# A Constant Approximation for Maximum Throughput Multicommodity Routing

And Its Application to Delay-Tolerant Network Scheduling

Mengxue Liu*, Andréa W. Richa*, Matthias Rost‡ and Stefan Schmid§
*Computer Science and Engineering, CIDSE, Arizona State University, USA
Email: Mengxue.Liu@asu.edu, Andrea.Richa@asu.edu
‡Department of Telecommunication Systems, TU Berlin, Germany
Email: mrost@inet.tu-berlin.de
§Faculty of Computer Science, University of Vienna, Austria
Email: stefan_schmid@univie.ac.at

***Abstract*—This paper considers the following fundamental maximum throughput routing problem: given a set of $k$ (splittable) multicommodity flows with equal demands in an $n$-node network, select and route a subset of flows such that the total number of commodities routed that satisfy their demands (i.e., the all-or-nothing throughput) is maximized. Our main contribution is the first constant (i.e., independent of $k$ and $n$) throughput-approximation algorithm for this NP-hard problem, with sublinear, namely $\tilde{O}(\sqrt{k})$, edge capacity violation ratio. Our algorithm is based on a clever application of randomized rounding. We also present an interesting application of our result in the context of delay-tolerant network scheduling. We complement our theoretical contribution with extensive simulation in two different scenarios, and find that our algorithm performs significantly better than predicted in theory, achieving an edge capacity violation ratio of at most 3.**

## I. Introduction

The admission and allocation of multiple commodities in a graph is among the most fundamental and intensively studied algorithmic problems in networking. It is well-known that while for a single source-destination pair $(s, t)$, the maximum possible flow from $s$ to $t$ is equal to the value of a minimum $s{-}t$ cut, this no longer holds for three or more commodities. While over the last decades, much progress has been made, some basic questions related to the maximization of the throughput achieved by multiple commodities in a network remain unresolved.

### A. ANF: A Fundamental Network Optimization Problem

This paper attends to the following maximum throughput routing problem, henceforth referred to as the *All-or-Nothing (Splittable) Multicommodity Flow (ANF)* problem: given a set of (splittable) multicommodity flows with equal demands, select and route a subset of flows such that the total number of commodities routed that satisfy their demands (i.e., the all-or-nothing throughput) is maximized.

We model a flow network as a capacitated directed graph $G(V, E)$, where $V$ is the set of nodes and $E$ is the set of edges, and where each edge $e$ has a given capacity $c(e) > 0$. We are also given a set of $k$ commodities $\mathcal{F} = \{F_1, ..., F_k\}$, each with equal demand $d$. Each commodity $F_i \in \mathcal{F}$ is denoted by a pair $(s_i, t_i)$ where $s_i, t_i$ denote the source and destination for that commodity. Commodity $F_i$ is *satisfied* if $d$ units of flow for this commodity can be successfully routed in the network.

In the ANF problem, one would like to maximize the total throughput in the network: the throughput is measured in terms of the total number of commodities that are concurrently satisfied in a valid multicommodity flow. Note that we do not insist that the flow for a commodity $F_i$ be non-splittable or even that the flow on each edge be integral. The only assumption we make is that, for any $i$, the network has enough capacity to route $d$ units of flow from $s_i$ to $t_i$, if we were to consider commodity $F_i$ alone in the network. The load of an edge $e$ is equal to $\sum_i f_{i,e}$, where $f_{i,e}$ is the flow for commodity $i$ on edge $e$.

In this paper, we seek to find a polynomial time $(\alpha, \beta) - \mathit{approximation}$ algorithm for the ANF problem, for constants $\alpha < 1$ and $\beta \geq 1$, for suitable values of $\alpha$ and $\beta$: Namely, we seek to find a solution for the ANF problem such that the throughput is at least an $\alpha$ fraction of the maximum throughput, and the load on any edge is at most $\beta$ times the edge capacity, with high probability (i.e. with probability at least $1 - 1/n^c$, where $c > 0$ is a constant). The parameter $\beta$ hence provides an upper bound on the *edge capacity violation ratio* incurred by the algorithm.

### B. Applications to Time-Scheduled Networks

Besides maximizing the throughput in a network, a fundamental problem in itself, we show another interesting application of our result in this paper: scheduling in Delay-Tolerant Networks (DTNs), and in particular, in *time-scheduled networks*. Time-scheduled networks are networks in which mobile nodes move between stationary nodes according to a fixed plan and route.

For example, it is expected that the number of "smart things" equipped with wireless communication capabilities will grow

Supported in part by an ASU-BGU Collaboration Seed Grant, and NSF grants CCF-1422603, CCF-1637393, and CCF-1733680.

quickly over the next years, with these mobile nodes (i.e., smart things) moving between stationary nodes (the hotspots) at regular, pre-determined time schedules as "*data mules*", where data can be uploaded (resp. downloaded) to (resp. from) the Internet. Besides the Internet-of-Things (IoT) [1, 2], time-scheduled networks can come in the form of (mobile) social networks: here, the moving nodes may be, e.g., students or tourists, commuting back and forth at pre-determined times, and the stationary nodes classrooms, restaurants, shopping malls, etc. [11, 16, 18]. Another example are cost-efficient transmissions in Internet Service Provider (ISP) networks: ISPs can reduce transmission costs by an intelligent scheduling of communications, taking advantage of already-paid-for, off-peak bandwidth resulting from diurnal traffic patterns and percentile pricing [14].

### C. Our Contributions

This paper makes three main contributions. Our main theoretical contribution is the first algorithm for the ANF problem that achieves a constant approximation on the max throughput with a sublinear (in $k$) edge capacity violation ratio; our algorithm is based on a non-trivial application of relaxation and randomized rounding techniques, which may be of independent interest. Our second contribution is a reduction of our result to scheduling transmissions in delay-tolerant networks. Third, we validate our proposed algorithm through extensive simulations in two case studies: throughput maximization in a synthetic benchmark network (German50) and scheduling in a DTN. The simulation results show that, in practice our algorithms perform significantly better than our theoretical worst-case bounds suggest, and violate the considered edge capacities by less than a factor of 2 on average and less than a factor of 3 in the worst-case.

### D. Technical Novelty

Let us briefly elaborate on the technical novelty of our result; a more detailed and comprehensive discussion of prior work will follow later in this paper. The polynomial-time algorithm presented in this paper provides an $(\alpha, \beta)$-approximation of the ANF problem, where the approximation ratio $\alpha$ on the network throughput is *constant*, and the edge capacity violation ratio $\beta$ is bounded by $O(\sqrt{(\log(|V|) \cdot k)})$. Unlike the results by Chekuri et al. [7] (the closest paper to ours), who present an approximation algorithm for the ANF problem with constant $\beta$ and $\alpha$ in $\Omega(1/(\log^3 |V| \log\log |V|))$, our approach keeps $\alpha$ constant and allows $\beta$ not to be constant. This is a fundamental and non-trivial difference, as the approach in [7] does not imply the $(\alpha, \beta)$-approximation tradeoff that we obtain in this paper: It does not seem possible to modify the approach in [7] to ensure a constant approximation on the throughput (without incurring a linear violation of the edge capacities).

### E. Organization

The remainder of the paper is organized as follows. Section II presents our randomized rounding algorithm together with an analysis of its achieved approximation ratio. In Section III we descibe an application to time-scheduled networks. We evaluate our algorithm in different case studies, using simulations, in Section IV and V. After reviewing related work in Section VI, we conclude our contribution in Section VII.

**IP 1** All-or-Nothing (Splittable) Flow Formulation (ANF)

$$\text{Maximize } \sum_{i=1}^{k} f_i \text{ subject to}$$

$$\begin{aligned}
&\sum_{(s_i,v)\in E} f_{i,(s_i,v)} = f_i && \forall F_i \in \mathcal{F} && (1)\\
&\sum_{(u,v)\in E} f_{i,(u,v)} = \sum_{(v,u)\in E} f_{i,(v,u)} && \forall F_i \in \mathcal{F}, v \in V \setminus \{s_i, t_i\} && (2)\\
&\sum_{i=1}^{k} f_{i,(u,v)} \le c_{(u,v)} && \forall (u,v) \in E && (3)\\
&f_{i,(u,v)} \le f_i \cdot c_{(u,v)} && \forall F_i \in \mathcal{F}, (u,v) \in E && (4)\\
&f_{i,(u,v)} \ge 0 && \forall F_i \in \mathcal{F}, (u,v) \in E && (5)\\
&f_i \in \{0,1\} && \forall F_i \in \mathcal{F} && (6)
\end{aligned}$$

## II. Algorithm and Analysis

This section presents and analyzes our approximation algorithm as described in Algorithm 2 for maximizing the total number of commodities that can be successfully delivered. The algorithm is based on randomized rounding, and uses a relaxation of the integer (linear) program, IP, that solves the ANF problem on a given network $G(V, E)$. We use a variable $f_i$ to represent whether or not commodity $F_i$ is successfully delivered; the variables $f_{i,e}$ denote the flow for commodity $F_i$ on edge $e$. For the sake of simplicity, and without loss of generality, we rescale each commodity demand and edge capacity in the network by $d$, so that this problem becomes a unit-flow, multicommodity problem.

IP 1 describes our (mixed) integer programming formulation of the ANF problem. To be more specific, we aim to maximize the sum of integral flows $f_i$, while subject to six constraints: Constraints (2) and (4) are the standard flow conservation and edge capacity constraints. Constraints (4) are redundant in this integer programming formulation since they are implied by Constraints (3): Constraints (4) will be needed and relevant for strengthening the quality of the linear relaxation of IP 1 as we will see later. Constraints (6) indicate whether or not commodity $F_i$ is successfully delivered ("all or nothing"). The ANF problem is NP-hard and hence one should seek approximate solutions for IP 1. Therefore, we propose to first relax IP 1 and then apply a randomized rounding approach. We depict the details of the algorithm in Algorithm 2.

Note that after we select a commodity $i$ for routing using randomized rounding, namely by setting the respective $f_i$ value of commodity $i$ to 1 in Line 3 of Algorithm 2, we need to rescale the flow on all the edges for commodity $i$ by a factor of $1/\tilde{f}_i$ so that the total flow for that commodity in the final solution output by Algorithm 2 is indeed equal to 1 (and hence the commodity is satisfied). Showing that this rescaling only violates the edge capacities by a sublinear, namely an $\tilde{O}(\sqrt{k})$

factor, with high probability is one of the main challenges in the splittable multicommodity flow setting. The strengthening of the relaxed solution of IP 1 given by Constraints (4) will be crucial in order to get the sublinear violation ratio of the edge capacities, as we show when proving Theorem II.5.

**Algorithm 2** Randomized Rounding Algorithm

```
Input:
    Directed Graph $G(V, E)$
    Commodities $\mathcal{F} = \{F_1, ..., F_k\}$
    Source-Sink Pair $(s_i, t_i)$ for each $F_i \in \mathcal{F}$
    Capacity $c(u, v)\ \forall (u, v) \in E$
Output: The final values of $f_i$ and $f_{i,e}$ and $OPT_{ALG} = \sum f_i$
1: Change the last constraint to be $0 \le f_i \le 1$
2: Relax and solve LP, obtain optimal solution $\tilde{f_i}$
3: With probability $\tilde{f_i}$, set $f_i = 1$, otherwise set it to 0
4: Scale up the fractional flow $\tilde{f}_{i,e}$ from the LP solution on edge $e$
   for commodity $i$ by $\frac{1}{\tilde{f_i}}$, i.e., $f_{i,e} = \tilde{f}_{i,e} \times \frac{1}{\tilde{f_i}}$, for $i$ s.t. $f_i = 1$
5: If the solution is greater than an $\alpha$ fraction of the optimal solution,
   return this solution; otherwise, repeat step 3 and 4, at most
   $\theta(\log |V|)$ times
```

Let $OPT_{IP}$ be the value (i.e., throughput) of the optimal solution and let $OPT_{LP}$ be the total amount of (fractional) commodities delivered by solving the linear relaxation LP of IP 1, where the 0-1 variables $f_i$ are relaxed to assume any value in $[0, 1]$ — i.e., $OPT_{LP} = \sum \tilde{f_i}$, where $\tilde{f_i}$ is the optimal solution of LP. Let $ALG$ be the solution obtained by Algorithm 2, whose throughput we denote by $OPT_{ALG}$.

We use Chernoff's Theorem over continuous random variables to bound the probability of achieving a fraction of the optimal solution, which we state bellow:

**Fact II.1** (Chernoff Bound). *[17] Let $X = \sum_{i=1}^{n} X_i$ be a sum of $n$ independent random variables $X_i \in [0, 1], 1 \le i \le n$. Then $Pr[X < (1-\epsilon) \cdot E[X]] \le exp(-\epsilon^2 \cdot E[X]/2)$ holds for $0 < \epsilon < 1$.*

Since we have already scaled down the flow by $d$, variables $\tilde{f_i}$ are all between 0 and 1, and we can apply the above Chernoff bound to prove the claim below.

**Claim II.2.** $Pr[OPT_{ALG} < (1-\epsilon) \cdot OPT_{LP}] \le e^{-\epsilon^2 \cdot OPT_{LP}/2}$

*Proof.* For each commodity $i$, the expectation of $f_i$ according to Algorithm 2 is $E[f_i] = 1 \cdot \tilde{f_i} + 0 \cdot (1 - \tilde{f_i}) = \tilde{f_i}$. Recall that $OPT_{LP} = \sum \tilde{f_i}$ and $OPT_{ALG} = \sum f_i$, where the $f_i$'s are the solution output by Algorithm 2, and let $OPT_{ALG} = \sum f_i$. From Fact II.1, we have

$$\begin{aligned} & Pr[OPT_{ALG} < (1-\epsilon) \cdot OPT_{LP}] \\ & = Pr[\sum f_i < (1-\epsilon) \cdot OPT_{LP}] \\ & \le e^{-\epsilon^2 \cdot OPT_{LP}/2} \end{aligned} \tag{7}$$

We also assume that all commodities can be fully delivered alone, and so we have:

$$OPT_{LP} \ge 1 \tag{8}$$

□

Since the optimal fractional solution is an upper bound on the optimal solution for the integer program, by using $\epsilon = 2/3$ in the Chernoff bound, we get:

**Theorem II.3.** *The probability of achieving less than 1/3 of the profit of an optimal solution is upper bounded by $e^{-2/9} \approx 0.8007$.*

*Proof.* By taking $\epsilon = 2/3$ in Equation 7, we get:

$$Pr[OPT_{ALG} < \frac{1}{3} \cdot OPT_{LP}] \le e^{-(\frac{2}{3})^2 \cdot OPT_{LP}/2} \tag{9}$$

$$= e^{-2 \cdot OPT_{LP}/9} \tag{10}$$

By Equation 8, we know that the minimum value of $OPT_{LP}$ is 1. Therefore

$$Pr[OPT_{ALG} < \frac{1}{3} \cdot OPT_{LP}] \le e^{-2/9} \tag{11}$$

We have that $OPT_{IP} \le OPT_{LP}$ and thus we get,

$$Pr[OPT_{ALG} < \frac{1}{3} \cdot OPT_{IP}] \le e^{-2/9}$$

□

We will use the following result by Hoeffding, in order to bound the edge capacity violation:

**Fact II.4** (Hoeffding's Inequality). *[9] Let $\{X_i\}$ be independent random variables, s.t. $X_i \in [a_i, b_i]$, then $Pr[\sum_i X_i - E[\sum_i X_i] \ge t] \le exp(-2t^2/\sum_i (b_i - a_i)^2)$ holds.*

We now analyze the probability that our algorithm violates capacity constraints by a certain factor, with the aid of Hoeffding's Inequality.

**Theorem II.5.** *Given a single edge $e$ with capacity $c_e$. Let $\Delta_{F,e}$ be the commodities going through edge $e$, thus $|\Delta_{F,e}| \le k$. For all commodities $i \in \Delta_{F,e}$, choose constant $\epsilon' \le 1$ such that $\max \frac{\tilde{f}_{i,e}}{\tilde{f_i}} \le \epsilon' \cdot c_e$. The probability that $ALG$ exceeds the edge capacity constraints by a factor of $\gamma = (1 + \epsilon' \cdot \sqrt{2 \log |V| |\Delta_{F,e}|})$ is bounded by $|V|^{-4}$.*

*Proof.* Fix an edge $e \in E$ and a commodity $i$. With probability $1 - \tilde{f_i}$, the flow on edge $e$ for commodity $i$ is set to 0, i.e., $f_{i,e} = 0$. With probability $\tilde{f_i}$, the flow on edge $e$ for commodity $i$ is set to $\tilde{f}_{i,e} \cdot \frac{1}{\tilde{f_i}}$.

Then the expectation of $f_{i,e}$ is

$$E[f_{i,e}] = \tilde{f}_{i,e} \cdot \frac{1}{\tilde{f_i}} \cdot \tilde{f_i} + 0 \cdot (1 - \tilde{f_i}) = \tilde{f}_{i,e} \tag{12}$$

Let $f^e$ denote the total flow on edge $e$ induced by $ALG$. Then $f^e = \sum_{i, f_{i,e} \neq 0} f_{i,e}$ and the expectation of $f^e$ is

$$E[f^e] = \sum_{i, f_{i,e} \neq 0} \tilde{f}_{i,e} \cdot \frac{1}{\tilde{f_i}} \cdot \tilde{f_i} = \sum_{i, f_{i,e} \neq 0} \tilde{f}_{i,e} \tag{13}$$

Since in the relaxed version of IP 1, a feasible solution must obey edge capacity constraints, the cumulative load on edge $e$ is equal to or less than the capacity of $e$, i.e., $\sum_{i,f_{i,e}\neq 0} \tilde{f}_{i,e} \leq c_e$, and therefore we get

$$E[f^e] \leq c_e \tag{14}$$

Let $t = \epsilon' \cdot \sqrt{2\log|V||\Delta_{F,e}|} \cdot c_e$. Note that $\frac{\tilde{f}_{i,e}}{\tilde{f}_i} \leq c_e, \forall i, e$, (since Constraints (4) of LP guarantee that $\tilde{f}_{i,e} \leq \tilde{f}_i \cdot c_e$) and hence we can always find $\epsilon' \leq 1$ such that $\max \frac{\tilde{f}_{i,e}}{\tilde{f}_i} \leq \epsilon' \cdot c_e$. Applying Hoeffding's Inequality, we get

$$\begin{aligned} Pr[f^e - E(f^e) \geq t] &\leq exp(\frac{-2t^2}{\sum_i (\frac{\tilde{f}_{i,e}}{\tilde{f}_i})^2}) \\ &\leq exp(\frac{-2 \cdot \epsilon'^2 \cdot 2\log|V| \cdot |\Delta_{F,e}| \cdot c_e^2}{\epsilon'^2 \cdot c_e^2 \cdot |\Delta_{F,e}|}) \\ &= |V|^{-4} \end{aligned} \tag{15}$$

Given Equation 14, let $\gamma = (1 + \epsilon' \cdot \sqrt{2\log|V||\Delta_{F,e}|})$.

$$\begin{aligned} & Pr[f^e - c_e \geq \epsilon' \cdot \sqrt{2\log|V||\Delta_{F,e}|}] \\ = \ & Pr[f^e \geq (1 + \epsilon' \cdot \sqrt{2\log|V||\Delta_{F,e}|}) \cdot c_e] \\ = \ & Pr[f^e \geq \gamma \cdot c_e] \\ \leq \ & |V|^{-4} \end{aligned}$$

□

Since there are at most $|V|^2$ edges, and we know that $|\Delta_{F,e}| \leq k$, applying the union bound over all edges using Theorem II.5, we obtain the following corollary:

**Corollary II.6.** *The probability that ALG exceeds any of the edge capacity constraints by a factor of* $1 + \epsilon' \cdot \sqrt{2\log|V| \cdot k}$ *is upper bounded by* $|V|^{-2}$.

Based on Theorem II.3 and Corollary II.6, if $|V| \geq 3$ holds, the probability of not finding a suitable solution with a 1/3-approximation on the throughput and an edge capacity violation ratio of $(1 + \epsilon' \cdot \sqrt{2\log|V| \cdot k})$ within a single round of Algorithm 1 is therefore upper bounded by $e^{-2/9} + 1/9 \leq 11/12$. The probability of finding a suitable solution within $c\log|V|$ many rounds (see line 10 of Algorithm 2), where $c$ is a constant, is then lower bounded by $1 - (11/12)^{c\log|V|} = 1 - \frac{1}{|V|^b}$, where $b$ is a constant, for $|V| \geq 3$. Hence the randomized rounding scheme yields a solution with a 1/3-approximation on the throughput and an edge capacity violation ratio of $(1 + \epsilon' \cdot \sqrt{2\log|V| \cdot k})$ with high probability, proving our main theorem below:

**Theorem II.7.** *The randomized rounding algorithm finds an* $(\alpha, \beta)$ *– approximation solution within* $c\log|V|$ *many runs with high probability, where c is a constant,* $\alpha = 1/3$, $\beta = 1 + \epsilon' \cdot \sqrt{2\log|V| \cdot k}$, $\epsilon' \leq 1$, $|V| \geq 3$.

## III. Application in Delay Tolerant Networks

Our algorithm has interesting applications in Delay Tolerant Networks (DTNs) and in particular, time-scheduled networks: wireless communication networks in which mobile nodes (e.g., IoT devices storing data) move between stationary nodes (e.g., access points or hotspots), according to pre-determined routes and schedule. The goal is to deliver as much information as possible across the DTN. In the following, we first introduce the time-scheduled network model and then show how our algorithm can be applied.

Let $\mathcal{M} = \{M_1, M_2, M_3, ..., M_m\}$ denote the set of $m$ mobile nodes (MNs), and let $\mathcal{S} = \{S_1, S_2, S_3, ..., S_n\}$ denote the set of $n$ stationary nodes (SNs) in the original network. In addition, we assign buffers of infinite size to both MNs and SNs.[1] Let $P_i$ be the time schedule associated with each MN $M_i$, which describes the paths and times to arrive at and depart from the SNs (we assume that each MN has full knowledge of its own time schedule). A connection is established between two MNs $M_i$ and $M_j$ at time $t$ if and only if their geographical distance is within a certain constant range $r_{i,j,t}$ at time $t$. Note that the ranges $r_{i,j,t}$ may be all different depending on the particular MN and the time (which also determine location) of the connection. However, without loss of generality and for ease of explanation, we will assume that $r_{i,j,t} = r$, for all $i, j, t$. Given also the sparsity of our network scenario, we will ignore considerations of interference of the wireless signal in this work, but we do assume that all communications are half-duplex (i.e., a node can either transmit or receive at a time). Hence a connection exists between $M_i$ and $M_j$ at time $t$ if and only if: $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r$, where $(x_z, y_z)$ are the coordinates of the respective $M_z$ and $(x_j, y_j)$ at time $t$. We also assume that we have a set of $k$ commodities $\mathcal{F}$=$\{F_1, F_2, ..., F_k\}$, each with source and destination $s_i, t_i$ respectively and demand $d$, which can be transferred through wireless senders and receivers equipped on the NMs and SNs when a connection is established. Based on this observation, we transform our original network into a *connection graph*, which we will describe next.

We present our *connection graph model* whose nodes correspond to the moving and stationary nodes' connections in the original dynamic network. Based on the original MN time schedules, we can determine all MN and SN connections $\mathcal{C}$=$\{C_1, C_2, C_3, ..., C_q\}$ where $q \leq cm(m+n)$ and where $c$ is a constant. We can represent each connection $C_i$ as a 4-tuple $< A_i, B_i, Up_i, Down_i >$, where $A_i$ and $M_i$ are the two objects (either a MN or SN) that establish this connection, $Up_i$ is the starting time of the connection, and $Down_i$ is the ending time of the connection.

We construct a directed connection graph using the above connections and commodities as nodes plus a set of virtual sink nodes $sink_i$, one for each commodity $F_i$. Therefore the node set is $V = \mathcal{C} \cup \mathcal{F} \cup \{sink_i : F_i \in \mathcal{F}\}$ and $|V| = q + 2k$. A directed edge exists from connection node $C_x$ to connection node $C_y$ if and only if the two connections share a common object and $Up_x \leq Down_y$. For example, there is an edge from

[1]While an infinite buffer size may seem unrealistic, in practice, this would not be a concern, since the amount of data a node can hold will be limited by the amount the data that can be transfered onto that node, which in turn is limited by the connection durations and transfer rates (see Equation (16) and (17))

a connection node $< 1, 2, 300, 400 >$ to a connection node $< 2, 5, 500, 600 >$ since they share the common object 2 and $300 \leq 600$. The capacity of an edge $(C_x, C_y)$ is defined as

$$Cap_{(C_x,C_y)} = v \cdot \min\{L(C_x), L(C_y)\} \tag{16}$$

where $v$ is the data transfer speed (we assume it to be a constant in the absence of bad weather conditions), and $L(C_z) = Down_z - Up_z$ is the lifetime of connection $C_z$.

When two connections with at least one object in common overlap in time, we will take a conservative approach and assume that overlapping transmissions from two respective connections collide (generate interference) and hence that during the overlap time no successful transmissions occur for these connections. Hence, whenever two connections that share an object overlap, we will not count the overlap time when computing the capacity of the edge between them.

Now we consider the edges in the graph between a commodity node and a connection node. We represent each commodity $F_z$ as a node $F_z = < s_z, Generate_z >$, where $s_z$ is the SN where commodity $F_z$ is generated at time $Generate_z$.. There will be an edge from $F_z$ to connection node $C_x = < A_x, B_x, Up_x, Down_x >$ if and only if $s_z = A_x$ or $s_z = B_x$, and $Generate_z \leq Down_x$. The capacity of a commodity-connection edge $(F_z, C_x)$ is calculated as:

$$Cap_{(F_z,C_x)} = \min\{d, (Down_x - Generate_z) \cdot v\} \tag{17}$$

We also add an edge of infinite capacity between a connection node $C_x$ and a virtual sink node $sink_z$ if and only if either $A_x$ or $B_x$ is equal to destination $t_z$ of commodity $z$. We claim that any valid multicommodity flow on the connection graph (composed possibly of flows that meet only a fraction of the demand for each commodity, i.e., that is not necessarily a maximum ANF) always corresponds to a valid routing schedule on the original time-scheduled DTN network of the same total value, where the value of a (multicommodity) flow is measured by the sum of the flows for all commodities and the value (which corresponds to the total value of all delivered commodities in the DTN).

**Theorem III.1.** *A multicommodity flow that delivers $f_i$ units of flow from commodity node $F_i$ to virtual sink node $sink_i$, for each $i$, in the connection graph directly corresponds to a valid routing schedule for the respective commodities in the original time-scheduled DTN network that delivers the same value $f_i$ for each of the commodities, assuming that all MNs adhere to their regular schedules and that there is no transmission loss.*

*Proof.* Assume that each commodity is partitioned into infinitesimal packets. First note that the amount of flow that can leave any of the commodity nodes is at most equal to the respective commodity demand, given the capacity of the edges out of the commodity node. Second, all the edges in the connection graph respect time, in the sense that they only link connection and commodity nodes respecting the order in which they occur in time; the edges also only exist when the two corresponding endpoints share a common object, so that a feasible sequence of commodity transfers is possible. Thus if we focus on any of the infinitesimal packets that compose the flows, we see that the packet follows a feasible sequence of transfers in the original DTN network, that leads from the respective commodity node to virtual sink node. Hence any valid multicommodity flow in the connection graph corresponds to a feasible routing schedule in the DTN network. Conversely, any routing scheme for the data packets can be converted into a valid multicommodity flow in the connection graph. □

Hence we now have a tool for finding (or approximating) the maximum number of commodities that can be fully delivered in the time-schedule DTN network: We can solve (or approximate) the ANF problem on the respective connection graph. Note that this connection graph can also be used to upper bound the maximum transmission capacity on the DTN network, independent of the ANF setting we consider here.

Figure 1 shows an example of a connection graph and the respective valid multicommodity flow, where just one commodity is fully satisfied, namely commodity $F_1$. We use a solid line to indicate the flow and a dashed line to indicate that the respective edge carries no flow. We omitted edges between connections representing the same pair of objects at different points in time since they are redundant. On each edge, there is a pair of numbers $a(b)$, where $a$ represents the capacity of the edge and $b$ is the actual flow on the edge. There are two commodities $F_1$ and $F_2$ and each commodity has demand 400MB, so they all have edges from the commodity node with capacity 400MB. Commodity $F_1$ is generated from stationary node 1 at time 2:00 so it has edges to connection node $C_2$, which contains stationary node 1. The edge capacity is limited to 400MB by the connection life (40 min) and transmission speed (10MB/min). Commodity $F_2$ is generated from stationary node 3 at time 1:00 so it has edges to connection nodes $C_3$ and $C_6$, which contains stationary node 3.

When two connection nodes overlap, for example, node $C_2 =< 1, 2, 3{:}00, 3{:}40 >$ and node $C_5 =< 2, 5, 3{:}20, 4{:}00 >$, we need to subtract the overlap time, in this case, 20min so the lifetime of connection $C_2$ is $L(C_2) = 20$ min and lifetime of connection $C_5$ is $L(C_5) = 20$ min. Hence, the capacity of the edge between $C_2$ and $C_5$ is 20 min multiplied by the transmission speed (10MB/min) which is equal to 200MB. The resulting multicommodity flow on this example has $f_1 = 400MB$ and $f_2 = 250MB$ and only one commodity, $F_1$, is fully satisfied.

The connection graph can be trivially constructed in polynomial time in the size of the original time-scheduled network.

## IV. Case Study 1: Riverine Amazon Region

In our first case study, we consider a time-scheduled network arising in the context of an interesting communication network in the Brazilian Amazon Delta region.

### A. Scenario

Providing healthcare to the remote and isolated riverine communities in the Brazilian Amazon, which can only be reached

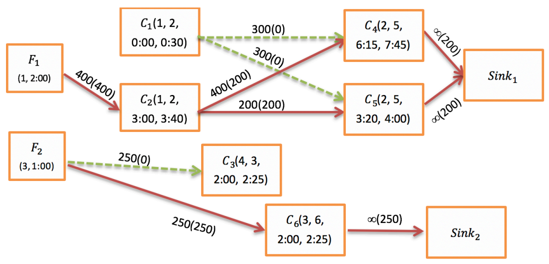

Fig. 1. Example of connection graph and respective max flow.

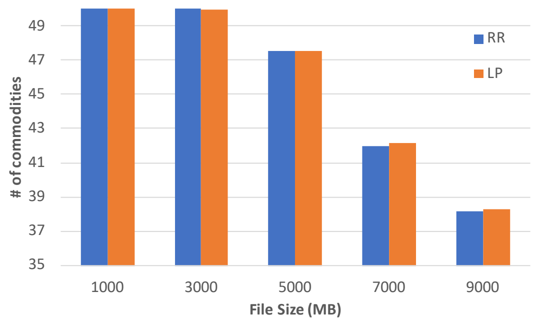

Fig. 2. RR and LP comparison in terms of average commodity delivery in Amazon scenario.

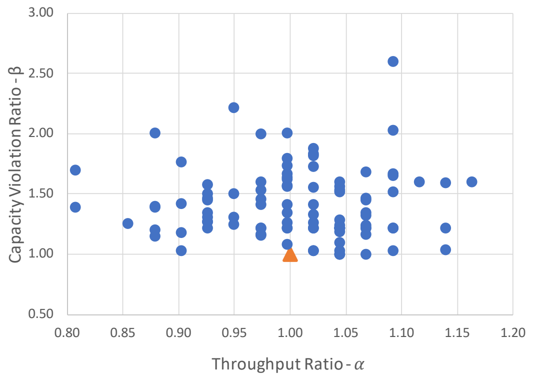

Fig. 3. $(\alpha, \beta)$-distribution for commodity size 7000 MB in Amazon scenario with orange triangle point $\alpha = 1, \beta = 1$.

through the Amazon basin rivers, poses a significant challenge. In those places, healthcare examinations are mainly run by sporadic visits from medical teams from the main city in the region, Belém. Due to the lack of modern communication infrastructure in these communities and of fast river transportation, it is important to fully utilize the regularly scheduled boats as *data mules*, to ensure fast and timely delivery of the examination records which could potentially include examination videos (e.g., ultrasound videos) from those communities to physicians in the city for remote analysis. Since the video files would have to be compressed and split into multiple packets due to the storage limitation of the boats, each file would need to be fully received at destination so that it can be recovered. Furthermore, there would be multiple videos files from different communities each day to different hospitals. We can use our connection graph as described above to transform this scenario into a static network, considering boats as MNs, and remote communities (also referred to as peer base stations) as SNs, and then apply our approximation algorithm. Hence we can view this scenario as an instance of the ANF problem.

In the simulation setup, we use real data schedules for the regular passenger boats serving the Amazon Delta Region in the state of Pará in Brazil [15]. We randomly generate 50 files (each file will be a commodity in $\mathcal{F}$), with a time interval of 60 seconds in between files, hence each file comes with a generation timestamp. In the connection graph, the commodity nodes represent the files in $\mathcal{F}$, and connection nodes are the nodes in $\mathcal{C}$. For each commodity, we create a super sink, and connect it to all the connection nodes, that could potentially deliver the commodity to its destination. The connection graph has 449 connection nodes, and 50 commodity nodes with 50 super sink nodes. Total number of edges is around 16,000 and the edge capacity varies from 200 to 400k due to different time schedules.

## B. Experimental Results

We first solve the relaxed ILP formulation and then use randomized rounding to select the commodities to deliver and then check the edge constraints to see if any edge constraint is violated. We repeat the randomized rounding process 100 times, and report the $(\alpha, \beta)$-distribution and the $(\alpha, \beta)$-distance as we explain later. Note that according to the theoretical bounds given by Corollary II.6, the violation ratio can be proportional to $\sqrt{2 \cdot \log |V| \cdot k}$, which is roughly 25.1 in our case.

The default setting for commodity size we consider is 7,000 MB. We also test with different commodity sizes to find out how the proposed algorithm performs under various settings. Each setting is executed for 10 times and the average commodity delivery is reported. Figure 2 shows the comparison between the number of commodities delivered using our proposed randomized rounding algorithm (RR) and the upper bound provided by the relaxed linear programming solution (which we call LP). As we can see, even though different commodity sizes result in different numbers of commodities received, the ratio stays around 1 which means that the RR solution is basically as competitive as the LP solution.

Figure 3 shows the $(\alpha, \beta)$-distribution of the 100 runs for the Amazon scenario. As we can see, most runs have a competitive throughput ratio $\alpha$ at least 0.8. In terms of edge capacity violation ratio $\beta$, most runs have a ratio between 1 and 2. [2]

Given the $(\alpha, \beta)$-distribution shown in Figure 3, we calculate the Euclidean distances among each $(\alpha, \beta)$ point with respect to the optimal point, which is given by $\alpha = 1, \beta = 1$ (the orange triangle in Figure 3), and plot the distribution of these

[2]Note that the graph in Figure 3 may give the wrong impression that there are points with $\alpha > 1$ and $\beta = 1$, which would imply that we obtain the solution with better throughput than optimal solution without violating edge capacity constraints. However, this is due to the low resolution of the rendering of the graph, and all points with $\alpha > 1$ have $\beta > 1$ in reality.

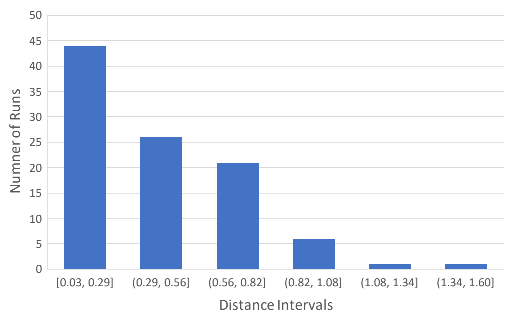

Fig. 4. $(\alpha, \beta)-$distances for commodity size 7000 MB in Amazon scenario.

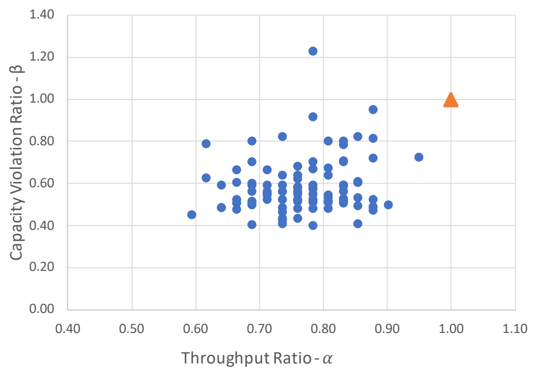

Fig. 5. $(\alpha, \beta)$-distribution for commodity size 7000 MB in Amazon scenario after scaling down edge capacities (by a factor of 2.5) with orange triangle point $\alpha = 1, \beta = 1$.

distances in Figure 4. As we can see, most runs are within a small distance (0.82) to the optimal solution.

While in some applications, a capacity violation for an edge $e$ may entail a penalty or additional charges for using a higher bandwidth than what was agreed with a network provider for a (virtual) edge $e$, in other applications of the ANF problem, the capacity of an edge may pose a strict requirement and cannot be violated. Hence we run some additional simulations where we scale down the capacity of each edge in the network by a factor of 2.5, since the original experiments (Figure 3) show that 95% of the runs have edge capacity violation ratios lower than 2.5. We compare this new scaled down solution with the solution to the original linear relaxation of IP 1 (i.e. $OPT_{LP}$). Figure 5 shows the $(\alpha, \beta)$-distribution of the 100 runs for the Amazon scenario after scaling down each edge capacity by 2.5. The best run achieves throughput ratio $\alpha = 0.95$ and edge violation ratio $\beta = 0.72$, which means the proposed randomized rounding algorithm finds a feasible solution that is competitive with an optimal solution found by the LP without incurring any edge capacity violations.

## V. Case Study 2: German50 Network

In a second case study, we evaluated the achieved throughput and edge capacity violation ratio in a synthetic scenario obtained from [20].

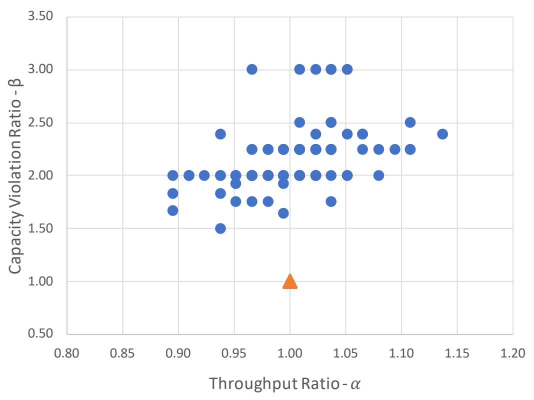

Fig. 6. $(\alpha, \beta)$-distribution for the German50 scenario with orange triangle point $\alpha = 1, \beta = 1$.

### A. Scenario

We use the testbed Germany50 network from SNDlib [20]: In this network, there are 50 nodes and 88 edges, and number of commodities is 662. The average node degree is 3.52. The edge capacity of each edge is 40, and in order to test the proposed method, we set the commodity size to be 50.

### B. Experimental Results

Again, we repeat the randomized rounding process 100 times, and report the following statistics: the $(\alpha, \beta)$-distribution and the $(\alpha, \beta)-$distance. Note that according to the theoretical bounds given by Corollary II.6, the edge capacity violation ratio can be proportional to $\sqrt{2 \cdot \log |V| \cdot k}$, which is roughly 71.9.

Figure 6 shows the $(\alpha, \beta)$-distribution of the 100 runs for the German50 scenario before scaling down the edge capacities. As we can see all runs have a competitive throughput ratio $\alpha$ greater than or equal to 0.9, and most runs have an edge capacity violation ratio between 1.5 and 2.5 (and no run has an edge capacity violation greater than 3).

Figure 7 shows the $(\alpha, \beta)-$distances given the $(\alpha, \beta)$-distribution shown in Figure 6. As we can see, most runs are within a small distance of 1.5 to the optimal solution.

As we did in Section IV-B, we also run experiments with scaled down edge capacities (by a factor of 2.5, since also here 90% of the original runs in Figure 6 have edge capacity violation ratios lower than 2.5) for the German50 network in order to obtain solutions with no edge capacity violations.

Figure 8 shows the $(\alpha, \beta)$-distribution of the 100 runs for the German50 scenario after scaling down edge capacities. As we can see, the best run achieves throughput ratio $\alpha = 0.37$ and edge violation ratio $\beta = 0.89$. This is also aligned with our analysis where the randomized rounding algorithm is able to find a feasible solution with at least $1/3$ of the throughput of the optimal LP solution. However, we now achieve this *without* any edge capacity violations, improving significantly on our theoretical predictions.

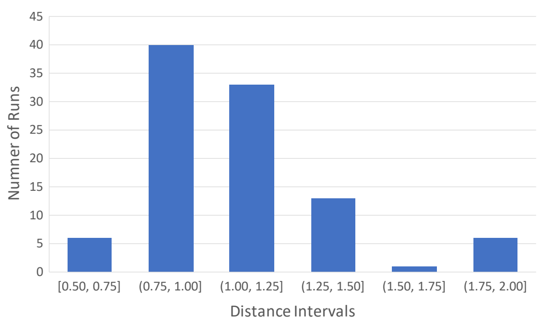

Fig. 7. $(\alpha, \beta)-$distances for the German50 scenario

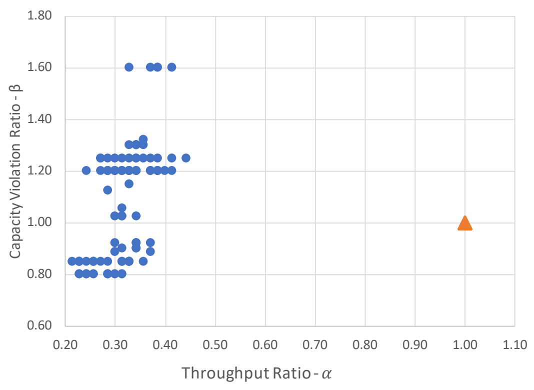

Fig. 8. $(\alpha, \beta)$-distribution for the German50 scenario after scaling down edge capacities (by a factor of 2.5) with orange triangle point $\alpha = 1, \beta = 1$.

## VI. Related Work

The study of routing and multicommodity flow problems is motivated by many real-world applications as well as the important role that flows and cuts play in combinatorial optimization [3]. There are two flavors of optimization problems related to our work: the *Maximum Edge-Disjoint Paths (MEDP)* [10] problem and the *All-or-Nothing Flow (ANF)* [7] problem. ANF is a "relaxed version" of MEDP for non-integral flows: In ANF, the goal is to select a largest subset of commodities that can be simultaneously fractionally routed from source to destination with regard to capacity constraints, whereas in MEDP the flow needs to be integral since the goal is to find the maximum number of edge disjoint paths. In other words, the MEDP problem considers a set of pairs to be routable if they can be connected using edge-disjoint paths; the ANF problem considers a set of pairs to be routable if there is a feasible multicommodity flow that fractionally routes one unit of flow from source to destination for each routed pair. Both problems, MEDP and ANF, are NP-hard. More specifically, ANF is APX-hard even in the case when the underlying graph is a tree, and there exists a 2-approximation algorithm for the tree and 4-approximation algorithm when commodities are associated with weights: Kawarabayashi et al. [13] proposee a constant-approximation algorithm in planar graphs, and also proved that the integrality gap is $O(1)$. Charbonneau et al. [24] studied the MEDP problem in planar graphs, and showed that a constant approximation is possible also with congestion 2, which improves on [6] where the congestion is 4. Chekuri et al. [4] study the multicommodity flow and cut problem in polymatroidal networks, where there are submodular capacity constraints on the edges incident to a node, by analyzing the dual of the flow relaxations via continuous Lovász extension; the underlying graph could be either directed or undirected. Chuzhoy [8] presented an approximation ratio of $\Omega(1/\text{polylog}k)$ with constant congestion of at most 14, using an efficient randomized algorithm in undirected graphs.

There exist several additional results on ANF, which are closely related to our work. The study of the *Symmetric All or Nothing Flow (SymANF)* problem in directed graphs with symmetric demand pairs was initiated by Chekuri et al. in [3]. In SymANF, the input pairs are unordered and a pair $(s_i, t_i)$ is routed only if both the ordered pairs $(s_i, t_i)$ and $(t_i, s_i)$ are routed, and the goal is to find a maximum subset of the given demand pairs that can be routed. The authors provide a poly-logarithmic approximation with constant congestion for SymANF, by extending the well-linked decomposition framework of [5] to the directed graph setting with symmetric demand pairs.

Our work differs from Chekuri et al. [3] in that their results depend on a more restricted assumption of unit edge capacity and symmetric unit demand. Our work considers a more general setting and our result of constant approximation with poly-logarithmic congestion is not directly comparable to theirs. Based on their observation, from previous work on the hardness of the ANF problem, the throughput for the SymANF with constant congestion $c$ is hard to approximate to within a factor of $(\log |V|)^{\Omega(1/c)}$.

The most closely related work to ours is the work in [7], where the authors present an approximation algorithm for the general ANF problem with constant congestion and approximation ratio of $\Omega(1/(\log^3 |V| \log\log |V|))$ based on hierarchical graph decomposition. While Chekuri et al. [7] present an approximation algorithm for the ANF problem which achieves constant $\beta$ and $\alpha$ in $\Omega(1/(\log^3 |V| \log\log |V|))$, our approach keeps $\alpha$ constant and allows $\beta$ not to be constant. It does not seem possible to obtain a constant throughput by modifying the proposed algorithm in [7], without incurring in linear congestion.

Several other applications also involve the ANF, such as [12], which solved the ANF to help for designing route networks for container ships, in particular, the authors studied the ANF with transit time constraints and proved that including time constraints does not necessarily increase the computational time. Finally, our work leverages randomized rounding techniques presented by Rost et al. [22, 23] in the different context of virtual network embedding problems (i.e., there flow endpoints are subject to optimization).

There is a large body of related work for Delay-Tolerant Networks (DTNs), e.g. in the context of Daknet [21]. Here, we focus on the related work on DTNs more closely related to our case study described in Section IV. The work in [15] provides some preliminary simulation results (only, no theoretical bounds or framework) on routing several files (commodities) to a single destination node in the Amazon Delta Region scenario considered in Section IV using a fountain code approach for increasing the robustness of the routing. Other work involves addressing the communication and information access needs of remote rural villages that lack of modern communication technologies, such as MotoPost proposed in [19]. In [1], the authors described a routing algorithm that aims at computing shortest routes based on a stochastic model of real-life bus traces in an urban network. They use buses as data carriers to deliver timely data to its final destination, tackling quasi-deterministic mobility scenarios. Their proposed routing algorithm outperforms other approaches that aim at minimizing the expected traversal time or at maximizing the delivery probability in the bus network. They did not consider direct data transmission between data carriers (buses).

## VII. Conclusion

We presented the first constant-approximation algorithm with sublinear edge capacity violations for the fundamental problem of maximizing throughput for all-or-nothing splittable commodities. We also showed, using simulations, that our algorithm behaves better than our theoretical worst case predictions in different case studies and when applied to scheduling and delay-tolerant networks where nodes have limited buffer size and short contact times. In particular, edge capacity violations are kept below 3 and can be removed entirely when applying rescaling, without sacrificing the throughput by much. We believe that our work provides interesting avenues for future research. In particular, it would be interesting to consider nonuniform commodity demands (since the algorithm proposed in this paper cannot be directly applied to a nonuniform demands scenario and thus new ideas need to be developed). It would be also interesting to consider deterministic (e.g., derandomized) algorithms, and to further improve the tradeoff between approximation quality and augmentation, also investigating lower bounds.

## References

[1] U. G. Acer, P. Giaccone, D. Hay, G. Neglia, and S. Tarapiah, “Timely data delivery in a realistic bus network,” *IEEE Trans. Vehicular Technology*, vol. 61, no. 3, pp. 1251–1265, 2012.

[2] J. Burgess, B. Gallagher, D. D. Jensen, and B. N. Levine, “Maxprop: Routing for vehicle-based disruption-tolerant networks,” in *INFOCOM*, 2006, pp. 1–11.

[3] C. Chekuri and A. Ene, “The all-or-nothing flow problem in directed graphs with symmetric demand pairs,” *Math. Program.*, vol. 154, no. 1-2, pp. 249–272, 2015.

[4] C. Chekuri, S. Kannan, A. Raja, and P. Viswanath, “Multicommodity flows and cuts in polymatroidal networks,” *SIAM J. Comput.*, vol. 44, no. 4, pp. 912–943, 2015.

[5] C. Chekuri, S. Khanna, and F. B. Shepherd, “Multicommodity flow, well-linked terminals, and routing problems,” in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, 2005, pp. 183–192.

[6] ——, “Edge-disjoint paths in planar graphs with constant congestion,” *SIAM Journal on Computing*, vol. 39, no. 1, pp. 281–301, 2009.

[7] ——, “The all-or-nothing multicommodity flow problem,” *SIAM J. Comput.*, vol. 42, no. 4, pp. 1467–1493, 2013.

[8] J. Chuzhoy, “Routing in undirected graphs with constant congestion,” *SIAM J. Comput.*, vol. 45, no. 4, pp. 1490–1532, 2016.

[9] D. P. Dubhashi and A. Panconesi, *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[10] T. Erlebach and K. Jansen, “The maximum edge-disjoint paths problem in bidirected trees,” *SIAM Journal on Discrete Mathematics*, vol. 14, no. 3, pp. 326–355, 2001.

[11] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay-tolerant networks,” *IEEE TMC’11*, vol. 10, no. 11, pp. 1576–1589.

[12] C. V. Karsten, D. Pisinger, and S. R. B. D. Brouer, “The time constrained multi-commodity network flow problem and its application to liner shipping network design,” *Transportation Research Part E*, vol. 76, pp. 122–138, 2015.

[13] K.-i. Kawarabayashi and Y. Kobayashi, “All-or-nothing multicommodity flow problem with bounded fractionality in planar graphs,” *SIAM Journal on Computing*, vol. 47, no. 4, pp. 1483–1504, 2018.

[14] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram, “Delay tolerant bulk data transfers on the internet,” in *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1. ACM, 2009, pp. 229–238.

[15] M. Liu, T. Johnson, R. Agarwal, A. Efrat, A. Richa, and M. M. Coutinho, “Robust data mule networks with remote healthcare applications in the amazon region: A fountain code approach,” in *17th International Conference on E-health Networking, Application & Services, HealthCom*, 2015, pp. 546–551.

[16] A. Mei, G. Morabito, P. Santi, and J. Stefa, “Social-aware stateless forwarding in pocket switched networks,” in *IEEE INFOCOM*, 2011, pp. 251–255.

[17] M. Mitzenmacher and E. Upfal, *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[18] A. Moghadam and H. Schulzrinne, “Interest-aware content distribution protocol for mobile disruption-tolerant networks,” in *IEEE WoWMoM*, 2009, pp. 1–7.

[19] S. Naidu, S. Chintada, M. Sen, and S. Raghavan, “Challenges in deploying a delay tolerant network,” in *Proceedings of the third ACM workshop on Challenged networks*. ACM, 2008, pp. 65–72.

[20] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, “SNDlib 1.0–Survivable Network Design Library,” in *Proceedings of the 3rd International Network Optimization Conference INOC*, April 2007, http://sndlib.zib.de, extended version accepted in Networks, 2009.

[21] A. Pentland, R. Fletcher, and A. Hasson, “Daknet: Rethinking connectivity in developing nations,” *IEEE Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[22] M. Rost, E. Döhne, and S. Schmid, “Parametrized complexity of virtual network embeddings: Dynamic & linear programming approximations,” in *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 49, no. 1, 2019.

[23] M. Rost and S. Schmid, “Virtual network embedding approximations: Leveraging randomized rounding,” in *Proc. IFIP Networking*, 2018.

[24] L. Seguin-Charbonneau and F. B. Shepherd, “Maximum edge-disjoint paths in planar graphs with congestion 2,” in *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS*, 2011, pp. 200–209.