Brief Announcement: Bridging the Theory-Practice Gap in Multi-Commodity Flow Routing

Siddhartha Sen, Sunghwan Ihm, Kay Ousterhout, and Michael J. Freedman
Princeton University

1 Introduction

In the *concurrent multi-commodity flow problem*, we are given a capacitated network $G = (V, E)$ of switches V connected by links E, and a set of commodities $\mathcal{K} = \{(s_i, t_i, d_i)\}$. The objective is to maximize the minimum fraction λ of any demand d_i that is routed from source s_i to target t_i. This problem has been studied extensively by the theory community in the sequential model (e.g., [4]) and in distributed models (e.g., [2,3]). Solutions in the networking systems community also fall into these models (e.g., [1,6,5]), yet none of them use algorithms from the theory community. Why the gap between theory and practice? This work seeks to answer and resolve this question. We argue that existing theoretical models are ill-suited for real networks (§2) and propose a new distributed model that better captures their requirements (§3). We have developed optimal algorithms in this model for data center networks (§4); making these algorithms practical requires a novel use of programmable hardware switches. A solution for general networks poses an intriguing open problem.

2 Existing Models: Theory vs. Practice

Prior solutions to the multi-commodity flow problem fall in one of three models. In the *sequential model* [4], the entire problem input (solution) is known (computed) by a single entity. In the *Billboard model* [3], routing decisions are made by agents at the sources, one per commodity, that can read and write to a global "billboard" of link utilizations. In the *Routers model* [2], routing decisions are made locally by all switches by communicating only with their neighbors.

The fundamental problem with these models is that they are designed for a static demand matrix (i.e., a single set of commodities), whereas real systems must respond to rapidly changing demand matrices. The cost of collecting demands and communicating the flows in the sequential model makes it impractical to respond to changing demands at small timescales. Thus, systems like Hedera [1] recompute the flows from scratch at large scheduling intervals (seconds). Similarly, the cost of flooding link utilizations to the sources in the Billboard model causes systems like MPTCP [6] to apply only coarse congestion control at sources based on indirect information. The Routers model evades these problems, but because switches have no *a priori* knowledge of the network topology, flows may change direction or circulate repeatedly in the network. Thus, systems like FLARE [5] use pre-established routes and avoid rerouting altogether.

The second problem is that all known polynomial-time solutions, in all models, require fractionally splitting flows. Splitting flows causes packets to get reordered, which causes throughput to collapse in the TCP protocol. If a flow's paths have inconsistent latencies, queuing occurs at the target; such uncertain packet delivery times make it

difficult to time retransmissions without exacerbating congestion. Thus, systems use either heuristics to solve the integer (unsplittable) multi-commodity flow problem [1], or complicated splitting heuristics that still cause reordering across subflows [6].

The third problem is that all models incorrectly assume that hardware switches are identical to end hosts. To forward traffic at line rate—1 or 10 Gbps for today's commodity switches—switches require high-speed matching on packet headers and offer limited general-purpose processing. Practical solutions must operate within these limits.

3 Routers Plus Pre-processing (RPP) Model

In almost any wired network, the demand matrix changes far more frequently than the network topology. Thus, we propose the following extension to the Routers model: *We allow arbitrary (polynomial-time) pre-processing of the network G at zero cost in time* (but charge for any space required to store the results). This decouples the problem of topology discovery from routing, turning the former into a *dynamic graph problem*.

We also introduce two novel issues of practicality. First, we allow $O(1)$-sized messages that are injectively mapped to flow packets, or *in-band messages*, to be sent for free, and charge only for *out-of-band messages*. Second, when possible, we ensure paths of the same commodity are roughly equal in length, to minimize queuing and reordering at the target. We are interested in algorithms that are *partially asynchronous*, since otherwise we would need expensive synchronizers to simulate rounds.

4 Algorithms

We have devised a simple algorithm in the RPP model for data center fat-tree networks [1]. Our algorithm locally splits and rate-limits the aggregate demand to each target with the help of in-band messages, routing the maximum concurrent flow in an optimal $O(H)$ parallel rounds, where H is the length of the longest flow path. By allowing approximate splitting, we can drastically reduce the amount of splitting in practice. Our solution uses carefully crafted rules in switches' forwarding tables that allow line-rate processing while minimizing packet reordering at the targets.

A solution for general networks in the RPP model poses an intriguing open problem. One approach may be to use the free pre-processing to initialize connectivity oracles that can route around "failed" (congested) links.

References

1. M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. Hedera: dynamic flow scheduling for data center networks. In *Proc. NSDI*, pages 281–296, 2010.
2. B. Awerbuch and R. Khandekar. Distributed network monitoring and multicommodity flows: a primal-dual approach. In *Proc. PODC*, pages 284–291, 2007.
3. B. Awerbuch and R. Khandekar. Greedy distributed optimization of multi-commodity flows. *Distrib. Comput.*, 21(5):317–329, 2009.
4. N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM J. Comput.*, 37(2):630–652, 2007.
5. S. Kandula, D. Katabi, S. Sinha, and A. Berger. Dynamic load balancing without packet reordering. *SIGCOMM Comput. Commun. Rev.*, 37, 2007.
6. D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. In *Proc. NSDI*, pages 99–112, 2011.