e., in order to facilitate larger scale, new DC architecture design might be introduced. The proposed lemmas in this paper are not dependent of the DC architectures, therefore, the method should cover usual architectures, e.g., iterative spine-leaf architectures.

Figure 12 plots the stochastic delay bounds when changing the link capacities (C^{HL}, C^{LS}, C^{SE}) from $(10, 10, 10)$ to $(10, 40, 40)$, while considering two arrival intensities such that the utilization of the host-to-leaf link equal to 0.2 or 0.5. The delays increase in utilization and decrease if enlarging the switch capacities. The delay improvement for the low utilization is not very impressive. Hence, for a data center with low utilization, we can carefully choose switches with lower capacity. Figure 13 illustrates the impact of out-of-rack probability and DCI bandwidth over stochastic delay bounds. Clearly, adequate DCI bandwidth will strongly diminish the out-of-rack burstiness.

Next, we show our calculus for Fat-Tree architecture by plotting the stochastic delay bounds calculated from Eq. (7)

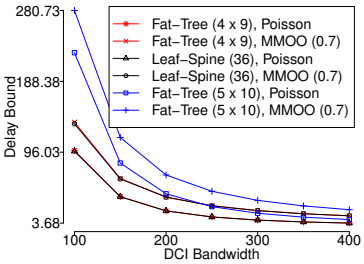


Figure 14. Delay bounds of DCI channel changing bandwidth with architecture Fat-Tree vs. Leaf-Spine.

and Proposition 1 in Figure 14. Again, we consider homogeneity, i.e., let $J^{N_m} = J, N_m = N$, $C^{m,n_m,j} = C^{HA}, C^{m,n_m} = C^{AL}$). Intuitively, the delay bounds decrease in DCI bandwidth. In turn, we compare these stochastic delay bounds with those obtained using Theorem 1 for Leaf-Spine architecture. Note, in order to equalize the source flow numbers in both calculation, we set $N^{Fat-Tree} \times J = N^{Leaf-Spine} = 36$ and $N^{Fat-Tree} = 4$. However, it is difficult to find the equivalence between a single layer link capacity (host-leaf) and two layer link capacity (host-access-aggregate), especially considering the restricted link capacity of aggregate switch. We set the link capacities (C^{HA}, C^{AL}) as $(10, 40)$. Interestingly, both architectures perform almost the same from the point of view of DCI channel, or more carefully, Leaf-Spine a little bit better. Thus, the focus of design can be put more onto the other aspects such as scalability, cost and energy efficiency, etc. In addition, we enlarge the scale of Fat-Tree architecture a bit to $N = 5, J = 10$, and plot the increased stochastic delay bounds. Stochastic delay bounds derived with smaller DCI bandwidth is more scalability sensitive.

VI. Conclusion

In this paper, we introduced a calculus on data center network, particularly from DCI's point of view. We believe that this paper can provide valuable aid to data center network design and operation with insightful analysis. Accordingly, this paper widens the modelling scope of network calculus. To that end, we have introduced several network calculus components to data center network model, and derived the flow and stochastic delay bounds for DCI channel. Consequently, the close-formed results have revealed the impact of different system parameters over the stochastic delay bounds analytically that we evaluated by numerical examples.

Acknowledgments

This work has been fully funded by the German Federal Ministry of Education and Research (BMBF) in the Celtic-Plus program SENDATE under contract 16KIS0483 and 16KIS0479 (cluster Secure-DCI).

References

[1] Cisco Data Center Spine-and-Leaf Architecture: Design Overview. White Paper, 2016.

[2] Cisco Global Cloud Index: Forecast and Methodology, 2015 - 2020. White Paper, 2016.

[3] M. Alfares, A. Loukissas, and A. Vahdat. A Scalable, Commodity Data Center Network Architecture. *ACM SIGCOMM Computer Communication Review*, 38(4):63–74, 2008.

[4] T. Benson, A. Akella, and D. Maltz. Network Traffic Characteristics of Data Centers in the Wild. In *Proceedings of Internet Measurement Conference*, November 2010.

[5] T. Benson, A. Anand, A. Akella, and M. Zhang. Understanding Data Center Traffic Characteristics. *ACM SIGCOMM Computer Communication Review*, 40(1):92–99, 2010.

[6] S. Bondorf and J. Schmitt. Boosting Sensor Network Calculus by Thoroughly Bounding Cross-Traffic. In *Proceedings of IEEE INFOCOM*, April 2015.

[7] C.-S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[8] Y. Chen, S. Jain, V. K. Adhikari, Z. Zhang, and K. Xu. A First Look at Inter-data Center Traffic Characteristics via Yahoo! Datasets. In *Proceedings of IEEE INFOCOM*, pages 1620–1628, 2011.

[9] F. Ciucu, J. Schmitt, and H. Wang. On Expressing Networks with Flow Transformation in Convolution-Form. In *Proceedings of IEEE INFOCOM*, pages 1979–1987, April 2011.

[10] R. L. Cruz. A Calculus for Network Delay, Part I and II. *IEEE Transactions on Information Theory*, 37(1):114–141, January 1991.

[11] M. Fidler. An End-to-End Probabilistic Network Calculus with Moment Generating Functions. In *Proceedings of IEEE IWQoS*, pages 261–270, June 2006.

[12] M. Fidler. A Survey of Deterministic and Stochastic Service Curve Models in the Network Calculus. *IEEE Communications Surveys & Tutorials (COMST)*, 12(1):59–86, 2010.

[13] M. Fidler and Y. Jiang. Non-asymptotic Delay Bounds for (k, l) Fork-Join Systems and Multi-Stage Fork-Join Networks. In *Proceedings of IEEE INFOCOM*, pages 1–9, 2015.

[14] 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. *ACM SIGCOMM Computer Communication Review*, 39(4):51–62, 2009.

[15] 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 *Proceedings of ACM SIGCOMM Conference on Data Communication*, pages 63–74, 2009.

[16] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: A Scalable and Fault-tolerant Network Structure for Data Centers. *ACM SIGCOMM Computer Communication Review*, 38(4):75–86, 2008.

[17] Y. Jiang and Y. Liu. *Stochastic Network Calculus*. Springer-Verlag, 2008.

[18] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The Nature of Data Center Traffic: Measurements & Analysis. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement*, pages 202–208, 2009.

[19] J.-Y. Le Boudec and P. Thiran. *Network Calculus - A Theory of Deterministic Queuing Systems for the Internet*. Number 2050 in Lecture Notes in Computer Science. Springer-Verlag, 2001.

[20] C. Li, A. Burchard, and J. Liebeherr. A Network Calculus with Effective Bandwidth. *IEEE/ACM Transactions on Networking*, 15(6):1063–6692, December 2007.

[21] X. Li and C. Qian. An NFV Orchestration Framework for Interference-Free Policy Enforcement. In *Proceedings of IEEE ICDCS*, pages 649–658, 2016.

[22] Z. Lu, Y. Yao, and Y. Jiang. Towards Stochastic Delay Bound Analysis for Network-on-Chip. In *Proceedings of the 8th IEEE/ACM International Symposium on Networks-On-Chip*, pages 64–71, 2015.

[23] R. Morris and D. Lin. Variance of Aggregated Web Traffic. *Proceedings of IEEE INFOCOM*, 1:360–366, 2000.

[24] P. Nikolaus and J. Schmitt. On Per-Flow Delay Bounds in Tandem Queues under (In)Dependent Arrivals. In *Proceedings of IFIP NETWORKING*, 2017.

[25] A. Rizk, F. Poloczek, and F. Ciucu. Computable Bounds in Fork-Join Queueing Systems. In *Proceedings of ACM SIGMETRICS*, pages 335–346, 2015.

[26] H. Wang, F. Ciucu, and J. Schmitt. A Leftover Service Curve Approach to Analyze Demultiplexing in Queueing Networks. In *Proceedings of ICST VALUETOOLS*, pages 168–177, October 2012.

[27] H. Wang and J. Schmitt. Load Balancing - Towards Balanced Delay Guarantees in NFV/SDN. In *Proceedings of the 2016 IEEE NFV-SDN*, November 2016.