High Network Utilization Load Balancing Scheme for Data Centers

Yang Chen and Jie Wu
Department of Computer and Information Sciences, Temple University, USA
Email: {yang.chen, jiewu}@temple.edu

***Abstract*—We propose a high network utilization CONGestion-Aware load balancing approach, Multi-hop CONGA. It aims at improving the performance of handling traffic asymmetry, which happens occasionally in data centers. It leverages regular Clos topology and overlays for virtualization. In this paper, we schedule at the unit of flowlet, and make routing decisions on the first packet of each flowlet, based on the current global network situation, which is obtained by using a feedback mechanism. Moreover, in order to further balance loads, Multi-hop CONGA provides an alternative of a two-hop routing path by leveraging relative leisure links to relieve local traffic pressure, when all direct one-hop paths are heavily-loaded. We evaluate Multi-hop CONGA with extensive simulations, and compare it with several popular methods under different traffic conditions. Results demonstrate feasibility and efficiency of Multi-hop CONGA.**

***Index Terms*—High network Utilization; Load balancing; Multi-hop.**

I. Introduction

In recent years, data centers or large clusters of servers have been increasingly employed in university, enterprise and consumer settings to run a variety of applications, ranging from real time traffic and power monitoring to web services. Many kinds of design principles have been presented aiming at obtaining high performance and maintaining load balancing in data centers.Distributed routing methods such as ECMP [1], Flare [2], Back Pressure [3] and MPTCP [4, 5] are responsive, which is an essential property for handling data centers' volatile and bursty traffic. In ECMP, when a flow arrives at the switch, its path is decided locally based on hashed values of multiple equal-cost paths. However, its load balancing performance is unstable because of hash collisions and totally-local decisions. Another local congestion-aware approach, LocalFlow [6, 7], independently acts on each local switch to divide flows to proper paths. Flare [2] first schedules paths in the unit of bursts of packets, which serves as a finer granularity for load balancing. Moreover, the destination does not need to reorder the packets. Our multi-hop scheme is related to Back Pressure [3], which arranges data in directions that maximize the differential backlog between neighboring nodes. In other words, it schedules traffic to the shortest length queue, no matter where the packet's destination is. It is also a local obstruction information-based scheduling mechanism. MPTCP [4, 5] splits flows into subflows, and schedules each subflow into a less-congested path to balance loads and control congestion. However, it is host-based, which is hard to deploy and may be trapped into incast problems [8].

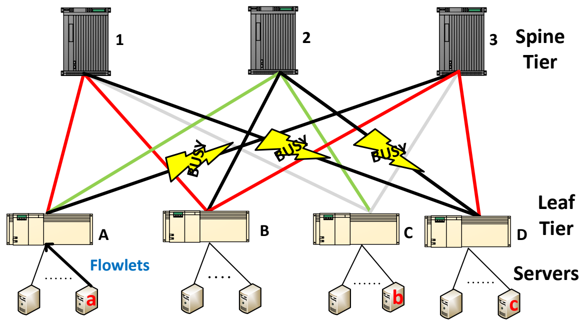


Fig. 1: Multi-hop CONGA Switching Scheme Example.

There also has been a great deal of research related to data center performance issues. Hedera [9] provides centralized flow arrangement, as well as B4 [10] and SWAN [11]. They use a central controller to compute good paths for detected large flows, and install scheduling information on switches. However, because of volatile and bursty traffic in data centers, it is too slow for centralized schemes to properly respond to congestion. Another type of approaches focuses on scheduling granularity. For example, DRB [12] presents a per-packet round-robin based routing protocol to implement load balancing, and achieve low latency. In traditional routing algorithms [1, 12–15], the granularity is always per-flow, where the disadvantage lies in low network utilization and long transmission latency.

A recent proposed mechanism CONGA [16] catches our attention. It schedules paths based on a real-time fabric congestion situation, obtained by feedbacks from remote switches. But CONGA limits packets to a one-hop routing path. Specifically, the one-hop path means traversing only one spine switch to its destination. However, traffic is likely to be heavily-unbalanced in data centers, for example, because of bursts in just a few congested links such as black lines, shown in Figure 1. In this example, when Server a wants to send packets to Server b, the green one-hop path is chosen. But if the destination is Server c, there is no available congestion-free one-hop path, and CONGA will suffer severe transmission delays. However, we notice that a two-hop red line path, which detours at the leaf switch B, is able to avoid congested links as well as share loads to relieve local traffic pressure.

Inspired by the above observation, we present a more flexible scheduling method, Multi-hop CONGA, to fully balance traffic and increase network utilization. We select the best of previous work CONGA [16], and combine with multi-hop routing techniques. Under the normal traffic pattern, Multi-hop CONGA performs as CONGA, but when bursty flows or load spikes come, it ingeniously bypasses congested links by applying two-hop paths. It further improves utilization by exploiting idle bandwidth, though to some extent, it leverages extra resource and adds more transmission burden.

In summary, our major contributions are:

1) We review the majority of recent academic works on load balancing for data centers. Main advantages and disadvantages of these methods are summarized.
2) We propose a distributed high network-utilization mechanism, Multi-hop CONGA. It is flexible to bursty traffic, easily-deployable to current network topology, highly efficient to congestion, and requires no modification to end-hosts.
3) We evaluate Multi-hop CONGA and compare it with several popular load balancing approaches in simulations. It proves its effectiveness in the regular traffic pattern, and obvious advantages in unbalanced situations with bursty traffic or link failures.

Our paper is organized in the following way: Section 2 presents detailed insights of Multi-hop CONGA, signposted by pertinent examples and concrete explanations. In section 3 Multi-hop CONGA will be analyzed theoretically to illustrate its advantage over CONGA. Section 4 extensively evaluates Multi-hop CONGA in packet-level simulations and demonstrates its better performance than other mechanisms.

II. Multi-hop CONGA Architecture

Multi-hop CONGA routes flowlets to the hand-picked paths based on global congestion feedback information. A flowlet [2] is bursts of packets belonging to a flow. Scheduling decisions are made by the switch on the first packet of each new flowlet, and then stored in its Flowlet Table for following packets to use afterwards. In a desirable one-hop path, a packet is initially transfered uplink towards one spine switch, and then downlink to the leaf switch, which its destination server connects to. If there is no such a path, we will schedule a two-hop alternative to detour away from crowded links. We trade a longer transmission path for better load balancing by leveraging leisure bandwidth resources.

We consider a set of server users U, where each user u holds a set of flows F_u, and each flow $f_u^k \in F_u$. The server u connects to the source leaf switch $s_{f_u^k}$ and the flow f_u^k's destination connects to the destination leaf switch $d_{f_u^k}$. A one-hop path is represented as $p(s_{f_u^k}, d_{f_u^k}) \in P_{f_u^k}$, while a two-hop path as $p'(s_{f_u^k}, i_{f_u^k}, d_{f_u^k}) \in P'_{f_u^k}$. ($P(f_u^k)$ and $P'(f_u^k)$ are the general collection of paths for user u's flow f_u^k, and $i_{f_u^k}$ is the "transfer station" leaf switch.) Denote the congestion metric of one path as $c(p(f_u^k))$, which introduces the maximum single link delay along the path. Set the one-hop to two-hop path switching threshold as η, and conversely σ. Our scheme chooses the path with $\min(c(p(s_{f_u^k}, d_{f_u^k})), c(p'(s_{f_u^k}, i_{f_u^k}, d_{f_u^k})))$ for every available path $p \in P$ and $p' \in P'$. We display details of the scheduling procedure in Algorithm 1.

Algorithm 1 Multi-hop CONGA Routing

```
Input:  The packet's Source $s_{f_u^k}$ and Destination $d_{f_u^k}$ fields,
        source switch 's Congestion-Table $T$, one-hop to two-hop
        threshold $\eta$, two-hop to one-hop threshold $\sigma$;
Output: Scheduled routing path (one-hop or two-hop)
        $\mathcal{PATH}$ goes from $s_{f_u^k}$ to $d_{f_u^k}$;
1:  Use T to compute $c(p(f_u^k))$ for each one-hop path from
    $s_{f_u^k}$ to $d_{f_u^k}$;
2:  if $\exists p \in P$ satisfying $c(p(s_{f_u^k}, d_{f_u^k})) < \eta$ then
3:      select the path $\mathcal{P}$ with $\min(c(p(s_{f_u^k}, d_{f_u^k}))), \forall p \in P)$;
4:  else
5:      for each possible internal leaf switch $i_{f_u^k}$ do
6:          ${\mathcal{P}'}_{i_{f_u^k}} = \min(c(p'(s_{f_u^k}, i_{f_u^k}, d_{f_u^k}))), \forall p' \in P')$
7:      select $\mathcal{PATH} = \min(c(p'), \forall p' \in \mathcal{P}')$, $\mathcal{P}'$ as the
        general collection of ${\mathcal{P}'}_{i_{f_u^k}}$
8:      if the selected two-hop path $c(\mathcal{PATH}) > \sigma$ then
9:          go back to find the least crowded one-hop path
            $\mathcal{PATH}$ with $\min(c(p(s_{f_u^k}, d_{f_u^k}))), \forall p \in P)$;
10: return the selected path $\mathcal{PATH}$;
```

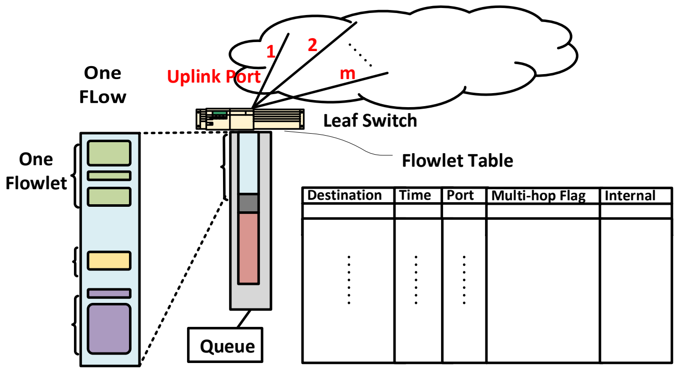
A. Scheduling Granularity

In one hand, operating at the granularity of packet bears too much computation complexity, and needs to reorder arrived packets, though it can route each packet to the optimal path. On the other hand, routing in units of flow saves trouble in reordering, however, selected paths are possibly overshot, which would cause degraded performance in throughput. To the best of our knowledge, flowlet switching is a compromise. Flowlets are bursts of the same-flow packets, and first introduced in [2]. Packets are carefully partitioned by an interval μ, which is necessarily larger than the delay between parallel paths.

Flowlet detection is based on the Flowlet Table shown in Figure 2. It contains five columns: Destination, Time, Port, Multi-hop Flag and Internal. Destination keeps the destination information; Time is the arrival time of the latest packet of the flowlet, and is updated everytime a packet arrives; Port is the desired routing path's sending port; Multi-hop Flag and Internal record the two-hop path corresponding information. When a packet arrives, its destination's IP address is hashed into one of the table's entries. If the difference between its arrival time and Time value in the Flowlet Table is smaller than μ, the flowlet is still active, and the packet just refers to the Port value to decide its next hop. Otherwise, the incoming packet starts a new flowlet, whose path needs to be rescheduled, specified in Section 2.3. Then the entry's related values refresh to the latest version.

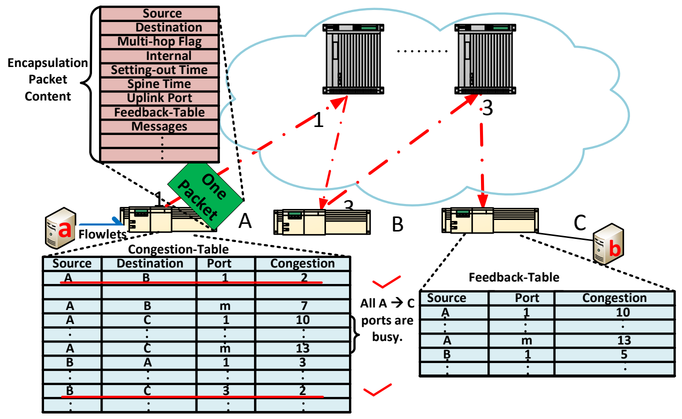
B. Congestion Information Metrics

Multi-hop CONGA is a global congestion-aware mechanism to deal with asymmetry such as bursty flows and link failures. Each path's load condition is measured by the packet's



Destination	Time	Port	Multi-hop Flag	Internal
⋮	⋮	⋮		

Fig. 2: Extract of a Chunk of a Leaf Switch



Congestion-Table

Source	Destination	Port	Congestion
A	B	1	2
A	B	m	7
A	C	1	10
⋮	⋮	⋮	⋮
A	C	m	13
B	A	1	3
⋮	⋮	⋮	⋮
B	C	3	2
⋮	⋮	⋮	⋮

Feedback-Table

Source	Port	Congestion
A	1	10
⋮	⋮	⋮
A	m	13
B	1	5
⋮	⋮	⋮

Fig. 3: Multi-hop CONGA System Diagram

maximum transmission link delay, proportional into 4 bits. Every leaf switch i supports three values: Tag, Congestion-Table,, and Feedback-Table. Tag shows whether the Feedback-Table has refreshed; Congestion-Table holds traffic information metrics, which is updated by the feedback of all other switches; Feedback-Table conveys the latest congestion condition for paths, whose destination connects to the leaf switch i. Feedback-Table contains three properties: Source leaf switch (source as well as destination leaf switches if in Congestion-Table), Uplink port number, and Congestion metric. The tables' examples are exhibited in Figure 3. With n leaf switches and each has m available uplink ports, the Feedback-Table has $m * n$ entries, while the Congestion-Table holds $m * n^2$ entries to record the global traffic state.

C. Routing Path Decision

To make a routing decision, a leaf switch first checks its Congestion-Table whether a one-hop path is available. If there exists one metric less than threshold η, which is an experimental parameter, then the path with the least congestion is selected. Otherwise, it illustrates all one-hop paths are crowded. Searching for a light-loaded two-hop path begins by identifying an internal leaf switch as a transfer station. Furthermore, if the traffic is everywhere heavy, there is no need to adopt this alternative, because our method takes advantage of utilizing leisure resources though exacerbates traffic burden. When this happens, it will go back to search for the least crowded one-hop path.

D. Feedback Mechanism

Leaf switches possess the most functionality, while Spine ones just add the arrival Spine time to packets. Multi-hop CONGA passes the global congestion situation to each leaf switch through a feedback mechanism. Figure 3 shows an example of the entire procedure.

First, when receiving a new packet, the source switch tags the current time in the packet's Setting-out Time value. Then it checks the Flowlet Table to determine whether it belongs to a new flowlet. If it does then, a routing decision is made for the new flowlet. Otherwise, the packet is routed according to the Flowlet Table record.

The packet's header needs to refresh, including Uplink Port and the Flowlet Table's time column (if a two-hops path is adopted, Destination, Flag and Internal fields are also included). Moreover, if the switch's Feedback-Table has updated, it will be delivered by the packet.

Afterwards, the packet records its Spine Time value with the arrival time at the spine switch. Then it is sent based on its Destination information.

When receiving a new packet, the destination switch examines whether it is just a "transfer station" or not. If yes, the packet continues traversing towards its actual destination. Otherwise, the leaf switch updates the delay information into its Congestion-Table. If a Feedback-Table is carried by this packet, the corresponding areas also refresh in order to keep trace with latest congestion situation.

III. Theoretical Analysis

In this section, we give a distinct theoretical analysis of the necessity of Muti-hop CONGA. Suppose there is a topology with N leaf nodes and N spine nodes. We aim at comparing the non-obstruction possibility P of CONGA and P' of Multi-hop CONGA under the same situation. Consider one source switch s, who has a blocking links of all its N links, and one destination d, who has b blocking links. Assume that other leaf node's link has a probability μ to be blocked. When CONGA is P possibly congested at s, then we can specify the relationship among N, a, and b by the Equation 1. Because of Multi-hop CONGA's alternative two-hop path, it illustrates that its blocking possibility can be expressed as Equation 2. In order to make comparison explicitly, we might as well assume $a = b$. A table of P' with variables of P and μ, is introduced in Table I.

$$P = \frac{N-a}{N} * \frac{N-b}{N} = \frac{N^2 - N*(a+b) + ab}{N^2} \qquad (1)$$

TABLE I: Blocking Possibility Comparison

P’ μ / P	0	0.1	0.2	0.3	0.4	0.5
0.1	0.5325	0.4503	0.3768	0.3119	0.2557	0.2081
0.2	0.6944	0.6005	0.5164	0.4423	0.3780	0.3236
0.3	0.7954	0.7013	0.6171	0.5428	0.4784	0.4239

$$\begin{aligned} P' &= P + \frac{N-a}{N} * \frac{b}{N} * (1-\mu)^2 + \frac{a}{N} * \frac{N-b}{N} * (1-\mu)^2 \\ &= P + (\frac{N-a}{N} * \frac{b}{N} + \frac{a}{N} * \frac{N-b}{N}) * (1-\mu)^2 \\ &= P + \frac{N*(a+b)-2ab}{N^2} * (1-\mu)^2 \end{aligned} \tag{2}$$

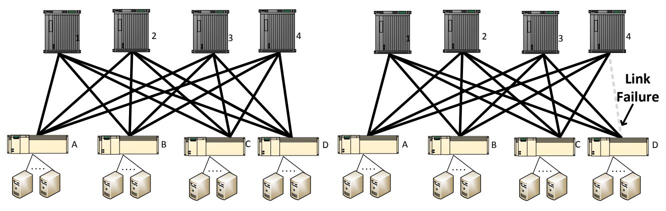
In the same column of the same μ, as P becomes larger, P' increases. The difference between P' and P is nearly the same, and it decreases as μ becomes bigger. This trend illustrates that the more the remaining leaf switches' links are light-loaded, the better our scheme defeats CONGA in avoiding congestion. In addition, μ's increment leads to a reduction in P'. Because if traffic is heavily-loaded everywhere, it is unlikely for Multi-hop CONGA to find a feasible two-hop path to detour. Furthermore, the delay may be worse than CONGA due to long transmission path. Multi-hop CONGA initially utilizes idle bandwidth to diffuse local congested traffic to global, so that it can avoid resource wasting and decrease transmission delay. In other words, when the data center's traffic distributes rather unevenly with bursty flows between two nodes and light-loaded traffic among other nodes, Multi-hop CONGA is more likely to exhibit its superiority over CONGA.

IV. Evaluation

In this section, we evaluate Multi-hop CONGA's performance in packet-level simulations and compare it with ECMP [1], Back Pressure [3] and CONGA [16] . Our simulations include three sights in data centers: regular traffic under intact topology; bursty flows; link failure. The intact topology is shown in Figure 4(a), while Figure 4(b) is the situation with link failure. As displayed in Figure 4, we test their performance with a network fabric consisting of 16 leaf switches and 16 spine switches, meanwhile each leaf switch is connected to 8 servers. Regular traffic pattern means a normal and virtually even flow distribution from each server, while a bursty pattern is emulated by creating one pair of source and destination traffic with three times volume than normal. Our experiments prove the efficiency and superiority of Multi-hop CONGA comprehensively in three aspects: the average packet transmission delay, the throughput and the average queue length. Our simulation is accomplished by using MATLAB.

A. Regular Traffic Pattern

Figure 5 shows the result for a regular workload with the baseline topology in Figure 4a. They illustrate the relationship of average packet transmission delay, throughput, queue length



(a) Baseline (Intact Topology) (b) Topology with link failure

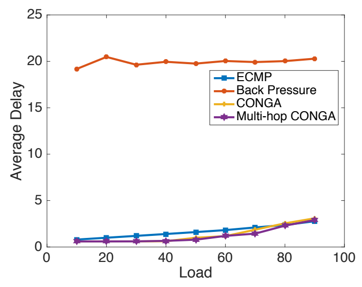
Fig. 4: Network Topology for simulations

with load percentage changes among $10 \sim 90\%$ (same measurements in Figure 6 and Figure 7). (Load percentage is the ratio between average traffic arrival rate and link bandwidth.) ECMP serves as the baseline in our experiments.

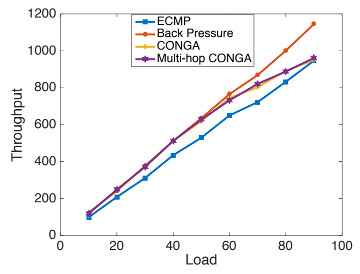
When the data center is working properly without any abnormality, our Multi-hop CONGA is well-matched with CONGA's performance. As traffic increases, Multi-hop CONGA and CONGA's advantages disappear due to the addition of too much load burden. When the traffic is essentially busy, packets of the same flow arrive consistently in little intervals. As a result, there is no difference in scheduling granularity of flow and flowlet. Multi-CONGA and CONGA can achieve ideal steady performances with relative small load percentages through scheduling based on global congestion situations. However, with a load of more than 50%, their queue length increases significantly and throughput's growing trend slows down. In particular, when traffic spreads evenly, or the load is heavy, two-hop paths are applied to the system sparingly to avoid additional transmission burden. While Back Pressure achieves the biggest throughput, it performs poor in transmission delay when traffic load increases. Generally, Back Pressure is able to transfer more packets under the same situation, as it directs packets to the least congested links to obtain more throughput. In Figure 7c, Back Pressure's queue length is the shortest because of its transfer-in-shortest-queue strategy, regardless of the packet's destination. In order to compare with our scheme of two-hop path, we limit Back Pressure 's path in at most two (spine switch) hops.

B. Bursty Traffic Pattern

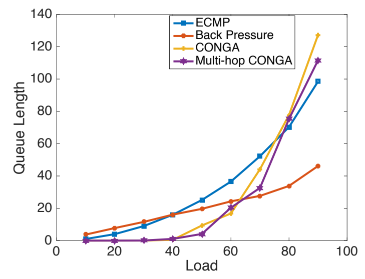
Regular traffic pattern is the most usual situation, but that does not mean the ability to handle abnormality is not as important. Today's data center calls for high robustness, scalability, and flexibility. This demands the scheduling mechanisms to perform smooth and steady. Figure 6 is obtained of bursty flows, whose volume are three times greater than usual. Naturally, traffic may locally stub in heavy obstruction, which possibly leads to serious loss. Multi-hop CONGA is superior than the other three under this traffic pattern. Its transmission time is the smallest, and the throughput as well as queue length is also satisfying. This excellent performance is obvious, when load is less than 50%. CONGA is a tight match, but slightly inferior, compared to our mechanism in delay and throughput. In regards to queue length, the distinction



(a) Average Transmission Delay

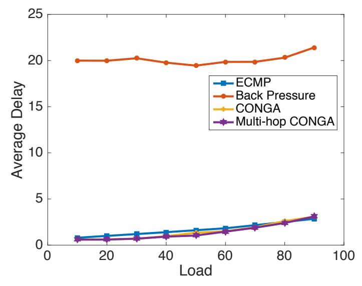


(b) Throughput

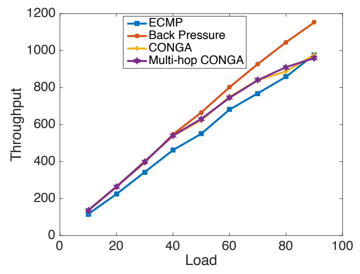


(c) Queue Length

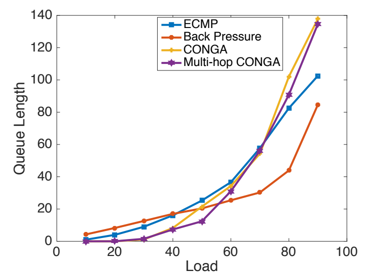
Fig. 5: Regular Traffic Pattern with Intact Topology



(a) Average Transmission Delay

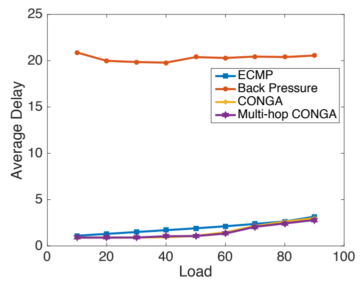


(b) Throughput

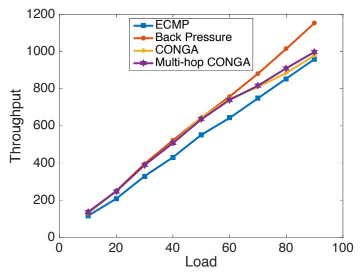


(c) Queue Length

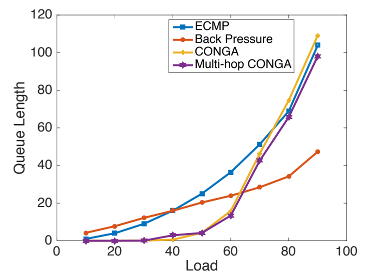
Fig. 6: Bursty Traffic Pattern with Intact Topology



(a) Average Transmission Delay



(b) Throughput



(c) Queue Length

Fig. 7: Regular Traffic Pattern with link failure topology

between these two methods is evident as a result of Multi-hop CONGA's higher possibility of scheduling a proper path. It avoids those congested links, finds idle bandwidth resource, and relives local traffic pressure.

C. Link Failure Pattern

Figure 7 is the result with one link failure, shown in Figure 5(b). Link failure occasionally happens in data centers due to their vast network scale. Link failure cuts off all traffic passing through it, which consequently stops flow from using it as part of its path. Local congestion-based schemes, like ECMP and Back Pressure, are unaware of links being broken when the schedule paths for packets. For that reason, the routing decision may trap flows in an awkward situation of no downlink, which may further cause loop or blackhole problems. Additionally, as an uplink, it has to pass its transmission burden to other ports of the same leaf switch. However, global congestion-aware methods will take consideration of overall traffic and topology to balance loads. CONGA and Multi-hop CONGA display their strengths of managing traffic with global congestion metrics. They direct flowlets to the least obstructed path according to its record obtained by piggyback. Additionally, Multi-CONGA can further distribute flowlets to all available paths (one-hop and two-hops) though potentially incurring more traffic loads. However, it is an approach to utilize relative leisure bandwidth to achieve both more throughputs and shortened queue length. We carefully control the scheduling granularity of flowlet which is between packets and flows. Flowlets can have greater flexibility than flows. However, the advantage disappears as traffic becomes more dense. This is because when the interval between two successive packets of the same flow is small, there is little difference between flows and flowlets.

TABLE II: Longest Packet Transmission Delay at $\lambda = 8$

	ECMP	Back Pressure	CONGA	Multi-hop CONGA
Regular Traffic Pattern	11	23	10	15
Bursty Taffic Pattern	11	25	14	20
Link Failure Pattern	14	27	13	18

D. Longest Transmission Delay

As demonstrated above, it is clear that no matter what situation the data center is experiencing, Multi-hop CONGA is overall the best and the most steady method. In order to understand their properties more comprehensively, we further compare their longest single packet travel times, shown in Table II. Multi-hop CONGA's result is not so good. Analytically, a two-hop path needs to travel additional two links, which generates extra transmission time. The worse circumstance happens to Back Pressure. This is because Back Pressure is locally congestion-aware and merely sends packets to the shortest queue, regardless of destination. So even with light traffic, its delay is also bad. However, Multi-hop CONGA ingeniously schedules a one-hop or two-hop path based on a global traffic knowledge vision. As a result, it is superior in load balancing in simulations. As for the other two methods, ECMP's longest transmission delay fluctuates little. CONGA has a satisfying performance, which shows its advantage of avoiding packet timeout. Additionally, we observe bursty flows and link failures have distinct influences. For example, our scheme's longest travel delay is worse with bursty flows than with link failure. Because it is harder to find a lighter-loaded path when bursty flows have spread their traffic across the whole topology.

V. Conclusion

Multi-hop CONGA is motivated to further improve network utilization and better balance loads in data centers. At its very core, Multi-hop CONGA leverages global traffic information to choose the least-congested path through one or two hops. We split flows into flowlets, schedule paths for the first packet of each new flowlet, and broadcast congestion metrics through piggyback. Multi-hop CONGA utilizes relative leisure links to distribute traffic in a more balanced manner, which in return obtains better throughputs and shorter transmission delay. We theoretically analyze and and test Multi-hop CONGA in simulations, which demonstrates its feasibility and necessity.

VI. Acknowledgment

This research was supported in part by NSF grants CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, and ECCS 1231461.

References

[1] C. Hopps, "Analysis of an equal-cost multi-path algorithm," 2000.

[2] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic load balancing without packet reordering," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 51–62.

[3] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[4] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'11, 2011, pp. 99–112.

[5] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath tcp," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11, 2011, pp. 266–277.

[6] S. Sen, D. Shue, S. Ihm, and M. J. Freedman, "Scalable, optimal flow routing in datacenters via local link balancing." in *CoNEXT*, 2013, pp. 151–162.

[7] M. Roughan, M. Thorup, and Y. Zhang, "Traffic engineering with estimated traffic matrices," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '03, 2003, pp. 248–258.

[8] Y. Chen, R. Griffith, J. Liu, R. H. Katz, and A. D. Joseph, "Understanding tcp incast throughput collapse in datacenter networks," in *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, ser. WREN '09, 2009, pp. 73–82.

[9] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'10, 2010, pp. 19–19.

[10] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM '13, 2013, pp. 3–14.

[11] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM '13, 2013, pp. 15–26.

[12] J. Cao, R. Xia, P. Yang, C. Guo, G. Lu, L. Yuan, Y. Zheng, H. Wu, Y. Xiong, and D. Maltz, "Per-packet load-balanced, low-latency routing for clos-based data center networks," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13, 2013, pp. 49–60.

[13] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: A scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, 2009, pp. 51–62.

[14] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, ser. SIGCOMM '08, 2008, pp. 63–74.

[15] S. L. A. Elwalid, C. Jin and I. Widjaja, "Mate: Mpls adaptive traffic engineering," *IEEE Transactions on Automatic Control*, vol. 3, no. 12, pp. 1300–1309, 2001.

[16] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, and G. Varghese, "Conga: Distributed congestion-aware load balancing for datacenters," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14, 2014, pp. 503–514.