Transmission Timing on Optical Data-Center RotorNet for Reduced Cost, Energy and Latency

Yitzhak Birk
Viterbi Faculty of Electrical Engr.
Technion – Israel Inst. of Technology
Haifa, Israel
birk@ee.technion.ac.il

Tamir Friedman
Viterbi Faculty of Electrical Engr.
Technion – Israel Inst. of Technology
Haifa, Israel
tamir.fri@campus.technion.ac.il

***Abstract*—RotorNet comprises switches that provide direct all-to-all connectivity among data-center nodes by cycling among all cyclic shift permutations. Combined with two-hop routing, whereby packets traverse the optical network twice, they provide high throughput regardless of the traffic pattern and with no need for centralized control. Although being a circuit-switched optical interconnect intended primarily for latency-insensitive traffic, reducing latency saves energy and buffer space, and is thus important. We show that latency can be reduced by properly timing the transmissions by the sender, and offer insights along with some useful results.**

Keywords—Data-center networks; RotorNet, transmission scheduling, Optical Data-Center Networks.

I. INTRODUCTION

A. Background

Data-centers are communication intensive, and communication has become a substantial cost and power component. The desire for high throughput in conjunction with increasing equipment density has also given rise to a data-rate density (e.g., the total data rate that can be sent out of or into a 1U blade in a rack) bottleneck. This has caused demand for ever increasing line speed (per port), even when no single application needs such data rates. The high data rates, in turn, have forced the use of optical fibers for connections among (electronic) switches, and Electrical-Optical conversions must take place at the ends of each link, increasing cabling cost and power consumption.

A traditional data-center communication architecture includes compute racks that are connected to "top of rack" (ToR) switches, which are in turn interconnected via one or more additional layers of switches. The originally most prominent topology, commonly used for high performance computing (HPC), is the Fat Tree [1]. However, recent research discovered the advantages of using flat, yet scalable, static topologies such as Jellyfish [2] and Xpander [3]. These topologies do not dedicate special graph nodes for routing; instead, ToR switches also serve as relays. All these approaches route packets from source to destination over multi-hop paths via electronic packet switches. They all incur per-hop costs of E-O-E conversion, some packet processing and buffering. Moreover, the maximum number of hops in a fixed topology is bounded from below by its graph diameter which, according to Moore's bound [4], is:

$$diameter \geq \lceil \log_{degree^*-1}|V| - \log_2 3 \rceil, \qquad (1)$$

where *degree* (>2) is the number of ports per node.

In practice, not all traffic is latency-sensitive: a very large flow, e.g., copying or moving a large amount of data from one node to another, cannot benefit from very low packet latency. It has been proposed to view traffic as comprising "mice" flows that are latency sensitive, and "elephant" flows that only care about throughput. There is even evidence [5] that in some cases most of the messages belong to mice flows, but most of the data are sent as part of elephant flows.

The per-flow equipment and energy cost along with the above observation gave rise to the idea of hybrid networks, comprising a circuit-switched optical network in addition to the conventional packet switched one. The elephant flows are sent over the optical network, which provides point-to-point circuits via simple, inexpensive "dumb" optical switches that are configured on demand. This reduces the number of E-O-E conversions along a path, and eliminates much of the buffering and processing, thereby reducing the amount of energy per bit and saving equipment. The conventional network serves the mice as well as control traffic for the optical network, and can be less expensive because it carries less traffic.

The main drawback of this hybrid network [6] approach is that it entails a hard partitioning of network resources (ToR switch ports and fibers). Other issues that arise are changes that must be made to the conventional switches, as well as the control mechanism, identification of "elephant" flows, etc. We do not discuss these, but there is wide agreement that for all but huge flows, the control traffic and centralized scheduling of the optical network may be prohibitively complex.

B. *RotorNet* [7]

Recent work by W.M. Mellette et al introduced RotorNet [7], a scalable low-complexity optical datacenter network. It comprises one or more Rotor Switches that are cycled through predetermined sets of permutations, independent of traffic requirements. The originator of traffic is aware of the schedules, and decides when to place packets on the network. In its simplest form, depicted in Fig. 1, RotorNet comprises a single $N \times N$ optical switch capable of providing all cyclic shift permutations, and it is cycled among them continuously.

This work was supported in part by the Israel Innovation Authority of Israel's Ministry of Economy through the PetaCloud Consortium.

978-1-5386-6378-3/18/$31.00 ©2018 IEEE

RotorNet, even in its simplest form, provides direct connectivity between any two end nodes, but does so only during a small, possibly tiny, fraction of time, making it impractical for high-throughput flows. To solve this, RotorNet employs a load-spreading technique known as Valiant Load Balancing (VLB) [8]: Two hops (possibly more, but two suffice) are allowed. With this, a sender can send at any time packets to intermediate nodes (that are themselves also end-nodes), and each intermediate node forwards the packets it receives to the final destination when the rotor setting provides it a direct connection to the destination.

Since both the originator and destination nodes are connected to all other nodes at some point in the cycle, it can readily be seen that any source can communicate with any destination at line speed. Moreover, regardless of traffic pattern except for bottlenecks in the final destination, total throughput can be as high as one-half of the theoretical maximum (due to the consumption of two transmissions and receptions per packet). No central control, reservations, etc. are required, the load is always balanced, and the optical switches are simple, making RotorNet worthy of detailed exploration.

RotorNet is composed of three functional layers: 1) the fiber infrastructure connectivity, i.e. the set of permutations offered by each switch; 2) the cyclic schedules; and 3) a routing algorithm, which in RotorNet simply means, for a given infrastructure and rotor schedules, the decision when to transmit a message on each hop, given its source, destination, and creation time. This paper focuses on the routing, with an aim to reduce latency.

C. *Routing for low latency*

In RotorNet, the timing of transmissions is either obvious (in the final hop there is no choice) or unimportant for throughput, and if used for latency-insensitive traffic then latency is seemingly not a consideration either. However, we claim that latency is important, at least for the following reasons:

- **Energy per bit.**
  The total energy per bit (from source to destination) can be expressed as
  $$E_{bit} = n_{hops} \cdot E_{hop} + latency \cdot P_{memory} \qquad (2)$$
  where E_{hop} includes transmission, E-O-E conversion and any processing in a conventional switch, intermediate node, etc., and P_{memory} is the power consumed by memory in which the bit is buffered. For a given number of hops, latency is thus the controllable contributor to the energy consumption.

- **Buffer size (and cost).**
  During the time that a packet spends in the network, it must be in a buffer. According to Little's law [8], the total amount of buffer memory equals the product of throughput and mean latency, so reducing latency permits smaller buffers.

- **Applicability.**
  Reducing latency may increase the fraction of traffic that can use the RotorNet, thereby further saving power and cost of conventional switches.

In this paper, we begin to explore latency reduction via timing of transmissions. We consider a single rotor switch interconnecting all N end nodes, each connected via a single port. The rotor is cycled through all cyclic shift permutations, in order. While mostly focusing on a single packet, we assume the use of multiple hops for throughput reasons; we derive the optimal transmission and forwarding times for multiple hops, and plot the latency as a function of creation time. Since increasing the number of hops eventually increases energy per bit, limits throughput, and better results can be achieved with a static topology, we focus on two and three hops. Finally, we provide some insights and heuristics for dealing with potential contention at an intermediate node, and briefly address the case of a long flow whose required throughput is nonetheless only a fraction of line speed.

II. Definitions

Let N be the number of end nodes, numbered $0, 1, \ldots, N-1$. We use the congruence modulo N equivalence relation, denoted $a \equiv b$. A rotor is an additive mapping that transfers packets from source node i to destination node j if and only if $j - i \equiv \Delta[t]$, where t indicates the time slot in which the operation occurs, and $\Delta[t]$ is a periodic function with period N. We call a rotor *linear* if it has an affine delta function $\Delta[t] \equiv A \cdot t + B$, with constant integer parameters A (which is relatively prime to N) and B (i.e., a constant increase in the amount of shift between consecutive time slots). For mean latency calculation, we introduce the discrete uniform random variable $T \sim U\{0, \ldots, N-1\}$, representing randomization of the time slot in which the packet was created in the source. We define an h-hop path by the waiting times $(k_0, \ldots, k_{h-1})$ in the intermediate nodes, indexed from k_0 for the waiting at the source, to k_{h-1} for the waiting at the last intermediate node. All these waiting times must be non-negative, and we take them to be integers. The latency of a packet sent over a given h-hop path is $L = \sum_{j=0}^{h-1} k_j$; waiting time at the source is included.

Remark: j is used in some places to denote destination nodes and in others – to denote indexed hop numbers. The meaning is self-evident in each case.

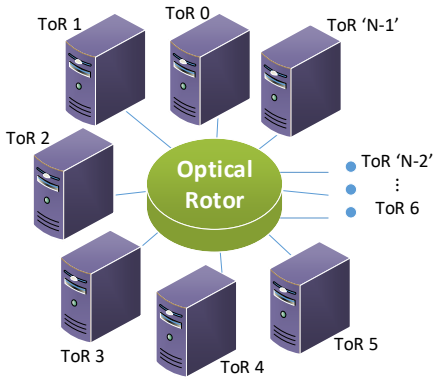

Fig. 1. A single-rotor RotorNet.

III. MINIMUM-LATENCY ROUTING

In the following sections, we design and analyze low-latency-routing algorithms for different representative common cases.

We begin by assuming that a packet can traverse its entire multi-hop path during a single time slot. This is not an unreasonable assumption, as slot duration may be substantially longer than packet transmission time for reasons such as time coordination overhead, burst synchronization time, and switching time. We also consider the restriction to at most a single hop per time slot, with which the waiting times at the intermediate nodes (other than the source) must be positive integers ($\forall j\ s.t.\ 1 \le j \le h: k_j \ge 1$). There are more complex time models, like several (but not all) permissible hops per time slot. Nevertheless, in this paper we only consider only the two aforementioned extremes.

In the case of a maximum throughput (line speed) flow, we must use the VLB technique and transmit packets at all times. As mentioned earlier, each elephant flow generator inside a rack may pre-order its flow so as to cause its packets to arrive in-order, assuming a deterministic schedule and no contention. Therefore, even though the packet sending order does not have to be FIFO, we consider it as such, and send the packets as soon as they are created. Therefore, we set $k_0 = 0$.

A. A Single Packet

For a single packet and assuming no contention, we do not have to use VLB since there is no load in the source, and may therefore delay the first-hop transmission (hold the packet at the source).

Consider without loss of generality a packet created at end-node *i=0* at time *t=0*, whose destination is node *D*. For facility of exposition, suppose that the end nodes are arranged along a circle, and the shifts are in the clockwise direction. Suppose that the rotor provides a shift by *s* at *t=0*. Transmission of the packet at *t=0* over *h* hops would land it at node $s \cdot h\ (mod\ N)$. Let *d* denote the "clockwise" distance from this landing point to the destination *D*. Delaying the packet (only) at the source closes this gap, *d*, in steps of *h* per time slot worth of additional delay. This happens because the shift incrementation of the rotor applies to all hops. Similarly, delaying it in the first intermediate node would result in a stride of *h-1*, since the incrementation applies only from the second hop, onwards. Delaying it in later intermediates leads to linearly smaller strides, and in the last intermediate node it would result in a stride of *1*, caused by the last hop. Latency (the sum of the holding times) is minimized by using the maximal stride, but we must not overshoot. Therefore, if *D<h*, we delay the packet for only one slot, only at the appropriate intermediate node. Else, we delay it at the source for the maximum number of slots that does not result in an overshoot, then transmit it; it arrives at the next node with a remaining gap of at most *h*, so we continue as in the first case. It follows that minimum latency always entails delaying the packet at the source if necessary, plus at most an additional single-slot delay at a single intermediate node. We now state and prove this formally.

Proposition 1: The lowest latency *h*-hop path in a RotorNet with a single linear rotor and no waiting time limitations ($\forall j: k_j \ge 0$), is given by the waiting times:

$$k_0 = \lfloor C/h \rfloor$$
$$\forall j \ge 1: k_j = \begin{cases} 1, & if\ \ j = h - (C\ mod\ h) \\ 0, & otherwise, \end{cases}$$

where $C = (A^{-1} \cdot (\Delta - h \cdot B) - h \cdot t)\ mod\ N$

This path acheives the following minimum latency:

$$L_{min,h} = \lceil C/h \rceil$$

With a mean (over creation time) of:

$$E\left[L_{min,h}\right] = \frac{N}{2h}$$

Proof:

The delta for a given *h*-hop path is:

$$\Delta \equiv \underbrace{(A \cdot (t + k_0) + B)}_{first\ hop} + \underbrace{(A \cdot (t + k_0 + k_1) + B)}_{second\ hop} + \cdots + \underbrace{\left(A \cdot \left(t + \sum_{j=0}^{h-1} k_j\right) + B\right)}_{last\ hop}$$
$$\equiv A \cdot \left(h \cdot t + \sum_{j=0}^{h-1} (h-j) \cdot k_j\right) + h \cdot B$$

Since the rotor parameter A is relatively prime to N, we can use its multiplicative inverse:

$$\sum_{j=0}^{h-1} (h-j) \cdot k_j \equiv A^{-1} \cdot (\Delta - h \cdot B) - h \cdot t \equiv C$$
$$when: 0 \le C \le N-1$$

Assuming that $L_{min} < \frac{C+N}{h}$, we bound the left-hand side of the equation: $0 \le \underbrace{\sum_{j=0}^{h-1} (h-j) \cdot k_j}_{LHS} \le h \cdot L_{min} < C + N$

This bound is the reason we can write the residue equivalence as an exact equality. We use it in the latency expression: $C = \sum_{j=0}^{h-1} (h-j) \cdot k_j = h \cdot L_{min} - \sum_{j=1}^{h-1} j \cdot k_j \quad \Rightarrow$ $L_{min} = \frac{1}{h} \cdot \left(C + \sum_{j=1}^{h-1} j \cdot k_j\right)$

Minimum latency is attained when all of the waiting time is in k_0, since it does not influence the current expression. An additional single-slot waiting time may be added afterwards in order to make the expression integer, without residue.

Mean latency in this case is the expected value over the parameter *C*, along all possible time slots:

$$T \sim U\{0, \dots, N-1\}$$
$$C = (A^{-1} \cdot (\Delta - h \cdot B) - h \cdot T)\ mod\ N$$
$$\Rightarrow C \sim U\{0, \dots, N-1\}$$

This last observation is true if the number of hops h is relativly prime to *N*; otherwise, it is uniform over a sparser set of integers. The calculated expected value is:

$E\left[L_{min,h}\right] = \frac{1}{h} \cdot \left(E[C] + \sum_{j=1}^{h-1} j \cdot E\left[k_j\right]\right) = \frac{1}{h} \cdot \frac{N}{2} = \frac{N}{2h}$ ■

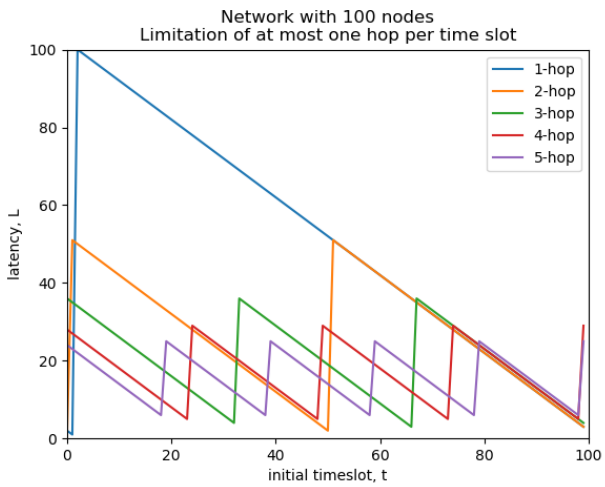

Fig. 2. Minimum latency of a packet sent between source-destination pair with $\Delta = 1$ versus its creation time. It is a sawtooth function with period N/h, maximum N/h, and a mean of $0.5N/h$.

Fig. 2 depicts the minimum latency versus creation time for a packet that needs to travel from node 0 to node 1, assuming that at time 0 the rotor shifts by one. Plots are shown for 1-5 hops. For any given number of hops, we observe a sawtooth pattern, whose period and maximum (and thus mean) are all inversely proportional to the number of hops. If any number of hops is permitted up to a certain maximum, then at each creation time we take the minimum among the delays for the permissible numbers of hops.

Proposition 2: The lowest latency h-hop path in a RotorNet with a single linear rotor and a limitation of at most one hop per time slot ($\forall j \geq 1: k_j \geq 1$), is given by the waiting times:

$$k_0 = \lfloor C/h \rfloor$$
$$\forall j \geq 1: k_j = \begin{cases} 2, if\ \ j = h - (C \bmod h) \\ 1, \qquad\qquad else \end{cases}$$
$$where\ C = \left(A^{-1} \cdot (\Delta - h \cdot B) - \binom{h}{2} - h \cdot t\right)\ mod\ N$$

This path acheives the following minimal latency:

$$L_{min,h} = \lceil C/h \rceil + h - 1$$

With an expected value, over changing time slots, of:

$$E\left[L_{min,h}\right] = \frac{N}{2h} + h - 1$$

Proof:

A reduction can be made from this case to the previous one using the variable transformation $\forall j \geq 1: k_j^* = k_j - 1$; we substitude the congruence relation derived from the delta constrain for the original wating times $(k_0, \dots, k_{h-1})$:

$$\sum_{j=0}^{h-1} (h-j) \cdot k_j^* \equiv \sum_{j=0}^{h-1} (h-j) \cdot k_j - \sum_{j=1}^{h-1} h - j$$
$$\equiv A^{-1} \cdot (\Delta - h \cdot B) - h \cdot t - \binom{h}{2} \equiv C$$
$$when: 0 \leq C \leq N-1$$

From proposition 1, we know that the minimum latency path for this case is:

$$k_0 = \lfloor C/h \rfloor$$
$$\forall j \geq 1: k_j = 1 + k_j^* = \begin{cases} 2, if\ \ j = h - (C \bmod h) \\ 1, \qquad\qquad else \end{cases}$$
$$\Rightarrow L_{min,h} = \lceil C/h \rceil + h - 1$$

Finally, the expected value changes by a constant value.

$$E\left[L_{min,h}\right] = \frac{N}{2h} + h - 1 \qquad \blacksquare$$

B. Full Throughput Flow

In the case of a full-throughput flow, we have some additional limitations. First, we must use VLB and transmit packets continuously during all time slots. An observation from the sawtooth function is that within each "tooth", the latency is decreasing with first-transmission time. Packets transmitted within a single "tooth" therefore arrive at the destination in reverse order. Assuming no contention, the originator of an elephant flow can send its packets in clusters, in reverse order within each cluster, causing intra-cluster in-order arrivals at the destination.

A second limitation arises from the desire to prevent self-contention. A first hop is always from the source, a last hop is always towards the destination, and intermediate hops are always between intermediate end nodes. Therefore, self-contention may occur if there is more than one intermediate hop. Hence, to be on the safe side with full throughput flows, we would limit the number of hops to no more than three.

Proposition 3: The lowest latency 3-hop path in a RotorNet with a single linear rotor, zero waiting time at source ($k_0 = 0$), and no other waiting time limitations ($k_1, k_2 \geq 0$), is given by the waiting times:

$$k_0 = 0\ ;\ \ k_1 = \lfloor C/2 \rfloor\ ;\ \ k_2 = (C\ mod\ 2)$$
$$where\ C = (A^{-1} \cdot (\Delta - 3 \cdot B) - 3 \cdot t)\ mod\ N$$

This path acheives the following latency:

$$L_{min,3^*} = \lceil C/2 \rceil$$

With an expected value, over changing creation time slots:

$$E\left[L_{min,3^*}\right] = \frac{N}{4}$$

Proof:

By analogy to earlier proofs: $\Delta \equiv A \cdot (3 \cdot t + 2 \cdot k_1 + k_2) + 3 \cdot B \Rightarrow (2 \cdot k_1 + k_2) \equiv A^{-1} \cdot (\Delta - 3 \cdot B) - 3 \cdot t \equiv C\ ;\ when\ 0 \leq C \leq N-1$. Under the assumption: $2 \cdot k_1 + k_2 < C + N$, we can write the exact equality: $C = 2 \cdot k_1 + k_2 = 2 \cdot L_{min,3^*} - k_2 \Rightarrow L_{min,3^*} = \frac{1}{2} \cdot (C + k_2) \Rightarrow L_{min,3^*} = \lceil C/2 \rceil; k_1 = \lfloor C/2 \rfloor; k_2 = C\ mod\ 2; E\left[L_{min,3^*}\right] = \frac{N}{4}$ $\blacksquare$

A third limitation that may occur is when the network is in high total utilization. With h permissible hops, each packet requires up to h transfers over the network. Accordingly, the total throughput possible with an h-hop path policy is at most $\frac{1}{h}$ of the network capacity. If we use 2-hop paths, we are left with only one coefficient (k_1) and without any degrees of freedom how to choose it. Therefore, the only option the packet has is to be sent to an intermediate node as soon as it is created, and then it has to wait until it gets a direct connection to its destination.

C. Partial Throughput Flow

The case of a partial-throughput flow, that consumes a fraction $p<1$ of line-speed, is somewhere in the middle between a single packet and a full-throughput flow. Now, we do not have to send in every time slot; however, we do need to utilize a fraction p of time slots during a full rotor cycle. According to the sawtooth plots created for single packet latency, we can choose to transmit in each rotor cycle only over the N/p lowest latency paths, i.e., send packets in bursts during time slots corresponding to the low points in every sawtooth.

IV. CONCLUSIONS

We formulated the relationship between energy, cost and latency in RotorNet, revealing that latency reduction is important even for traffic that is insensitive to latency. For a single rotor and a single packet, we derived the latency minimizing transmission timing and the resulting latency. For long flows, we provided important insights and initial directions.

Directions for future research include extension to multiple rotors, joint consideration of the schedule of each rotor, inter-rotor "phase" differences, and transmission timing. Also, accommodation of contention. Yet another topic is incorporating the trade-off between number of hops and latency so as to derive the minimum energy transmission timing for each message based on its source, destination, creation time and the rotor schedules.

REFERENCES

[1] C. E. Leiserson, “Fat-trees: Universal networks for hardware-efficient supercomputing,” in IEEE Transactions on Computers, 1985.

[2] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, “Jellyfish: Networking data centers randomly,” in NSDI ’12. USENIX, 2012.

[3] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira, “Xpander: Towards Optimal-Performance Datacenters,” in Proc. of the 12th International on Conference on emerging Networking EXperiments and Technologies, pp. 205–219, 2016.

[4] M. Miller and J. Siran, “Moore graphs and beyond: A survey of the degree/diameter problem,” Electronic J. of Combinatorics, vol. 14, 2005.

[5] L. Guo and I. Matta, “The War Between Mice and Elephants,” in Proc. IEEE ICNP ’01, Nov. 2001.

[6] J. Bowers, A. Raza, D. Tardent, and J. Miglani, “Advantages and control of hybrid packet optical-circuit-switched data center networks,” in Photonics in Switching, pp. PM2C–4. Optical Soc. of America, 2014.

[7] William M Mellette, R. Mcguinness, A. Roy, A. Forencich, G. Papen, A. C. Snoeren, and G. Porter, “RoterNet: A Scalable, Low-complexity, Optical Datacenter Network,” in Proc. ACM SIGCOMM 2017.

[8] L. G. Valiant, “A scheme for fast parallel communication,” SIAM J. Comput., vol. 11, no. 2, pp. 350-361, May 1982.

[9] J. D. C. Little, ”A proof for the queuing formula: $L = \lambda W$,” Operations Research, vol.9, no.3, pp. 383-387, June 1961