Multicommodity Flow on Connection Graph in Time Schedule Networks

***Abstract*—Many real world scenarios such as boat or bus transportation networks, include fixed time schedules of moving nodes (boats or buses), and these nodes strictly follow their schedules and move within the given network. Given the successful development of IoT devices, it is much easier to transfer data among nodes with less energy consumption, as well as less delivery latency. This provides a way to use the nodes as data mules and deliver data from place to place and take advantage of these predefined schedules if the underlying network could be delay tolerant. However, since the buffer sizes and the contact time period of nodes are limited, it is difficult to get everything delivered to destinations due to the small capacity of the opportunistic connection. In order to maximize the data that could be delivered in the network, data with less probability of being fully delivered could be ignored so that it makes room for data with higher delivery probability (so that the original whole data could be restored).**

In this paper, we first proposed a connection graph model that can be used to describe the connectivity among moving nodes that have time schedules over time. Given their movement schedules, a time expanded contact opportunistic network could be obtained. Secondly, we proposed an all-or-nothing splittable multi-commodity formulation model data delivery in the connection graph and maximize data delivery, and furthermore, we present a polynomial time approximation algorithm using randomized rounding based approach that can achieve constant approximation ratio of optimal solution and poly-logarithmic congestion with high probability.

I. Introduction

We consider the scenario where there is an underlying Delay Tolerant Network (DTN) consisting of a set of moving nodes and a set of stationary nodes. Each moving node has a predefined route and time schedule that it will strictly follow. Moving nodes are also equipped with wireless communication devices, which can be used for data transmission. Stationary nodes are stations that moving nodes can stop by, upload and download data from, and only few of these stationary nodes can be connected to Internet, which are called Hotspots. Such scenarios are frequently seen in other researches, such as [1, 2, 9]. Furthermore, the similar concept is also widely seen in the social networks, a specific type DTN is call Mobile Social Networks, where the moving nodes are people, stationary nodes could be classrooms, restaurants, shopping malls,etc.

Since in these networks, the network structure is dynamic in time, traditional max flow methods cannot be directly applied, we need to modify it into a static graph model so that we could calculate the optimal data delivery. Moreover, in these networks, end-to-end paths are usually unstable, and hence the nodes need to employ the "store, carry and forward" strategy in order to deliver the data from source to destination. However, if the schedules of these moving nodes are known ahead of time, we can actually take advantage of these knowledge and make smart decisions on choosing the data to be delivered and the routes of data without violating capacity constraints and time schedules.

Realizing these challenges, we develop a randomized rounding based algorithm to solve the All or Nothing splittable multicommodity flow problem in directed graph efficiently and effectively. The main contributions of this paper are three-fold:

- We present a standard ILP formulation for the All or Nothing splittable multicommodity flow problem in directed graph. (Section IV)
- We then developed a randomized rounding based approach that can ensure a constant approximation ratio with high probability so that the multicommodity flow problem can be solved efficiently in polynomial time with poly-logarithmic congestion.

The remainder of the paper is organized as follows: in Section II, we introduce some background information related to the network model, and also some of most recent work on multicommodity flow problem. In Section IV, we derive an ILP formulation of the All-or-Nothing splittable multicommodity problem with the randomized rounding based approach to solve the problem in polynomial time. For the experiments, to use the real world dataset, we present the connection graph model in Sections **??**, which transforms the original dynamic node network into a contact based connectivity graph, and Section VI concludes our work.

II. Related Work

In this section, we review the literature on different models and routing algorithms.

In the recent years, MultiCommodity Flow Problem (MCFP) has drawn significant attention from the researcher community and has been extensively studied. In this multicommodity flow field there are two well-studied optimization problems: the Maximum Edge Disjoint Paths (MEDP) and the All-or-Nothing Flow (ANF) problem and they are both NP-hard problems. In MEDP, it is is a well known APX-hard even in the case of the underlying graph is a tree, and there existing a 2-approximation for tree. In [6], the authors proposed an $O(1)$ approximation algorithm for the all-or-nothing multi-commodity flow problem in planar graphs, and also proved that the integrality gap is $O(1)$. In [4], the authors studied the multicommodity flow and cut problem in polymatroidal networks, where there are submodular capacity constraints on the edges incident to a node.

The underlying graph could be either directed or undirected, by analyzing the dual of the flow relaxations via continuous extensions of the Lovász extension. In ANF, usually viewed as a relaxed version of MEDP, 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 find maximum number of edge disjoint paths.

Over the last decade there are several non-trivial existing work about ANF, which are closely related to our work. The study of Symmetric All or Nothing Flow problem (SymANF) in directed graphs with symmetric demand pairs is initiated 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. However, their results depend on a more restricted assumption of unit capacity constraint and unit demand. Our work differs from them with a more general setting and our result of constant approximation with poly-logarithmic congestion is not directly comparable to theirs.

Several other applications also involve the MCFP, such as [8], solve the MCFP to help for designing route networks for container ships, in particular, the authors studied the MCFP with transit time constraints and proved that including time constraints does not necessarily increase the computational time.

III. Problem Statement

Assume that we are given a directed graph with a set of nodes $\mathcal{N}$ and edges $\mathcal{M}$, and we are also given a set of commodities $\mathcal{F} = \{F_1, ..., F_n\}$ with equal size s. Each commodity $F_i \in \mathcal{F}$ is a tuple $F_i = (sv_i, dv_i)$ where sv_i, dv_i denote the source and destination for that commodity. Commodity F_i is successfully transmitted if s units of it are transmitted from sv_i to dv_i Assume that all commodity can be fully fractionally delivered alone, which means each commodity can always be delivered in the form of splittable packets in the absence of other commodities. The goal is to maximize the number of commodities that could be potentially routed from their sources to destinations.

IV. All-or-Nothing Splittable Multi-Commodity Flow

In this section, we will describe in details of the proposed random rounding based approach for solving the multi-commodity flow problem.

Formulation 1: Maximize the total number of commodities that can be successfully delivered.

We use a variable f_i to represent whether or not F_i is successfully delivered. For the sake of simplicity, we rescale each commodity size and edge capacity in the network by s so that this problem thus becomes an unit flow multi-commodity problem.

Algorithm 1 Random Rounding Algorithm for Formulation 1

Input:
Directed Connection Graph $G(V, E)$;
Commodities $\mathcal{F} = \{F_1, ..., F_n\}$;
Source-sink pair (sv_i, dv_i) for each $F_i \in \mathcal{F}$;
Capacity $c(u, v),\ \forall (u, v) \in E$;
Output: OPT_{ALG}

1: Change the last constraint to be $0 \le f_i \le 1$;
2: Then it is relaxed to an LP, solve this LP and get 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 within a certain fraction of the optimal solution, return this solution, otherwise, repeat step 3 and 4, at most N times.

$$\begin{array}{ll} \text{Input: Directed graph } G(V,E), & \text{Commodities } \mathcal{F} = \{F_1, ..., F_n\} \\ \text{Source-sink pair: } (sv_i, dv_i), & \forall F_i \in \mathcal{F} \\ \text{Capacity: } c(u,v), & \forall (u,v) \in E \\ \text{Variable: Flow } f_{i,(u,v)}, & \forall F_i \in \mathcal{F}, (u,v) \in E \\ \qquad\qquad f_i, & \forall F_i \in \mathcal{F} \end{array}$$

$$\begin{aligned} \text{Maximize } & \sum_{i=1}^{n} f_i \\ \text{Subject to } & \sum_{(sv_i,v)\in E} f_{i,(sv_i,v)} = f_i, \quad \forall F_i \in \mathcal{F} \\ & \sum_{(u,v)\in E} f_{i,(u,v)} = \sum_{(v,u)\in E} f^i_{v,u}, \ \forall F_i \in \mathcal{F},\ \forall v \in V - \{sv_i, dv_i\} \\ & \sum_{i=1}^{f} f_{i,(u,v)} \le c_{(u,v)}, \quad \forall (u,v) \in E \\ & f^i_{(u,v)} \ge 0, \quad \forall F_i \in \mathcal{F}, \forall (u,v) \in E \\ & f_i \in \{0,1\}, \quad \forall F_i \in \mathcal{F} \end{aligned}$$

Let OPT_i be the optimal solution of the IP in Formulation 1, and let OPT_f be the total amount of commodities delivered by solving the linear relaxation LP of IP in Formulation 1 where the variables f_i are relaxed to assume any value in $[0, 1]$. It is obvious to see that $OPT_f = \sum \tilde{f}_i$. Define the solution from the above algorithm as ALG, and total amount of commodities delivered by ALG as OPT_{ALG}.

We use Chernoff Bound over continuous random variables to bound the probability of achieving a fraction of the optimal solution.

Fact 1 (Chernoff-Bound) . 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 $P(X < (1 - \epsilon) \cdot E(X)) \le exp(-\epsilon^2 \cdot E(X)/2)$ holds for $0 < \epsilon < 1$.

Since we have already scaled down the flow by s_r, thus the variables are between 0 and 1 so that we can apply the above Chernoff Bound.

Claim 2. $Pr[OPT_{ALG} < (1-\epsilon)\cdot OPT_f] \leq e^{-\epsilon^2 \cdot OPT_f/2}$

Proof: For each commodity i, the expectation of f_i is $E(f_i) = 1\cdot \tilde{f}_i + 0\cdot(1-\tilde{f}_i) = \tilde{f}_i$. Recall that $OPT_f = \sum \tilde{f}_i$, and let $OPT_{ALG} = \sum f_i$.

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

□

Since we also assume that all commodity can be fully fractionally delivered alone, therefore, we get:

$$OPT_f \geq 1 \tag{2}$$

Since we have proved that the optimal fractional solution is an upper bound on optimal solution for the ILP problem, by taking $\epsilon = 2/3$ in the Chernoff Bound, we get,

Theorem 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 $\epsilon = 2/3$, Equation 1 becomes:

$$\begin{aligned}
Pr[OPT_{ALG} < \frac{1}{3}\cdot OPT_f] &\leq e^{-(\frac{2}{3})^2 \cdot OPT_f/2} &(3)\\
&= e^{-2\cdot OPT_f/9} &(4)
\end{aligned}$$

By Equation 2, we know that the minimum value of OPT_f is 1, by taking $OPT_f = 1$, we get the upper bound of $e^{-\epsilon^2 \cdot OPT_f/2}$. Therefore,

$$Pr[OPT_{ALG} < \frac{1}{3}\cdot OPT_f] \leq e^{-2/9} \tag{5}$$

Since $OPT_i \leq OPT_f$, thus we get,

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

□

Fact 4 (Hoeffding's Inequality)[7] . Let $\{X_i\}$ be independent random variables, s.t. $X_i \in [a_i, b_i]$, then $Pr(\sum_i X_i - E(\sum_i X_i) \geq t) \leq exp(-2t^2/\sum_i (b_i - a_i)^2)$ holds.

Theorem 5 Given a single edge e with capacity c_e, and let $\Delta_{F,e}$ be the commodities going through edge e. For all the commodity $i \in \Delta_{F,e}$, choose ϵ' such that $\sum_i \frac{\tilde{f_{i,e}}}{\tilde{f}_i} \leq \epsilon' \cdot c_e$ .The probability that ALG exceeds the edge capacity constraint by a factor of $\gamma = (1 + \epsilon' \cdot \sqrt{2\log|V|})$ is bounded by $|V|^{-4}$.

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

Proof: Fix an edge $e \in E$, for 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

$$E(f_{i,e}) = \tilde{f_{i,e}} \cdot \frac{1}{\tilde{f}_i}\cdot \tilde{f}_i + 0\cdot(1-\tilde{f}_i) = \tilde{f_{i,e}} \tag{6}$$

Let F_e denotes the flow on edge e by ALG, then $F_e = \sum_{i, f_{i,e}\neq 0} f_{i,e}$ and the expectation of F_e is

$$E[F_e] = \sum_{i, f_{i,e}\neq 0} \tilde{f_{i,e}} \cdot \frac{1}{\tilde{f}_i}\cdot \tilde{f}_i = \sum_{i, f_{i,e}\neq 0} \tilde{f_{i,e}} \tag{7}$$

Since we relax the LP, and a feasible solution must obey edge capacity constraint, so the cumulative load on edge e is equal or less than the capacity of e:

$$\sum_{i, f_{i,e}\neq 0} \tilde{f_{i,e}} \leq c_e \tag{8}$$

Therefore, we can get the following:

$$E[F_e] \leq c_e \tag{9}$$

Let $t = \epsilon' \cdot \sqrt{2\log|V|}\cdot c_e$, by applying Hoeffding's Inequality and since $\sum_i \frac{\tilde{f_{i,e}}}{\tilde{f}_i} \leq \epsilon' \cdot c_e$, we get:

$$\begin{aligned}
Pr[F_e - E(F_e) \geq t] &\leq exp(\frac{-2t^2}{\sum_i (\frac{\tilde{f_{i,e}}}{\tilde{f}_i})^2}) &(10)\\
&\leq exp(\frac{-2\cdot \epsilon'^2 \cdot \log|V| \cdot c_e^2}{\epsilon'^2 \cdot c_e^2})\\
&= |V|^{-4}
\end{aligned}$$

Given Equation 9, and let $\gamma = (1 + \epsilon' \cdot \sqrt{2\log|V|})$,

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

There are at most $|V|^2$ edges, by applying union bound over all edges using Theorem 5, hence we obtain the following corollary.

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

Based on Theorem 3 and Corollary 6, if $|V| \geq 3$ holds, the probability of not finding a suitable solution, satisfying the objective and the capacity constraint, within a single round is therefore upper bounded by $exp(2/9) + 1/9 \leq 11/12$, . The probability to find a suitable solution within N many rounds is then lower bounded by $1 - (11/12)^N$ for $|V| \geq 3$ and hence the randomized rounding scheme yields a solution with high probability.

V. Experimental Evaluations

In this section, we will evaluate our proposed algorithm on both real world dataset and synthetic dataset.

A. Real World Dataset: Amazon Delta Region

1) Connection Graph Model: In order to evaluate on the real world traces, we employ the scenario described in [9] and create a connection graph model that transforms the original dynamic node network into a contact based connectivity graph.

Denote by $\mathcal{B}$=$\{B_1, B_2, B_3, ..., B_m\}$ and $\mathcal{P}$=$\{P_1, P_2, P_3, ..., P_n\}$ the sets of m boats and the set of n PBSs in the CoDPON network, respectively. We assume both the boats and the PBSs have infinite buffer sizes[1]. Let DP_{B_i} be the displacement plan associated with each boat B_i, which describes the routing paths and times to arrive and depart the PBSs (we assume that each boat has full knowledge of its own displacement plan). A connection is established between two boats B_i and B_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 boats 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 B_i and B_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 boat B_z and (x_j, y_j) at time t.

Based on the original boat displacement plans, we can then determine all boat-to-boat and boat-to-PBS connections $\mathcal{C}$=$\{C_1, C_2, C_3, ..., C_k\}$ where $k \leq cm(m + n)$ and 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 B_i are the two objects (each object is either a boat or PBS) 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 the directed connection graph using boat-to-boat and boat-to-PBS connections and files as nodes, therefore the node set is $V = \mathcal{C} \cup \mathcal{F}$ and $|V| = (k + f)$. 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 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 $e_i = (C_x, C_y)$ is defined as

$$Cap_{e_i} = min\{M(A_x, B_x), M(A_y, B_y)\}$$

where $M(A_x, B_x) = L(C_x) \cdot v$, v is the data transfer speed (we assume it to be a constant in the absence of decent weather conditions), and $L(C_x) = Down_x - Up_x$ is the lifetime of C_x.

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 file node and a connection node. WE represent each file F_k as a node $F_k =< p_k, Generate_k >$, where p_k is the PBS of the respective community where F_k was generated, and $Generate_k$ is the time when F_k was generated. There will be an edge from F_k to connection node $C_x =< A_x, B_x, Up_x, Down_x >$ if and only if $p_k = A_x$ or $p_k = B_x$, and $Generate_k \leq Down_x$. The capacity of a file-connection edge (F_k, C_x) is calculated as:

$$Cap_{(F_k, C_x)} = min\{FileSize, (Down_x - Generate_k) \cdot v\}$$

Then, we can transform the original dynamic network into the connection graph model, which is a static network, and our proposed method can be directly applied.

B. Synthetic Data

VI. Conclusions

Realizing the challenge of

[1] A reasonable assumption, given the ever-dropping costs of memory and also the much more stringent data transmission bottlenecks due to boat-to-boat transmission while in transit.

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. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 23-29 April 2006, Barcelona, Catalunya, Spain*, 2006.

[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, Baltimore, MD, USA, May 22-24, 2005*, 2005, pp. 183–192.

[6] J. Chuzhoy, "Routing in undirected graphs with constant congestion," in *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, 2012, pp. 855–874.

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

[8] 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.

[9] 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, Boston, MA, USA, October 14-17, 2015*, 2015, pp. 546–551.