

Maximizing Resilient Throughput in Peer-to-Peer Network: A Generalized Flow Approach

Bin Chang, Yi Cui, Yuan Xue

Department of Electrical Engineering and Computer Science
Vanderbilt University

Email: {bin.chang, yi.cui, yuan.xue}@vanderbilt.edu

Abstract—A unique challenge in P2P network is that the peer dynamics (departure or failure) cause unavoidable disruption to the downstream peers. While many works have been dedicated to consider fault resilience in peer selection, little understanding is achieved regarding the solvability and solution complexity of this problem from the optimization perspective. To this end, we propose an optimization framework based on the generalized flow theory. Key concepts introduced by this framework include resilience factor, resilience index, and generalized throughput, which collectively model the peer resilience in a probabilistic measure. Under this framework, we divide the domain of optimal peer selection along several dimensions including network topology, overlay organization, and the definition of resilience factor and generalized flow. Within each subproblem, we focus on studying the problem complexity and finding optimal solutions. Simulation study is also performed to evaluate the effectiveness of our model and performance of the proposed algorithms.

I. INTRODUCTION

Peer-to-peer (P2P) has been proved a highly cost-effective content distribution solution, where peers self-organize themselves into an overlay network and relay data to each other, thus reducing server load. A central problem in the overlay network construction is *peer selection*, the strategy a peer employs to select other peer(s) as its parent(s) to receive data from. Peer selections aggregate into multicast tree(s) spanning from the server, the source of the data, to all peers. Given the data-intensive nature of P2P applications (e.g., video streaming or bulk data distribution), a common objective is to maximize the data throughput to all peers.

Already challenging in its static setup, the optimal peer selection problem is further aggravated by the high volatility of the P2P network. Due to various reasons such as user leaving or machine/network failure, unscheduled peer departure constantly happens, which results in service disruptions or outages on all the downstream peers. Therefore, we argue that when designing peer selection solutions, fault resilience deserves the same level of attention as first-class performance metrics, e.g. throughput, delay, etc.

A significant amount of research has been conducted on this topic with different emphasis. While important heuristics have been proposed such as bandwidth first, age-first, or a hybrid of the two[1], some analytical works [2], [3] have tried to analyze and compare their performances under stochastic framework or real-system traces. However, this domain has been rarely examined from the optimization perspective. If we are able to model the fault-resilient peer selection problem under an

optimization framework which combines fault resilience with key performance metrics such as throughput, standard optimization techniques can be practiced to evaluate key questions such as the solvability of the problem and the complexity of its optimal solutions, if any. Also existing heuristics could be quantitatively evaluated under the same framework.

In this paper, we report our initial research towards this direction. Our optimization framework is based on the *generalized flow* theory [4], [5], [6]. It generalizes the classical network flow problem by specifying a gain factor to each link in the network. As such, the objective is to optimize the throughput of the generalized flow as the product of raw flow and the gain factor on each link, while the traditional capacity and flow conservation constraints still apply to the raw flow. Widely employed in operation research to model the loss, theft, or interest rate in commodity transportation, we find it a good match to the P2P domain. If we assign each peer a resilience factor as the probabilistic measure of its chance of survival within a given time horizon, this resilience factor could be considered as the gain factor in the generalized flow setting. Under this framework, the problem of fault-resilient peer selection becomes to maximize the aggregation of generalized flow received by each peer, which is the product of the raw flow and resilience factors of peers it passes along.

We study this problem under a multitude of problem settings. Specifically,

- Regarding network model, we consider two types of topologies: the general topology which models the underlying physical network as a graph, and the star topology which assumes the bottleneck does not exist in the physical network, but only on peer's access link.
- Regarding overlay organization, we consider cases where the number of trees interconnecting peers is unlimited or upper-bounded, e.g., single tree.
- Along the dimension of generalized flow definition, we consider concatenation model where the generalized flow delivered to a peer depends on the resilience of all its ancestors, and non-concatenation model which only considers the resilience of its immediate parent.

Along these dimensions, we explore the entire spectrum of this domain, and focus on studying problem complexity and finding the optimal solutions within each subproblem. The rest of this paper is organized as follows. In Sec. II,

we introduce our optimization framework and formally define key concepts such as generalized flow and resilience index. In Sec. III, we study the optimal generalized throughput problem from several different perspectives. Sec. IV presents evaluation results. Finally, we conclude in Sec. V.

II. FRAMEWORK OVERVIEW

A. Network Model

We consider two kinds of network models: star model and general model.

1) *General Network*: We model the network as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with capacity c_e on each physical edge $e \in \mathcal{E}$. On top of \mathcal{G} , an overlay network $G = (s, V, L)$ exists, where s is the server, and peers belong to the set $V = \{v\}$. Each overlay edge $l \in L$ connects two peers in V , and corresponds to the unicast route at the physical network \mathcal{G} .

2) *Star Network*: Many works have implicitly assumed that the bottleneck of a unicast path only happens at either access link of its two end hosts. In this way, we can simplify the general model into a star model. The central node of the star represents the Internet cloud, which reaches out to every peer. In this model, we denote the outbound bandwidth of peer $v \in V$ as c_v .

B. Overlay Organization

To transfer data among peers, the simplest and most straight forward strategy is a single multicast tree spanning from the server s to all peers in V . Although simple to manage, this solution has clear drawback since a peer departure can cause complete disruption to all its descendants.

An alternative solution is the recently popular multi-tree or mesh solution, where each peer schedules to receive data from multiple parents. Since the mesh structure can be usually decomposed as the sum of multiple spanning trees, therefore will be categorized as multi-tree solution¹. We denote the tree set as $T = \{t\}$, where each tree $t \in T$ covers all peers and has a single rate $f(t)$.

C. Resilience Factor and Generalized Flow

We assign a resilience factor r_v ($0 < r_v \leq 1$) to each peer $v \in V$. Our model makes no assumption on how r_v is defined. For the purpose of illustration, we introduce one way to define r_v . Suppose v follows certain lifetime distribution with c.d.f. $F(\tau)$, and T is a random variable denoting the time of departure, then the survival function of v is $1 - F(\tau) = Pr(T > \tau)$, the probability that its time of departure is later than time τ . If we denote $r_v = Pr(T > \tau^*)$, where τ^* is a fixed time point in the future, then it represents the chance of survival for v until τ^* .

Given the resilience factor of v , we consider two models to compute the rate of generalized flow.

1) *Concatenation Model*: For each peer v in tree t , there is a path from the server s to v , denoted as

$$\mathcal{P}_t(v) : s \mapsto v_1 \mapsto v_2 \mapsto \dots \mapsto v_k \mapsto v$$

Given t 's flow rate $f(t)$, the dependency model computes the generalized flow delivered to v as $f(t)$ timed by the concatenate product of r_{v_1} through r_{v_k} . We define such product as the resilience index of v in t :

$$R_t(v) = \prod_{i=1}^k r_{v_i} \quad (1)$$

Based on this definition, $f(t)R_t(v)$ is the generalized flow rate delivered to v in tree t . We can further define the resilience index of tree t as

$$R(t) = \sum_{v \in V} R_t(v) \quad (2)$$

Now we are able to define generalized throughput of t , which is the sum of generalized flow rates to all peers.

$$f_g(t) = f(t)R(t) \quad (3)$$

This model computes a peer's generalized flow by factoring in the resilience factors of all its ancestors. It fits the live P2P streaming scenario where a peer failure can cause disruptions on all its descendants. Also an implicit assumption in the definition of $R_t(v)$ is that the resilience factor of server s is 1, i.e., s will not departure.

2) *Non-Concatenation Model*: In this model, we define the generalized flow to a peer to be only dependent on its immediate parent. Formally, in the same sample context of concatenation model, we define the resilience index of peer v in tree t as follows.

$$R_t(v) = r_{v_k} \quad (4)$$

This model fits better to P2P applications with no real time constraints. For example, in some on-demand streaming and downloading applications, the parent peer serves its children from its local cache. This gives its children buffering time to find new parent(s) upon its own departure or failure, thus absorbing the impact of cascading disruption.

D. Summary of Contributions

In Tab. I, we summarize findings when exploring along the three dimensions outlined in this section. Of the eight subproblems, we find four of them polynomially solvable and present the optimal solutions. Of the four NP-hard problems, we are able to find a $O(\log \mathcal{E})$ -approximation algorithm, and only find heuristics to the other three.

We finally summarize notations appeared in this paper in Tab. II.

III. OPTIMIZING GENERALIZED THROUGHPUT

In this section, we present our study on optimal generalized throughput under both general and star topology models. Due to space constraint, all the algorithms and the proofs of all theorems can be found in our technical report [9].

¹We note that such categorization does not apply to the management of P2P network, but only suits the purpose of calculating throughput to each peer, which is the main focus of this paper.

	General Topology	Star Topology
Multiple Trees (Concatenation)	NP-hard (reduction to (α, β) -LAST)[7]	MultiTrees-Star , $O(n)$
Multiple Trees (Non-Concatenation)	MultiTrees-General , $O(\frac{ \mathcal{E} }{\epsilon^2} \log U \cdot T_{mst})$	MultiTrees-Star , $O(n)$
Single Tree (Concatenation)	NP-hard (reduction to MPSP[8], $O(\log \mathcal{E})$ -approximation)	NP-hard (reduction to Hamilton Path[4])
Single Tree (Non-Concatenation)	NP-hard (linear-programming-relaxation is NP-hard)	SingleTree-Star , $O(n^3)$

TABLE I
SUMMARY OF FINDINGS

Notation	Definition
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	Physical Network
$G = (V, L)$	Overlay Network
s	server node
$\mathcal{E} = \{e\}$	physical layer edges
$L = \{l\}$	overlay layer links
$V = \{v\}$	overlay nodes
r, R	resilience index, e.g. r_v, r_l, R_t
$T = \{t\}$	overlay multicast trees
$f(t)$	data flow over tree t
$f_g(t)$	generalized flow over tree t
c	bandwidth constraint, e.g. c_v, c_e, c_s
d_e	price of edge e
$\mathcal{P}_t(v)$	overlay routing path between s and v in overlay tree t

TABLE II
NOTATIONS TABLE

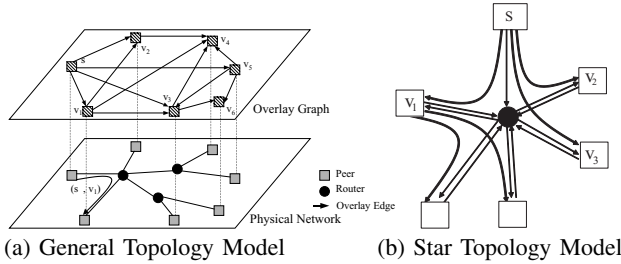


Fig. 1. Two Types of Topology Models

A. Multiple Trees Under General Topology Model

We start from the general topology model with the most basic setting, where an unlimited number of trees can be constructed for the purpose of maximizing generalized throughput. With notions introduced in Sec. II, we formulate it into the following linear programming (LP) problem.

$$\text{maximize } \sum_{t \in T} f(t)R(t) \quad (5)$$

$$\text{subject to } \sum_{t \in T} n_e(t)f(t) \leq c_e, \forall e \in \mathcal{E} \quad (6)$$

$$f(t) \geq 0, t \in T$$

$n_e(t)$ is an integer variable indicating the number of times tree t has passed through e . Note since t is an overlay tree, $n_e(t)$ can be greater than 1. The central difficulty of problem (5) is that its number of variables is exponential to the size of the P2P network. On the other hand, the dimensionality of this problem, i.e., the number of constraints, is $|\mathcal{E}|$, the number of physical links. This gives us a chance to solve this problem via its dual presented as follows, which contains $|\mathcal{E}|$ variables

but exponential constraints.

$$\text{minimize } \sum_{e \in \mathcal{E}} c_e d_e \quad (7)$$

$$\text{subject to } \sum_{e \in \mathcal{E}} n_e(t)d_e \geq R(t), \forall t \in T \quad (8)$$

$$d_e \geq 0, e \in \mathcal{E}$$

Although there exists exponential number of trees in T , if we can find a separation oracle able to check whether constraint (8) is met in polynomial time, then the dual problem (7) is solvable in polynomial time, hence the primal problem.

To find if such an oracle exists, we first adapt the definition of $R(t)$ from peer-based to link-based, to be consistent with the left side of constraint (8). This can be easily achieved as follows. We assign a resilience factor r_e to each link $e \in \mathcal{E}$, and define it as

$$r_e = \begin{cases} r_v & \text{if } e \text{ exits from } v \\ 1 & \text{otherwise} \end{cases} \quad (9)$$

As articulated in Sec. II-C, we have different definitions on $R(t)$ for concatenation and non-concatenation models. Here we focused on the non-concatenation model.

Based on the definition on resilience index $R_t(v)$ shown in Eq. (4), we can easily observe that $R(t)$ in this case is the sum of resilience factors of all non-leaf peers in tree t . Translated into the link-based definition, it is the sum of resilience factors of all links in t , i.e., $R(t) = \sum_{e \in t} n_e(t)r_e$. This allows us to reformulate Inequality (8) into the following.

$$\sum_{e \in \mathcal{E}} n_e(t)d_e \leq \sum_{e \in \mathcal{E}} n_e(t)r_e, \forall t \in T$$

It is now clear that the separation oracle is a minimum spanning tree algorithm that sees the cost on each link e as $(d_e - r_e)$. Constraint (8) will be satisfied if the cost of the found minimum spanning tree is still greater than 0.

To this end, we design a fully polynomial time approximation scheme (FPTAS). FPTAS is a family of algorithms which finds a ϵ -approximate solution returning a result at least $(1 - \epsilon)$ times the maximum value, for arbitrary error parameter $\epsilon > 0$. Based on the optimization scheme proposed in [10], we design the **MultiTrees-General** algorithm.

Theorem 1: Under the non-concatenation model, when $\beta = \frac{(1+\epsilon)^{1-1/\epsilon}}{U^{1/\epsilon}}$, the **MultiTrees-General** algorithm returns the solution at least $(1 - 2\epsilon)$ times the optimal result of problem

(5), with running time is $O(\frac{|\mathcal{E}|}{\epsilon^2} \log U \cdot T_{mst})$. U is the length of the longest unicast route and T_{mst} is the running time of the minimum spanning tree algorithm.

B. Multiple Trees Under Star Topology Model

Now we study the multiple trees solution under the star topology model, as shown in Fig 1 (b). To simplify the illustration, we remove notations associated with the general network \mathcal{E} . Instead, we introduce notations c_v to denote outbound bandwidth of peer $v \in V$, c_s to denote outbound bandwidth of the server s , and $n_v(t)$ or $n_s(t)$ to denote the number of children v or s have in tree t . The problem formulation is as follows².

$$\text{maximize } \sum_{t \in T} f(t)R(t) \quad (10)$$

$$\text{subject to } \sum_{t \in T} n_v(t)f(t) \leq c_v, v \in V \quad (11)$$

$$\sum_{t \in T} n_s(t)f(t) \leq c_s \quad (12)$$

$$f(t) \geq 0, t \in T \quad (13)$$

Inequalities (11) and (12) refer to the capacity constraint. In fact, problem (10) is only a special case of the problem (5), thus can be solved by algorithm **MultiTrees-General** in the same linear programming fashion. However, given the simplified topology, we are interested to find out if this problem can be simply addressed through combinatorial optimization techniques. With this consideration, we design a new algorithm with complexity of $O(|V|)$.

Theorem 2: MultiTrees-Star algorithm returns the optimal result of problem (10).

C. Single Tree Under General Topology Model

A salient feature of the **MultiTrees-General** algorithm is that it reveals the maximum generalized throughput a P2P network can achieve. However, given the exponential selection space in tree set T , the algorithm often returns a high number of trees, which are hardly manageable in practice. For practical purposes, we enforce a limit on the number of trees we can construct. To achieve so, we modify problem (5) into the following integer programming problem.

$$\text{maximize } \sum_{t \in T} f(t)R(t)x(t) \quad (14)$$

$$\text{subject to } \sum_{t \in T} n_e(t)f(t)x(t) \leq c_e, \forall e \in \mathcal{E} \quad (15)$$

$$\sum_{t \in T} x(t) = k \quad (16)$$

$$f(t) \geq 0, x(t) = \{0, 1\}, t \in T$$

²We note that unless otherwise notified, our discussion in this section assumes that the inbound bandwidth of each peer v is unbounded, thus removed from the problem formulation. By the end of this section, we will introduce how our algorithms could be adapted to incorporate the inbound bandwidth constraint.

Problem (14) introduces a 0-1 variable $x(t)$, and k , the upper limit on the number of trees. This constraint is enforced by Eq. (16). This problem is NP-hard since its special case has been proved so [9].

We also design a FPTAS algorithm, the k -**Tree** algorithm, in [9]. The algorithm runs k iterations, in each of which a tree is returned. The following theorem summarizes the property of our algorithm. Let f^* denote the optimal value of problem (14), $f_{achievable}$ the generalized throughput of our algorithm, we define competitive ratio as $f^*/f_{achievable}$.

Theorem 3: Under the non-concatenation model, the k -**Tree** algorithm achieves the competitive ratio bounded by $\log \mathcal{E}$.

D. Single Tree Under Star Topology Model

The number of trees returned by the **MultiTrees-Star** algorithm scales up linearly with $|V|$, the size of the P2P network. Although more scalable than the **MultiTrees-General** algorithm, the number of trees can be still too big as the P2P network grows. It motivates us to study the single tree problem under the star topology model. Similar to problem (14), we can get Problem (17).

$$\text{maximize } \sum_{t \in T} f(t)R(t)x(t) \quad (17)$$

$$\text{subject to } \sum_{t \in T} n_v(t)f(t)x(t) \leq c_v, v \in V \quad (18)$$

$$\sum_{t \in T} n_s(t)f(t)x(t) \leq c_s \quad (19)$$

$$\sum_{t \in T} x(t) = k \quad (20)$$

$$f(t) \geq 0, x(t) = \{0, 1\}, t \in T \quad (21)$$

In particular, we are interested in the case when $k = 1$, i.e., when only one tree is allowed. We design **SingleTree-Star** algorithm and prove its optimality as follows.

Theorem 4: Under the non-concatenation model, the **SingleTree-Star** algorithm returns the optimal solution for problem (17) when $k = 1$, with running time $O(|V|^3)$.

IV. EVALUATIONS

In this section, we present our evaluation study, which mainly carries two purposes. First, we will evaluate the validity of the generalized flow optimization framework at capturing the key characteristics of fault resilient peer selection problem. Second, we will study the performance of the algorithms proposed in this paper, as well as several well-known heuristics, at maximizing the generalized throughput and maintaining fairness. Due to space constraint, we only report a subset of our experimental results. The complete set of results can be found in our technical report[9].

A. Experimental Setup

We use simulation to evaluate the performance of our algorithm. Two experimental topologies are chosen. The first

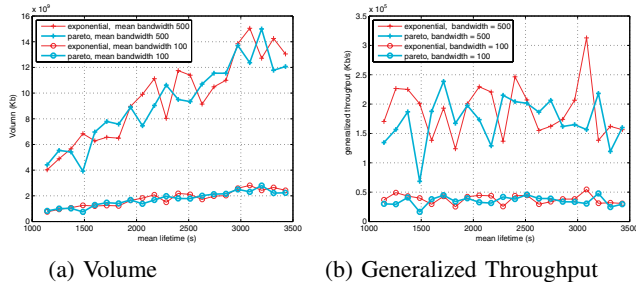


Fig. 2. Performance of **MultiTrees-General** under Non-Concatenation Model

one is a 1000-node router-level network (2000 edges) created by the Boston BRITE topology generator using the Waxman model. Any pair of routers are connected by a pair of links with opposite directions. The bandwidth of physical links between routers, as well as peers' access links, are normally distributed from 100Kbps to 1000Kbps. The second topology follows the star configuration outlined in Fig. 1 (b).

Under both topologies, we create 100 peers with unlimited inbound bandwidth. Under the general topology, they are randomly attached to the routers in the network.

Each simulation run lasts a finite time period. Starting from time 0, each peer is assigned a lifetime based on exponential and Pareto lifetime distributions with mean lifetime varying from 1500 seconds to 3500 seconds. The simulation run expires when the lifetime of the longest-lived peer expires. In our simulation, we assign resilience factor to each peer based on its expected lifetime in each particular run. Our algorithms are executed at the beginning of each run, taking the resilience factors and outbound bandwidths of all peers as the input, and returning single or multiple trees whose combined generalized throughput is maximized.

As time proceeds, peers expire one by one, which gradually tears down the tree(s) constructed at the beginning of the simulation. To capture this effect, we accumulatively calculate the amount of data collected by each peer until its ancestor or itself fails. We term this result as *volume*, which represents the capability of the constructed tree(s) at collecting data for all peers before they demise.

B. Generalized Throughput vs. Volume

In Fig. 2, we run the **MultiTrees-General** algorithm under the general topology, and contrast the generalized throughput returned by the algorithm in (a), calculated volume in (b). We observe that the performance difference under two lifetime distributions are consistently obeyed in both figures when varying the mean peer lifetime.

We then run the **MultiTrees-Star** under the star topology, and contrast the generalized throughput and volume by varying the mean outbound bandwidth. We further introduce two heuristic single-tree algorithms. As shown in Fig. 3, performance ordering of these algorithms under different lifetime distributions are consistent in both figures.

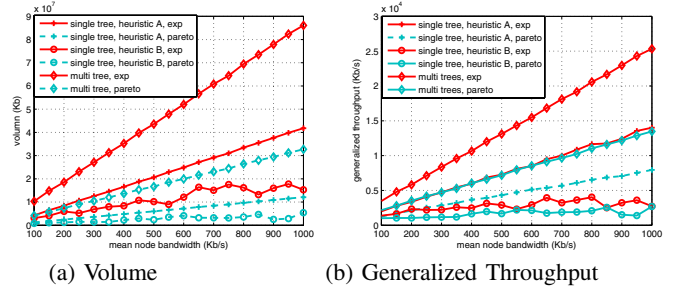


Fig. 3. Performances of **MultiTrees-Star** and Two Heuristics under Concatenation Model (Mean Lifetime = 1500s)

V. CONCLUSIONS

In this paper, we propose an optimization framework based on the generalized flow theory. Utilizing the concept of gain factor in this theory, we introduce the resilience factor of peer to model its chance of survival in a probabilistic measure. Based on this idea, an optimization framework is constructed, whose objective is to maximize the P2P network's generalized throughput, the product of raw throughput and combined resilience factors of all peers. We report our findings in this problem domain along several dimensions including network topology, overlay organization, etc.

Our future work will carry along two directions. On the theoretical front, we will study whether improvement space exists for optimal algorithms presented in this paper, such as **SingleTree-Star**. We will also continue to search approximation algorithms for the NP-hard problems. On the practical front, we will investigate the practicability of our algorithms, such as the distributed solution. We are also interested to search for simple and practical heuristics and have quantitatively evaluated within our optimization framework.

REFERENCES

- [1] Michael Bishop, Sanjay Rao, and Kunwadee Sripanidkulchai, "Considering priority in overlay multicast protocols under heterogeneous environments," in *Proc. of INFOCOM*, April 2006.
- [2] G. Tan and S. Jarvis, "On the reliability of dht-based multicast," in *Proc. of INFOCOM*, 2007.
- [3] D. Leonard, Z. Yao, V. Rai, and D. Loguinov, "On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, 2007.
- [4] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Englewood Cliffs, NJ, 1993.
- [5] K. Wayne, "A polynomial combinatorial algorithm for generalized minimum cost flow," in *ACM Symposium on Theory of Computing*, 1999.
- [6] K. Wayne and L. Fleischer, "Faster approximation algorithms for generalized flow," in *ACM-SIAM symposium on Discrete algorithms*, 1999.
- [7] S. Khuller, B. Raghavachari, and N. Young, "Balancing minimum spanning trees and shortest-path trees," *Algorithmica*, vol. 14, no. 3, 1995.
- [8] R. Cohen and G. Kaempfer, "A unicast-based approach for streaming multicast," in *Proc. of IEEE INFOCOM*, 2001.
- [9] B. Chang, Y. Cui, and Y. Xue, "Maximizing resilient throughput in peer-to-peer network: A generalized flow approach," in <http://vanets.vuse.vanderbilt.edu/infocom2008.pdf>, 2008.
- [10] N. Garg and J. Konemann, "Faster and simpler algorithms for multi-commodity flow and other fractional packing problems," in *Proc. of IEEE FOCS*, 1998.