Measuring Throughput of Data Center Network Topologies

Sangeetha Abdu Jyothi, Ankit Singla, P. Brighten Godfrey, Alexandra Kolla
University of Illinois at Urbana–Champaign

ABSTRACT

High throughput is a fundamental goal of network design. While myriad network topologies have been proposed to meet this goal, particularly in data center and HPC networking, a consistent and accurate method of evaluating a design's throughput performance and comparing it to past proposals is conspicuously absent. In this work, we develop a framework to benchmark the throughput of network topologies and apply this methodology to reveal insights about network structure. We show that despite being commonly used, cut-based metrics such as bisection bandwidth are the wrong metrics: they yield incorrect conclusions about the throughput performance of networks. We therefore measure flow-based throughput directly and show how to evaluate topologies with nearly-worst-case traffic matrices. We use the flow-based throughput metric to compare the throughput performance of a variety of computer networks. We have made our evaluation framework freely available to facilitate future work on design and evaluation of networks.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication networks**]: Network Architecture and Design—*network topology*

Keywords

throughput; topology comparison; network design

1. INTRODUCTION

A fundamental property of a network is its carrying capacity – how much data can be transported between the desired end-points per unit time? In settings varying from transport networks to communication networks, high capacity has been a core goal of network design. A large number of topologies have been proposed in the past few years in the communication networks domain, particularly in the areas of data centers [1, 2, 7] and high performance computing (HPC), to achieve high capacity at low cost.

However, there exists no framework that allows comparison of throughput performance of networks. Hence, network operators face an increasingly complex problem of choosing their network topology. The absence of a well specified benchmark also complicates research on network design, making it difficult to evaluate a new design against the numerous past proposals.

In fact, not only are we lacking throughput comparisons across a spectrum of topologies, we argue that the situation is worse: the community has been using the wrong metrics for measuring throughput. Cut-based metrics, in particular bisection bandwidth, are commonly used to estimate throughput, because minimum cuts are assumed to measure worst-case throughput [4]. However, while this is true for the case where the network carries only one flow, it does not hold for general traffic matrices. In the latter case, the cut is only an upper bound on throughput. The resulting gap between the cut-metric and the throughput leaves open the possibility that in a comparison of two topologies, the cut-metric could be larger in one topology, while the throughput could be greater in the other. An example scenario with precisely such an anomaly can be found in our technical report [5].

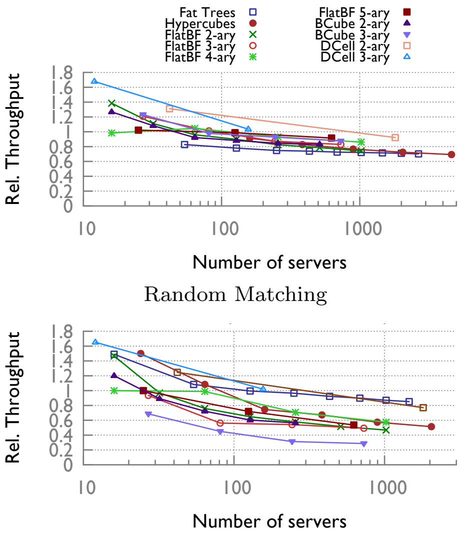
The goals of this work are precisely to overcome the above described problems: (a) allowing a fair, consistent, and accurate evaluation of network throughput to facilitate future network design efforts; (b) benchmarking existing network design proposals to inform the choice of topologies for applications; (c) examining the relationship of throughput with other graph metrics across several natural topologies such as biological networks and social networks to understand what gives a topology high throughput.

2. EVALUATION OF TOPOLOGIES

A **network topology** is a graph $G = (V, E)$ with capacities $c(u, v)$ for every edge $(u, v) \in E_G$. Among the nodes V are **servers**, which send and receive traffic flows, connected through non-terminal nodes called **switches**. Each server is connected to one switch, and each switch is connected to zero or more servers, and other switches. For switch-to-switch edges (u, v), we set $c(u, v) = 1$, while server-to-switch links have infinite capacity. This allows us to stress-test the topology. A **traffic matrix (TM)** T defines the traffic demand: for any two servers v and w, $T(v, w)$ is an amount of requested flow from v to w.

The **throughput** of a network G with TM T is defined as the maximum value t for which $T \cdot t$ is feasible in G. That is, we seek the maximum t for which there exists a feasible multicommodity flow that routes flow $T(v, w) \cdot t$ through the

SIGMETRICS'14, June 16–20, 2014, Austin, Texas, USA.
ACM 978-1-4503-2789-3/14/06.
http://dx.doi.org/10.1145/2591971.2592040.



Random Matching

Longest Matching

Figure 1: Relative throughput of computer networks under different traffic matrices

network from each v to each w, subject to the link capacity and the usual flow conservation constraints. This can be formulated in a standard way as a linear program (which we omit for brevity) and is thus computable in polynomial time. If the nonzero traffic demands $T(v, w)$ have equal weight, as they will throughout this paper, this is equivalent to maximizing the minimum throughput of any of the requested end-to-end flows. This maximization is a standard problem called *maximum concurrent flow* [6]. Furthermore, this objective function captures the notion of fairness among flows. We use the Gurobi [3] solver to compute throughput by solving this linear program.

We use this framework to evaluate the performance of different classes of computer networks. Our evaluation addresses the following questions: (a) How do we fairly compare networks with different sizes and degree distributions? (b) Which networks achieve highest throughput?

Our high-level approach to evaluating a network is to: (i) compute the network's throughput for a certain TM; (ii) build a random graph with precisely the same degree distribution and compute its throughput under the same traffic model; (iii) normalize the network's throughput with the random graph's throughput to obtain the relative throughput for comparison against other networks. For example, a fat-tree network may have a relative throughput of around 0.75, indicating that it has 25% lower throughput than a random network built with the same equipment.

For ensuring high network performance across different applications (TMs), it is necessary to benchmark networks using worst-case throughput, *i.e.,* to evaluate networks with TMs for which it is hard to achieve high throughput. While finding the worst-case TM is NP-Hard, in our technical report [5], we prove that an all-to-all TM is within a factor of two of the worst-case TM. Further, we experiment with other traffic patterns that are (empirically) even harder than an all-to-all TM. We include here results for two such nearly worst-case traffic matrices – (a) random permutation (each server sends traffic to and receives traffic from exactly one other randomly chosen server in the network) and (b) longest matching (each server sends traffic to and receives traffic from a server located farthest from it). Figure 1 shows that as size increases, relative throughput drops below 1 for all networks we tested, indicating that random graphs achieve higher throughput than the others. Further, the longest matching TM widens the throughput gap between different networks, with some (such as BCube) taking a large performance hit. Among structured networks, DCell and flattened butterflies achieve highest throughput for random matching, but for longest matching, the fat-tree achieves highest throughput at the largest scale. The worst degradation was observed in BCube under longest matching TM.

Our technical report [5] includes additional results addressing the following questions:

- What are the shortcomings of existing cut-based throughput metrics such as bisection bandwidth and sparsest-cut? Why is a flow-based metric more accurate than a cut-based metric?
- What traffic matrices should we use to test the performance of a network?
- What throughput characteristics do naturally occurring networks exhibit?
- How well do other graph metrics such as clustering or path length correlate with throughput? Can we use such correlations (if any) to improve network design?
- Do models of natural networks faithfully reproduce their throughput?

3. CONCLUSION

In this work, we present a framework that allows fair comparison of network topologies by direct, flow-based measurement of throughput. Using this benchmark, we analyzed the throughput performance of a variety of networks and demonstrated that carefully structured computer networks perform poorly compared to random graphs. Our evaluation should be useful to network operators and researchers alike. Our tool is freely available [8]. More comparisons can be found in the technical report [5].

4. REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *SIGCOMM*, 2008.

[2] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: A scalable and fault-tolerant network structure for data centers. In *SIGCOMM*, 2008.

[3] Gurobi Optimization Inc. Gurobi optimizer reference manual. `http://www.gurobi.com`, 2013.

[4] D. Padua. *Encyclopedia of Parallel Computing*. Number v. 4 in Springer reference. Springer, 2011.

[5] A. J. Sangeetha, A. Singla, B. Godfrey, and A. Kolla. Measuring and understanding throughput of network topologies. *CoRR*, abs/1402.2531, 2014.

[6] F. Shahrokhi and D. Matula. The maximum concurrent flow problem. *Journal of the ACM*, 37(2):318–334, 1990.

[7] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Network data centers randomly. In *NSDI*, 2012.

[8] A. Singla, C.-Y. Hong, A. J. Sangeetha, B. Godfrey, and L. Popa. Our topology evaluation framework. `https://github.com/ankitsingla/topobench`.