Future Data Center Networking: From Low Latency to Deterministic Latency

Feixue Han, Mowei Wang, Yong Cui, Qing Li, Ru Liang, Yashe Liu, and Yong Jiang

Abstract

Many cloud applications in modern data centers are very demanding on latency, thus researchers have paid much attention to building a data center network with ultra-low latency, even deterministic low latency. Motivated by this research trend, this article focuses on effective latency reduction designs, which mainly aim at reducing the queuing delay in switches. We first summarize a development roadmap to give an overview of the three developing stages of existing schemes, and point out the essential difference between these stages is the amount of traffic information maintained by the control nodes. After briefly reviewing the features and deployment complexity of recent advances, we map them to three stages, introducing their design principles and identifying the problems they target at. Finally, we present the challenges and opportunities for future work.

Introduction

The data center network has become an indispensable infrastructure for modern Internet and cloud computing. In the past decade, data center networks have been rapidly growing in terms of size and link speed. At the same time, a substantial amount of computing, storage, and communication is shifting to data centers. Nowadays, the data center carries most of the network traffic for cloud service providers like Amazon, Microsoft and Google.

Many data center applications (e.g., web search and data mining) are based on partition/aggregation workflow pattern, which is often sensitive to delay and thus making latency a primary metric for evaluating the data center network performance [1]. As represented by web search, to serve a user request, thousands of back-end servers may have to work cooperatively and exchange information across the whole data center within a short period. Modern data center networking hardware offers the potential for ultra-low processing delay [2], under this circumstance, queuing in the network dominates the end-to-end latency. To reduce the average latency, numerous schemes throttle data sending rates according to network status to avoid queuing as much as possible [3–7].

For applications such as real-time database query [7], resource disaggregation [8], and storage [6], completing transfers before their specific deadlines becomes one of the most important performance requirements. Ensuring the timely delivery of each packet is the critical foundation to meet the deadlines of flows, especially for the applications (e.g., distributed user query) that are dominated by very short messages [2]. Deterministic latency refers to that the end-to-end latency of each data packet must not exceed a prescribed bound. To achieve bounded end-to-end latency, there are two requirements. On the one hand, the queuing delay should be limited, that is, the queue length should have an upper bound. On the other hand, the buffer overflow caused by the burst traffic should be avoided, especially in the case that modern data center switches have very shallow buffers. In light of the above problems, a series of designs are proposed that apply credit-based schemes [1, 2, 9–11] or reconfigurable networks [8, 12–14] to prevent the load from exceeding the capacity of the bottleneck. Ideally, a completely zero-queuing network can be constructed through reasonable design and scheduling [15]. Under this circumstance, it can not only minimize the queuing delay but also alleviate the maintenance cost of hardware, for removing buffer can reduce the probability of hardware failure and the complexity of maintenance simultaneously. But the accuracy of fine-grained traffic control is limited by the inevitable deviation of the clock on network devices, so eliminating congestion in data centers remains a great challenge.

This article presents a survey of schemes that contribute to building a low latency data center network. Instead of concentrating on a certain aspect of technology such as congestion control (CC) or flow scheduling, this article pays more attention to the characters from low latency designs to deterministic low latency designs and presents part of the representative works. We contend that the essential difference between the exhibition designs is the amount of traffic information held by the control nodes. Therefore, three developing stages are summarized according to the traffic information increment obtained by the control nodes: network traffic agnostic, partial traffic information, or global traffic information for control nodes. We draft a development roadmap to present these stages and point out their key design considerations more intuitively. After briefly reviewing the features and deployment complexity of recent advances, we map them to the three development stages and introduce their design principles and innovations. In addition, we present the challenges and opportunities for future research in this area.

The rest of this article is structured as follows: the next section briefly elaborates the elements of the technology roadmap. Following that we introduce the representative advances in the three

Digital Object Identifier:
10.1109/MNET.102.2000622

Yong Cui (corresponding author) and Mowei Wang are with Tsinghua University; Feixue Han and Yong Jiang are with Tsinghua Shenzhen International Graduate School; Qing Li is with Peng Cheng Laboratory; Ru Liang and Yashe Liu are with Huawei Technologies.

developing stages. We then present the challenges and opportunities for future work. Following that, we give a brief conclusion in the last section.

Development Stages and Design Considerations

In this article, the traffic information refers to the transmission requests (i.e., flow size, source, and destination for each transfer) or the transmission ability (i.e., when a server can deliver the data) of end-hosts. The control node represents the devices (switches or end-hosts) that regulate the transfers in the network. Based on the increase of traffic information, we divide the existing designs into three development stages. And the designs in each stage usually have the same considerations (e.g., use what kind of congestion signal) but make different choices. Figure 1 shows the division of the three stages and lists the key design considerations in these stages respectively. We will discuss them in detail in the following.

Network Traffic Agnostic

Network traffic agnostic means the control nodes do not explicitly collect the traffic information in advance to judge the network congestion status, and then the sending rate is adjusted according to the traffic information reflected in the network feedback (such as ECN) until it converges to a relatively stable value. These reactive schemes usually have same the considerations: what kind of congestion signal to adopt, how to adjust the congestion window or sending rate (Adjustment Algorithm), and the frequency of adjustments (Adjustment Interval).

The most adopted congestion signals include the round trip time (RTT) and notifications carried by ACKs, such as Explicit Congestion Notification (ECN) and Inband Network Telemetry (INT). ECN is a single bit marker that marks the Congestion Experienced (CE) codepoint of the packet header when the buffer occupancy in switches exceeds the set threshold. INT can return fine-grained information of each port passed by, but it has not been widely deployed in commodity switches. Based on the feedback, senders can adopt window-based adjustment [7] or pure rate control [3]. Besides, it is important to choose a reasonable adjustment interval. Adjusting per RTT can hardly make a timely reaction in burst scenarios while adjusting per ACK can be susceptible to the noise. Reasonable tuning of the above parameters can ensure network stability and overall fairness among different flows, whereas these designs can only take effect after the congestion occurs.

Partial Traffic Information

Since it's hard for the reactive transports to achieve quick convergence and maintain low latency, practitioners propose that more information can be provided for the control nodes before data transmission. In these schemes, each receiver collects the information of the transfers destined for it but does not concern about other nodes (which means partial). The receivers regulate the transfers proactively with the credits, usually according to their link speeds.

At least one RTT is required for the proactive transports to allocate credits for a new flow, which is called the pre-credit phase [11]. Different transmission strategies can be employed in the pre-credit phase. If the sender sends no packet (e.g., ExpressPass), the new flows will be paused by one RTT and the link may stay idle. But if the packets burst at a high rate, it can cause congestion and even packet drops. Besides, if the credits are rate-limited in the network, path symmetry should be guaranteed in case of great performance degradation. What's more, a sender may receive credits from different receivers at the same time but can only respond to one of them. Under this circumstance, to maximize the bandwidth utilization, slightly overcommit the links should be considered.

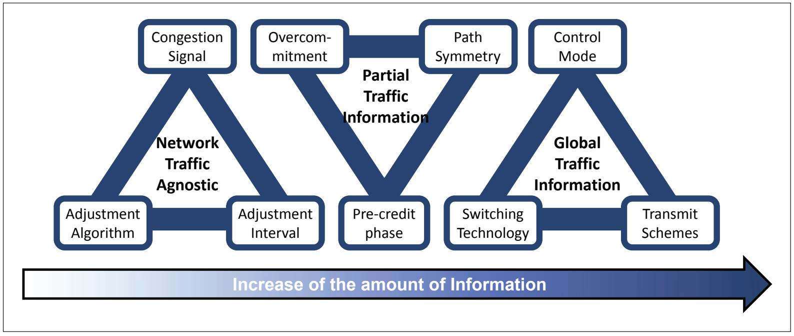


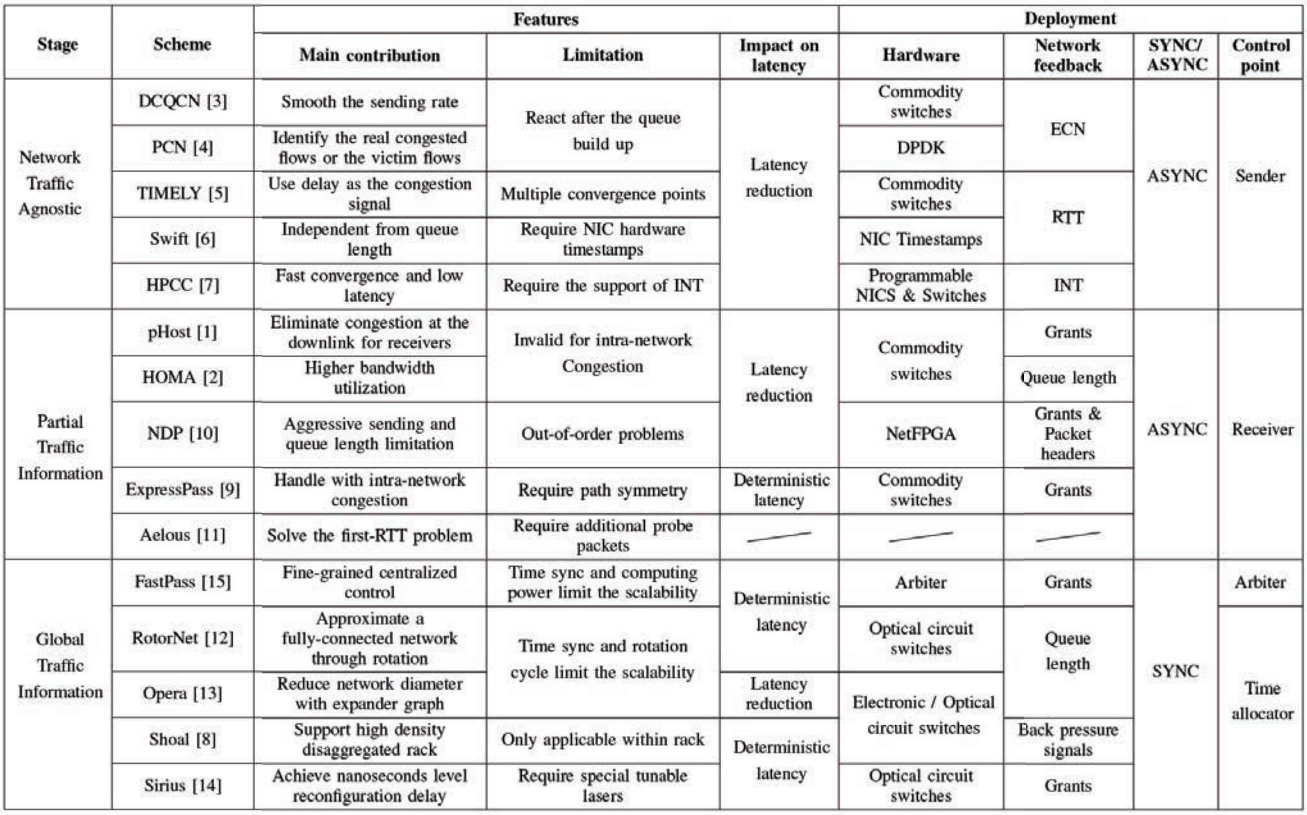
FIGURE 1. Development roadmap: illustration of the three development stages (marked in triangles) divided according to the amount of information; there are three key design considerations (marked in boxes) for each stage.

In proactive transports, the link capacities are proactively allocated to ensure high link utilization and low end-to-end latency. However, considering that the receivers perform independent scheduling, multiple flows may arrive in burst, which can cause queuing at the bottleneck [9]. This reveals that simple scheduling on control nodes can't guarantee the limited queue length on switches, so does the deterministic latency. Note that based on the proactive bandwidth allocation, if the sending rate of the credits is further limited by the switches, the queue build-up can be bounded.

Global Traffic Information

In these transports, the control nodes hold the traffic information of all the end-hosts in the data center. With precise time synchronization, these solutions can avoid the collisions of packets through reasonable scheduling. But to address the skewed traffic and reduce the waiting time for direct connection, they usually utilize several optimization methods, such as Valiant Load Balancing (VLB). These methods can cause packet queuing at the transit nodes. If there are no measures to limit the queue length, the upper bound of the latency can not be guaranteed.

In the light of control mode, these schemes can be divided into two categories. The centralized one uses a central arbiter to gather all the transmission requests in the network and schedule traffic at a fine-grained level. Regardless of time accuracy, centralized arbitration can eliminate the queues in packet switches [15]. The distributed one utilizes the principle of time division multiplexing and circuit switches to build a reconfigurable network [8, 12–14]. In these designs, specific time slices are allocated for all the senders in the network, which is pre-known by the transit switch-



Stage	Scheme	Features			Deployment			
		Main contribution	Limitation	Impact on latency	Hardware	Network feedback	SYNC/ ASYNC	Control point
Network Traffic Agnostic	DCQCN [3]	Smooth the sending rate	React after the queue build up	Latency reduction	Commodity switches	ECN	ASYNC	Sender
	PCN [4]	Identify the real congested flows or the victim flows			DPDK			
	TIMELY [5]	Use delay as the congestion signal	Multiple convergence points		Commodity switches	RTT		
	Swift [6]	Independent from queue length	Require NIC hardware timestamps		NIC Timestamps			
	HPCC [7]	Fast convergence and low latency	Require the support of INT		Programmable NICS & Switches	INT		
Partial Traffic Information	pHost [1]	Eliminate congestion at the downlink for receivers	Invalid for intra-network Congestion	Latency reduction	Commodity switches	Grants	ASYNC	Receiver
	HOMA [2]	Higher bandwidth utilization				Queue length		
	NDP [10]	Aggressive sending and queue length limitation	Out-of-order problems		NetFPGA	Grants & Packet headers		
	ExpressPass [9]	Handle with intra-network congestion	Require path symmetry	Deterministic latency	Commodity switches	Grants		
	Aelous [11]	Solve the first-RTT problem	Require additional probe packets	—	—	—		
Global Traffic Information	FastPass [15]	Fine-grained centralized control	Time sync and computing power limit the scalability	Deterministic latency	Arbiter	Grants	SYNC	Arbiter
	RotorNet [12]	Approximate a fully-connected network through rotation	Time sync and rotation cycle limit the scalability		Optical circuit switches	Queue length		Time allocator
	Opera [13]	Reduce network diameter with expander graph		Latency reduction	Electronic / Optical circuit switches			
	Shoal [8]	Support high density disaggregated rack	Only applicable within rack	Deterministic latency		Back pressure signals		
	Sirius [14]	Achieve nanoseconds level reconfiguration delay	Require special tunable lasers		Optical circuit switches	Grants		

TABLE 1. Comparison among existing latency reduction works.

es. And the topology can be reconfigured according to the real-time traffic distribution or a series of preset configurations. Besides, two switching technologies, that is, packet switching and circuit switching, are applied in these schemes. Compared to packet switches, circuit switches (optical or electrical) are significantly more power and cost-efficient because there are no buffers, no arbitration, and no packet inspection mechanisms. However, traditional circuit switches have reconfiguration delays in the order of few microseconds to even milliseconds [8]. These designs also adopt different transmit schemes, that is whether delivering the data through one or multi-hops. Delivering data through one-hop can fully utilize the bandwidth, but the flow may have to wait for a whole rotation cycle for a direct connection. Transmitting data through multi-hops can greatly cut down the waiting time. However, this imposes a "bandwidth tax" which means the indirect transmission occupies more bandwidth.

Summary

These three stages reflect the transition from merely latency reduction to deterministic latency. In the traffic agnostic stage, the senders analyze the congestion status through network feedback carried by ACKs, which makes the adjustment lag behind the occurrence of congestion. The reactive transports can effectively relieve the congestion in the network, but can hardly eliminate the queuing in the network. When providing partial traffic information for the receivers, the receivers proactively manage the transfers with a fine-grained scheduling, that is, the data must be sent according to the credits (excluding the first RTT). This mechanism achieves good latency performance and reduces the possibility of congestion. Since the receivers usually set the sending rate of credits according to their link speed, the packets may still get congested at a specific bottleneck. But based on the proactive scheduling, if the credits are further rate-limited (e.g., to match the capacity of the reverse path), the latency of packet transmission is also bounded. In schemes with time synchronization, through central arbitration for each request or assigning time slices for senders, the deterministic delay can be ensured without additional scheduling.

To present a preliminary understanding of these works and facilitate the following introduction, we map them into their corresponding stages mentioned above, and list their most notable advantages and disadvantages, as Table 1 shows. Since most of them have not been widely deployed in data centers, the table also shows the deployment complexity in terms of hardware, control nodes, and so on. In this table, hardware refers to the required hardware for each design, especially the costly and uncommon ones. And scalability restriction refers to the design points which may influence the growth of the network scale. The control nodes indicate who controls the sending of data.

The influence of different parameters and approaches should be taken into account when designing latency-reduction schemes. In addition to the points mentioned, more perspectives such as the priorities and scheduling granularity worth further exploring.

Network-Traffic Agnostic Schemes

In the following reactive transports, the network traffic is transparent to the senders before data transmission. They adjust their congestion window or sending rate according to network feedback (e.g., ECN). These schemes can effectively alleviate the congestion in the network, but can not guarantee deterministic latency.

To reduce the CPU processing overhead, Remote Direct Memory Access (RDMA) has been introduced into data centers. RDMA implements kernel bypass and zero-copy which significantly reduces the latency at the end-host and network interface cards (NICs) [3, 9]. Thus, it can meet the high throughput and ultra-low latency requirements of modern data center applications when there is no congestion.

The current RDMA transmission protocol on Ethernet is RDMA over Converged Ethernet (RoCEv2) [7], which is based on the connectionless UDP protocol, thus packet loss can greatly reduce the transmission efficiency. Therefore, operators utilize Priority-based Flow Control (PFC) to build a lossless network [3, 4, 7]. PFC creates eight virtual channels on Ethernet links, allowing virtual channels to be suspended and restarted individually while traffic in other channels can pass through without interruption. When congestion occurs, PFC does not distinguish between flows and forces the immediate upstream entity to pause data transmission. This can lead to head-of-line blocking and unfairness. So, there is a need for reasonable congestion control designs to avoid the triggering of PFC.

DCQCN [3] was proposed to add per-flow basis congestion control in RDMA networks. DCQCN is a hardware rate-based protocol that utilizes the principle of DCTCP. Instead of controlling the congestion window, DCQCN directly controls the sending rate through timers and byte-counters. When a Quantized Congestion Notification (QCN) is received, the transmission rate is reduced, otherwise, it will increase slowly. DCQCN reduces the occurrence of PFC and alleviates the impairment brought by it.

DCQCN alleviates the occurence of PFC, but the problem of victim flow remains unsolved. PCN [4] proposes a mechanism to identify the congested flows and the victim flows when they are churned in the same queue. Based on the traditional ECN method, when a switch receives a RESUME packet, the packets queuing in the egress port will not be treated as the cause of congestion or be marked with ECN.

Then the subsequent packets will be marked with a threshold of zero. Besides, PCN sends data according to the receiving rate, which greatly accelerates the convergence of rate adjustment.

TIMELY [5] is the first design that regards the RTT as the congestion signal. Since NICs can provide high-quality time-stamping at the level of microseconds, TIMELY measures the data center RTTs with sufficient precision. Different from the idea of presetting a threshold, TIMELY pays attention to the rate of RTT variation to predict whether the congestion is about to occur. TIMELY does not require any support on switches and can be easily deployed. But the sending rate can either converges to a point without congestion or a point with deep queuing. In other words, there can be multiple convergence points in this algorithm.

Swift [6] is an evolution of TIMELY [5] based on Google's production experience. The major idea of Swift is to adapt the rate to a target end-to-end delay. It decomposes the RTT into fabric and host portions to respond separately to different causes of congestion. Accordingly, a fabric congestion window (fcwnd) and an end-host congestion window (ecwnd) are maintained to provide different congestion responses. For simplicity, the fcwnd and ecwnd are modulated with AIMD algorithm, and the effective congestion window (cwnd) is combined as min(fcwnd,ecwnd). At last, the pacing delay is calculated with the RTT and the cwnd.

The above solutions represent the state-of-art congestion control schemes with ECN and RTT, but they are all heuristic algorithms that require multiple iterations to converge to a stable transmission rate. Because the information carried by RTT and ECN is insufficient to directly calculate the appropriate transmission rate, thus impede fast convergence. Since INT became available, HPCC [7] leverages its features to obtain fine-grained network load information from switches and controls traffic precisely. HPCC computes the real-time output rate of each port through the transmitted data and timestamp provided by INT, then it uses the computed output rate and queue length to estimate the number of inflight bytes in the most congested link. At last, HPCC adjusts the congestion window according to the inflight bytes. HPCC can quickly converge to utilize free bandwidth while avoiding congestion and can maintain near-zero in-network queues for ultra-low latency.

The essence of the designs mentioned above is to estimate or calculate the inflight bytes and compare them with the Bandwidth-Delay Product (BDP, data sent at line rate within a base RTT) to throttle the senders' sending rates. These designs greatly reduce the latency and improve link utilization without additional hardware, but they cannot eliminate PFC especially under incast scenarios. Therefore, they can't provide an explicit latency upper bound.

Partial Traffic Information for Receivers

To enhance the link utilization and reach the right rate in each round, receivers proactively collect the information of the transfers destined for it (partial traffic information) and allocate bandwidth to senders as credits. In the pre-credit phase, the senders can wait idly or burst packets at a high rate. And they all transmit the pending packets according to the credits from the first credit's arrival. The packets transmitted in the pre-credit phase and following are called unscheduled and scheduled packets separately.

pHost [1] is a representative of receiver-driven proactive congestion control, which was proposed in 2015. The sender puts the flow's information in the request packet and the receiver

To enhance the link utilization and reach the right rate in each round, receivers proactively collect the information of the transfers destined for it (partial traffic information) and allocate bandwidth to senders as credits. In the pre-credit phase, the senders can wait idly or burst packets at a high rate.

The global centralized control schemes use a central arbiter to process all the requests in the data center. In addition to each packet's timing, it can also control the paths taken.

schedules flows according to a specific scheduling algorithm. pHost decouples the network fabric from the scheduling decisions. It solves the congestion at the downlink for receivers and achieves near-optimal performance.

However, pHost only sets two priorities for scheduled and unscheduled packets and can hardly resolve the intra-network congestion. To solve this problem, researchers from KAIST proposed ExpressPass [9] in 2017. ExpressPass performs congestion detection and traffic scheduling on all the switches. The switches shape the flow of credit packets, allowing credits to take one maximum transmission unit (MTU) transmission interval at the bottleneck link. By limiting the credit packets' sending rate, ExpressPass can determine the available bandwidth in the reverse path and the latency upper bound. At the same time, it arranges the sending time of each packet, which can avoid queuing if the RTT of all transmission paths is consistent.

Compared to pHost and ExpressPass, NDP [10] is a more aggressive data center transport architecture without a three-way handshake connection establishment phase, and it allows the senders to start transmission at a line rate. Considering that per-flow equal-cost multipath routing (ECMP) hashing may cause flow collision, NDP employs a load balancing mechanism in packet granularity and it can establish connections but this scheme causes inevitable packet out-of-order. The sender randomly arranges the paths and directly controls the path selection of each packet. When the queue length reaches a fixed threshold (8 packets), the switch trims the packet and only keeps the header in a high priority queue to achieve fast retransmission. With very shallow-buffered switches, NDP achieves both low latency and high throughput through an aggressive sending mechanism.

To avoid the out-of-order problem for packet scheduling and head-of-line blocking problem for flow scheduling, Homa [2] schedules messages instead. To bypass queues for short messages, receivers pick cut-offs based on the cumulative distribution function (CDF) of message sizes, then Homa applies the shortest remaining processing time first (SRPT) to dynamically assigns network priorities for messages based on the current cut-offs. Homa points out when several credits arrive at the same server, the sender can't respond to all receivers simultaneously. This can cause bandwidth waste and significantly poor performance under heavy load. Thus Homa lets the receivers overcommit their downlink, and the degree of overcommitment depends on the number of priorities. Homa performs well especially for small messages and can sustain higher workloads.

Whether to transmit data at the first RTT poses a basic dilemma for the above solutions that compromises their performance. To achieve the best performance, two principles should be met. First, the unscheduled packets should burst in the first RTT to fully utilize the spare bandwidth. Second, the scheduled packets should not be influenced by the congestion or dropouts caused by unscheduled packets. Aelous [11] achieves these two goals through a selective dropping method. When there exists the spare bandwidth leftover by scheduled packets, Aelous allows the packets in the pre-credit phase to burst at line rate, but immediately drops them once the bandwidth is used up. Therefore, Aelous effectively utilizes the available bandwidth while safeguarding the scheduled packets.

Global Traffic Information Synchronization

This section introduces the designs in which the control nodes hold the request or configuration information across the whole data center. They can achieve more fine-grained and efficient scheduling, further reducing the end-to-end latency. But as Table 1 shows, these schemes usually require time synchronization that restricts their scalability.

Centralized Control

The global centralized control schemes use a central arbiter to process all the requests in the data center. In addition to each packet's timing, it can also control the paths taken.

Fastpass [15], as a representative work of centralized control, achieves fine-grained control over time assignment and path selection through a centralized arbiter. Fastpass treats the whole data center as a big switch and achieves network-wide sub-microsecond time synchronization. The arbiter uses quick maximal matching to assign the senders a set of timeslots and thus realizing max-min fairness. It also uses bipartite graph to determine path selection. To solve the problem of arbiter failures in Fastpass, multiple arbiters are used to receive requests simultaneously, but only the preset primary arbiter responds. Meanwhile, packet loss in Fastpass can be used as a signal of link failure. Fastpass matches the flow rate with the capacity of links, hence packets experience no queuing delays in the network. And it can also adapt to heterogeneous loads well.

Without regard to the influence of clock accuracy, this scheme can eliminate the queuing and achieve high network link utilization. However, in a centralized control scheme, the central arbiter has to process all the requests and calculate the appropriate data transmission time, which brings challenges to the computing power of the arbiter. Besides, since the path has been selected in advance, it is hard to deal with link failures.

Distributed Control

Compared to centralized control schemes, a distributed network has good fault tolerance and scalability. To design a distributed synchronous network, existing schemes mostly utilize reconfigurable circuit switches and a time division multiplexing mechanism. It allocates link capacity in a time-varying fashion and this can be realized through fixed scheduling or response to traffic demands. The former can achieve rapid and deterministic configuration, the latter can address skewed traffic better. In addition, since the workload of the applications is not always consistent with the network structure, under-provisioned networks generally adopt indirect routes, which means some packets may be transmitted through longer paths. This can cause bandwidth waste

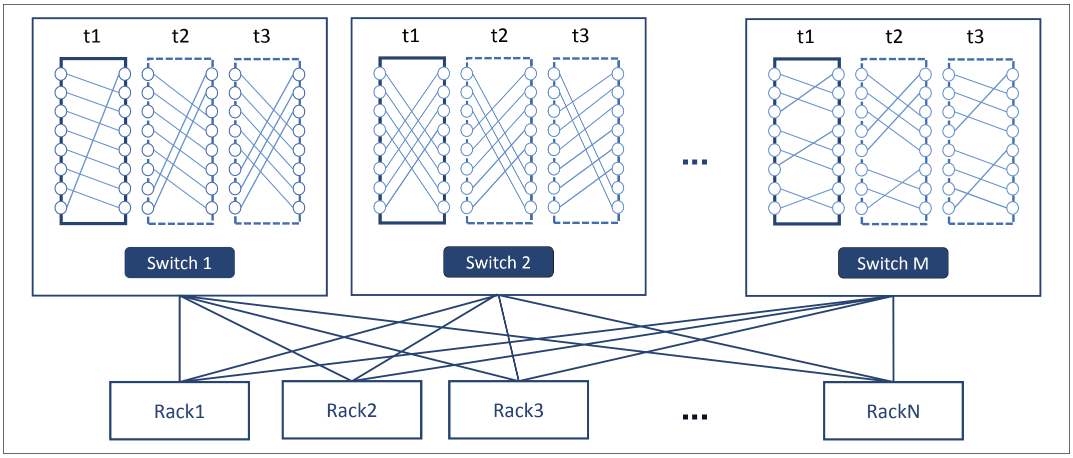


FIGURE 2. Circuit switches rotate among a set of configurations.

called bandwidth tax.

RotorNet [12], Opera [13] and Shoal [8] all adopt reconfigurable circuit switches and decouple the switch configuration from the traffic pattern. They don't require unified scheduling of demands or a centralized controller. Each switch independently rotates according to a predetermined set of configurations, as Fig. 2 shows. Through rotation, these configurations approximate a fully connected network, providing the same bandwidth between all nodes. Racks can exchange traffic and scheduling information when they are directly connected.

On this basis, due to the long waiting time for direct connection, RotorNet adopts an additional packet switch to guarantee the timely delivery for mice flows, while this increases deployment complexity and has poor performance under high burst scenarios.

Opera [13] solves this problem. To ensure the timely delivery of mice flows, they are sent directly at any time and possibly through multiple hops. Large flow covers the vast majority of the bytes in the data center, therefore, they can wait for a direct connection to avoid bandwidth tax. Opera allows only one switch to be reconfigured at each time and re-routes the flow passing through the selected switch to other paths in advance. And the links in the active ones can constitute an expander topology which can provide multiple short paths between any racks simultaneously. Besides, Opera utilizes the expander graph to construct a low-diameter network and guarantees each pair of racks can directly connect within a cycle.

Shoal [8] was proposed to solve the problem of redundant nodes caused by resource decomposition in racks. It is motivated by fast circuit switches which can be reconfigured in tens of nanoseconds, so it introduces a multi-layer structure composed of low port-count circuit switches. To achieve high bandwidth utilization, Shoal employs a two-hop forwarding which can cause unavoidable queue accumulation. Thus, Shoal employs the back-pressure mechanism, each node maintains a queue that can only hold one fixed-size cell. When there is a cell in the queue, the source node will be notified to stop sending packets.

The reconfiguration delay of circuit switches brings great challenges to meet the stringent latency requirement of the hardware-driven cloud workloads. Therefore, Sirius [14] provides the abstraction of an optical switch which reduces the delay to nanoseconds level. Sirius equips the nodes (hosts or TOR switches) with tunable lasers which can change the wavelength (used to carry data). Then Sirius uses a single layer of gratings to route the incoming light to an output port based on its wavelength. To achieve fast reconfiguration, it replaces the wavelength tuning with laser switching. Besides, Sirius also follows a pre-determined, static schedule that specifies the connectivity at any given fixed-size timeslot.

Distributed Control schemes reduce the waiting time for central arbitration and greatly optimize the latency for mice flows, especially when they can be sent directly without waiting for a direct connection. However, the characters of circuit switches lead to the unavoidable rotation cycle, thus elephant flows suffer a longer waiting time for a direct connection.

Broader Perspectives

Through the introduction of the existing schemes, we can observe that the traffic patterns and network architectures vary a lot in different scenarios, which leaves much design space for new solutions. This section presents the challenges and opportunities of designing a deterministic low latency traffic control scheme in data center.

Scalability

The synchronization data center network has attracted extensive attention, but its scalability is limited by the following factors. Due to the clock drift, the time deviation is inevitable. And the precision of time synchronization algorithms usually decreases as the network expands. Apart from the time synchronization precision, the synchronized networks also face other scalability restrictions. The global centralized control solutions use an arbiter to process all the requests in the network, thus the computing capability of the arbiter becomes the main bottleneck of network expansion. In the distributed control schemes, the switches rotate among a set of configurations. The port number of a switch is limited but all the racks (servers) have to connect with the switch once during a rotation cycle. As the number of servers increases, the length of the rotation cycle grows accordingly, which leads to poor scalability.

Optical Switching

Optical fiber greatly reduces the propagation delay, but it also brings the expensive cost of photoelectric conversion modules. Introducing optical circuit switches can effectively reduce this expense as well as the power costs. Besides, the optical switches save the processing time because there is no packet processing or arbitration. However, the deployment of optical switches faces several challenges. Optical switches have poor fault-tolerant capability because there is no buffer to deal with the packet conflicts. And the fine-grained scheduling puts stringent requirements for precise time synchronization.

Self-Configuring Networks

The existing reconfigurable network designs let the switches rotate among a set of preset configurations. However, large data centers have to deal with various cloud workloads and there often exist a few hotspots. Therefore, deploying adaptive rotation configurations is promising to alleviate the competition between flows and improve performance. However, dynamic topology brings inevitable rotation latency. The lately proposed Sirius [14] provides the abstraction of a high-radix switch which can reduce the configuration delay from milliseconds level to nanosecond-granularity, which makes it possible to bring about an efficient and dynamic network design.

Conclusion

In this article, we address the importance of constructing a low-latency deterministic network. We group the existing schemes into three development stages and make detailed analyses and summaries separately. Then, we extract their features and compare them in terms of deployment complexity. For encouraging future studies and discussions, we analyze the unsolved problems and present opportunities for future research.

Acknowledgments

This work was supported in part by the National Key RD Program of China (No. 2018YFB1800303), in part by the NSFC Project (No. 62132009), in part by the Research Cooperation Project between Huawei Tech and Tsinghua Shenzhen International Graduate School (No. YBN20200251123) , and in part by the Social Science Fund of Shenzhen (No. SZ2020D009).

References

[1] P. X. Gao *et al.*, "pHost: Distributed Near-Optimal Data center Transport Over Commodity Network Fabric," *Proc. 11th ACM Conf. Emerging Networking Experiments and Technologies (CoNEXT)*, 2015, pp. 1–12.

[2] B. Montazeri *et al.*, "Homa: A Receiver-Driven Low- Latency Transport Protocol Using Network Priorities," *Proc. ACM SIGCOMM 2018 Conf.*, 2018, pp. 221–35.

[3] Y. Zhu *et al.*, "Congestion Control for Large-scale RDMA Deployments," *Proc. ACM SIGCOMM 2015 Conf.*, 2015, pp. 523–36.

[4] W. Cheng *et al.*, "Re-Architecting Congestion Management in Lossless Ethernet," *Proc. 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 19–36.

[5] R. Mittal *et al.*, "TIMELY: RTT-Based Congestion Control for the Data center," *Proc. ACM SIGCOMM 2015 Conf.*, 2015, pp. 537–50.

[6] G. Kumar *et al.*, "Swift: Delay is Simple and Effective for Congestion Control in the Data center," *Proc. ACM SIGCOMM 2020 Conf.*, 2020, pp. 514–28.

[7] Y. Li *et al.*, "HPCC: High Precision Congestion Control," *Proc. ACM SIGCOMM 2019 Conf.*, 2019, pp. 44–58.

[8] V. Shrivastav *et al.*, "Shoal: A Network Architecture for Disaggregated Racks," *Proc. 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, 2019, pp. 255–70.

[9] I. Cho *et al.*, "Credit-Scheduled Delay-Bounded Congestion Control for Data centers," *Proc. ACM SIGCOMM 2017 Conf.*, 2017, p. 239–52.

[10] M. Handley *et al.*, "Re-Architecting Data center Networks and Stacks for Low Latency and High Performance," *Proc. ACM SIGCOMM 2017 Conf.*, 2017, pp. 29–42.

[11] S. Hu *et al.*, "Aeolus: A Building Block for Proactive Transport in Data centers," *Proc. ACM SIGCOMM 2020 Conf.*, 2020, pp. 422–34.

[12] W. M. Mellette *et al.*, "RotorNet: A Scalable, Lowcomplexity, Optical Data Center Network," *Proc. ACM SIGCOMM 2017 Conf.*, 2017, pp. 267–80.

[13] W. M. Mellette *et al.*, "Expanding Across Time to Deliver Bandwidth Efficiency and Low Latency," *Proc. 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 1–18.

[14] H. Ballani *et al.*, "Sirius: A Flat Data center Network With Nanosecond Optical Switching," *Proc. ACM SIGCOMM 2020 Conf.*, 2020, pp. 782–97.

[15] J. Perry *et al.*, "Fastpass: A Centralized" Zero-Queue" Data center Network," *Proc. ACM SIGCOMM 2014 Conf.*, 2014, pp. 307–18.

Biographies

Feixue Han received her B.S. degree in 2018 from BeiHang University. Currently she is a graduate student in Department of Computer Science and Technology, Tsinghua University. Her current research interests include congestion control and traffic scheduling.

Mowei Wang received his B.Eng. degree in communication engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently working toward his Ph.D. degree in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include data center networks and datadriven networking.

Yong Cui received his B.E. and Ph.D. degrees in computer science and engineering from Tsinghua University, China, in 1999 and 2004, respectively. He is currently a Full Professor with the Computer Science Department, Tsinghua University. His major research interests include mobile cloud computing and network architecture.

Qing Li received his B.S. degree from Dalian University of Technology, Dalian, China, the Ph.D. degree from Tsinghua University, Beijing, China. He is currently an associate researcher at Peng Cheng Laboratory, Shenzhen, China. His research interests include reliable and scalable routing of the Internet, intelligent self-running network, edge computing, and so on.

Ru Liang received his BE. degrees in communications and electronics systems from Dalian University of Technology, China, in 1997. He is currently an expert with Network Product Line, Huawei Tech. Co., Ltd., His research interests include network protocols and network architecture.

Yashe Liu received the Ph.D. degrees in communications and electronics systems from Xidian University, China, in 1998. He is currently a Senior Experts with Corporate Technology Strategy Department, Huawei Tech. Co., Ltd., His research interests include hardware design of broadband networks, E2E architecture and performance evaluation of data center networks.

Yong Jiang received his B.S. degree and the Ph.D. degree from Tsinghua University, Beijing, China, both in computer science and technology. He is currently a full professor at the Tsinghua Shenzhen International Graduate School. His research interests include the future network architecture, the Internet QoS, network function virtualization, and so on.