A Comparison of Network Coding and Tree Packing

Yunnan Wu*, Philip A. Chou†, and Kamal Jain†

*Dept. of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA
†Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399 USA
yunnanwu@princeton.edu, pachou@microsoft.com, kamalj@microsoft.com

1 Introduction

In this paper, we consider the problem of information multicast, namely transmitting common information from a sender s to a set of receivers T, in a communication network. Conventionally, in a communication network such as the Internet, this is done by distributing information over a multicast distribution tree. The nodes of such a tree are required only to replicate and forward, i.e., *route*, information received. Recently, Ahlswede *et al.* [1] demonstrated that it is in general suboptimal to restrict the network nodes to perform only routing. They show that the *multicast capacity*, which is defined as the maximum rate that a sender can communicate common information to a set of receivers, is given by the minimum $C = \min_{t \in T} C_t$ of max-flows $C_t = \text{maxflow}(s, t)$ between the sender and each receiver. Moreover, they showed that while the multicast capacity cannot be achieved in general by routing, it can be achieved by network coding. Network coding refers to a scheme where coding is done at the interior nodes in the network, not only at the sender and receivers. Li, Yeung, and Cai [2] showed that it is sufficient for the encoding functions at the interior nodes to be linear. Koetter and Médard[3] gave an algebraic characterization of linear encoding schemes and proved existence of linear time-invariant codes achieving the multicast capacity. Jaggi, Sanders, et al. [4][5][6] showed for acyclic networks how to find the encoding and decoding coefficients in polynomial time. Chou, Wu, and Jain [7][8] proposed a distributed scheme for practical network coding in real packet networks achieving throughput close to capacity with low delay that is robust to random packet loss and delay as well as robust to any changes to network topology or capacity.

In this paper, we compare network coding solutions and routing solutions for the problem of information multicast, and we investigate the potential advantages of network coding over routing. To maximize the performance of the routing solutions, we investigate the use of multiple multicast distribution trees. Finding the best collection of multicast distribution trees is called *packing Steiner trees* in graph theoretic language. A Steiner tree is a distribution tree that connects the sender with the set of receivers, possibly through Steiner nodes, which are pure relays. However, it has been shown that finding an optimal packing of Steiner trees, providing maximum throughput, is a NP-hard problem; see, for example, [9]. Thus we develop greedy tree packing algorithms. We compare network coding to tree packing on the network graphs of six Internet service providers. In terms of throughput, tree packing performs comparably to practical network coding in all of the six networks. However, network coding offers additional benefits, including fewer network resources consumed, ease of management, and robustness.

2 Preliminaries

A communication network can be represented as a *capacity graph* $G = (V, E, c)$, where V and E are the set of vertices and edges respectively and associated with each directed edge $e \in E$ is a non-negative edge capacity $c(e)$. In essence, a capacity graph models the connectivity of a communication network as well as the supported bit rates for the underlying communication links. We use the notation $G' = (V', E', c') \preceq G = (V, E, c)$ to indicate that (V', E') is a subgraph of (V, E) and $c'(vw) \leq c(vw), \forall vw \in E' \subseteq E$. A capacity sub-graph $G' \preceq G$ may indicate a particular feasible use of the available network resources, in terms of supported bit rates for the communication links. We also introduce the addition and scaling operations on capacity graphs. Given two capacity graphs $G_1 = (V_1, E_1, c_1)$, $G_2 = (V_2, E_2, c_2)$ and two non-negative scaling factors λ_1, λ_2, the linear combination $\lambda_1 G_1 + \lambda_2 G_2$ refers to a capacity graph $(V_1 \cup V_2, E_1 \cup E_2, \lambda_1 c_1 + \lambda_2 c_2)$, where the edge capacities c_1 and c_2 are each extended to $E_1 \cup E_2$ in the obvious way.

3 Packing Distribution Trees

A distribution tree gives one routing assignment, for information packets to go from the root of the directed tree, namely the sender s, to the receivers T. We can symbolically represent a distribution tree as a capacity sub-graph $G_k = (V_k, E_k, c_k) \preceq G = (V, E, c)$, in which the edge set E_k has been reduced to contain only those edges involved in the tree and the vertex set V_k has been reduced accordingly. A distribution tree, $G_k = (V_k, E_k, c_k)$, can deliver common information to the receivers at a rate

$$R(G_k) \equiv \min_{e \in E_k} c_k(e). \tag{1}$$

Multiple distribution trees, say $G_k, k = 1, \ldots, K$, may be used to achieve a high throughput by using different trees to distribute different information streams, as long as their sum $\sum G_k$ fits in the network, i.e., $\sum G_k \preceq G$. The problem of packing distribution trees for the maximum throughput can thus be stated as the following optimization

$$\max_{\{G_k\}: \sum G_k \preceq G} \sum_{k=1}^{K} R(G_k), \tag{2}$$

where G_k's are distribution trees connecting s with T. Note that this maximum throughput achievable by packing trees is certainly less than or equal to the multicast capacity $C = \min_{t \in T} C_t$.

In general, a distribution tree may involve nodes other than the sender s and the receivers in T. These additional nodes serve solely as information relays and are called *Steiner nodes*. Correspondingly, a *Steiner tree* refers to a general tree, possibly containing Steiner nodes. Two extremes of the Steiner tree packing problem are fundamental theorems in combinatorics. One extreme is the maximum flow problem, or "packing paths," for a single sender and a single receiver. This problem has been well solved. The other extreme is when all nodes are receivers. In this case, a Steiner tree becomes a spanning tree since it reaches all other nodes. The following theorem [10] shows the multicast capacity can be achieved by packing spanning trees.

Theorem 1 (Edmonds' Theorem).
Consider a capacity graph $G = (V, E, c)$ *under the* discrete model, *where all edges in* E

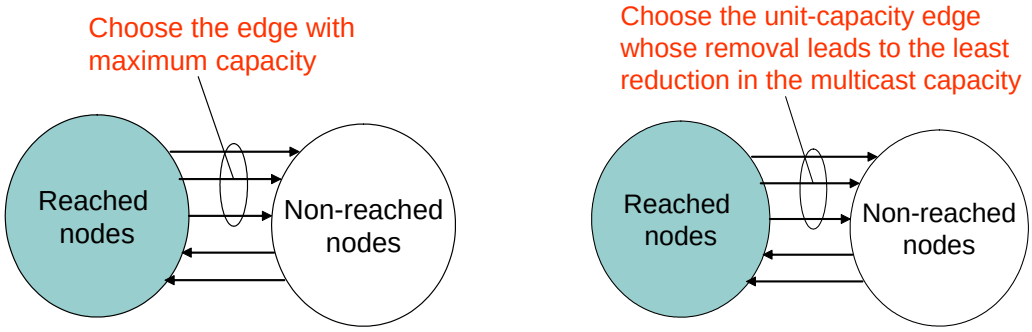

Figure 1: (left): constructing the maximum-rate distribution tree based on Prim's algorithm. (right): constructing a distribution tree based on Lovasz's proof to Edmonds' Theorem.

have unit capacity, and multiple edges may exist for each ordered pair (v, w), $v, w \in V$. *There are* C *pair-wise edge-disjoint spanning trees rooted at a sender* $s \in V$, *where* C *is the minimum of the maximum flow values for each sender-receiver pair* $(s, t), t \in V \backslash \{s\}$.

The presence of Steiner nodes makes the general packing problem (2) NP-hard [9]. Until now, known algorithms for packing Steiner trees have guaranteed a throughput only around $C/|T|$, which is often pessimistic in practice.

In the following, first, we show that packing the *best single Steiner tree*, i.e., the one providing the largest throughput among all distribution trees, is solvable in polynomial time. Based on this observation, a greedy algorithm is proposed, which repeatedly packs the maximum-rate distribution tree.

Let $G_1 = (V_1, E_1, c_1)$ denote a distribution tree. Finding the maximum-rate distribution tree is characterized by the following optimization

$$\max_{G_1 \preceq G} R(G_1) = \max_{G_1 = (V_1, E_1, c_1) \preceq G} \min_{e \in E_1} c_1(e). \tag{3}$$

At first glance, formulation (3) may appear similar with the classical minimum-cost Steiner tree problem, if we treat the negated capacity as the "cost" for each edge, except that the cost of a tree is defined as the sum of the costs for all the constituent edges in the minimum-cost Steiner tree problem, whereas the rate of a tree is defined as the minimum of the edge capacities. The minimum-cost Steiner tree problem is a well-known NP-hard problem in graph theory. Nevertheless, the degenerate version without Steiner nodes, namely the minimum spanning tree (MST) problem, is solvable in polynomial time. Representative algorithms for the minimum spanning tree problem include Kruskal's algorithm[11] and Prim's algorithm[12]. This difference in computational complexity again demonstrates the difficulties caused by Steiner nodes.

Fortunately, the maximum-rate Steiner tree problem turns out to be as easy as the minimum-cost spanning tree problem. Moreover, algorithms may be constructed based on Kruskal's algorithm and Prim's algorithm. A simple explanation of the polynomial solvability is as follows. The maximum-rate Steiner tree can be constructed by first taking the union of the maximum-rate paths from the sender s to each receiver $t \in T$, and then eliminating redundant edges. In fact, Prim's algorithm, or the idea of greedily augmenting a tree edge by edge, can be applied as an efficient way to find all the maximum-rate paths from the sender s to each receiver $t \in T$ together.

Figure 1(left) illustrates the process of constructing the maximum-rate distribution tree according to Prim's algorithm. During the process, each node v is classified as either

"reached" or "non-reached", showing if v has already been reached or not. The ellipse on the left represents the set of reached nodes, which we denote by U, and the ellipse on the right represents the set of non-reached nodes $V - U$. Initially, the set of reached nodes includes only the sender s. In each subsequent step, we select the edge uv with maximum capacity, among those pointing from a reached node to a non-reached node. Then the node v is added to the set of reached nodes, and the edge uv is recorded as an edge on the distribution tree. The process continues until all the receivers have been reached.

After the steps above, a post-processing step is then applied to the constructed distribution tree to prune unnecessary uses of rates on the edges. This pruning is based on two observations. First, there is no need to keep leaf nodes that are not receivers in the tree, since they are not contributing to the delivery to the desired receivers. Hence, we can backtrack from the receivers to the sender in order to identify the contributing edges and nodes and remove those non-contributing edges and nodes. Denote the distribution tree after this pruning step by $G' = (V', E', c') \preceq G$. Second, there is no need to let an edge in the tree consume more than the achievable throughput $R(G')$, which is the minimum rate over all the remaining edges, $\min_{e \in E'} c'(e)$. Thus, we can just set the rate for all edges in E' to be $R(G')$, as

$$G' \leftarrow (V', E', R(G')) \,. \tag{4}$$

Next, we can remove the constructed maximum-rate distribution tree G' from the graph G, i.e.,

$$G \leftarrow G - G'. \tag{5}$$

The above procedure may be repeated until no maximum-rate distribution tree, connecting the sender with the receivers, can be further constructed.

An alternate algorithm to find the maximum-rate distribution tree is based on the pruning step described above. Simply, consider all the edges, one by one, in the increasing order of their capacities. If the deletion of the edge does not disconnect a receiver from the sender then delete the edge else keep it. The edges which survive the deletion process form the maximum-rate distribution tree.

We also develop a third greedy tree-packing algorithm based on Lovasz's constructive proof [13] to Edmonds' Theorem[10]. Assuming all edges have unit-capacity and allowing multiple edges for each ordered node pair, the algorithm packs unit-capacity trees one by one and each tree is constructed by greedily augmenting a tree edge by edge, similar to the greedy tree-packing algorithm based on Prim's algorithm. The distinction lies in the rule of selecting the edge among those pointing from the set of reached nodes U to the set of non-reached nodes $V - U$. Figure 1(right) illustrates the rule, which is to choose the unit-capacity edge whose removal leads to the least reduction in the multicast capacity. For details, see [8].

4 Comparison of Network Coding and Tree Packing

Historically, throughput gain has been the primary motivation for network coding. In their pioneering work [1], Ahlswede *et al.* gave a simple example network, showing that network coding can potentially achieve a higher throughput than routing solutions. Jaggi *et al.*[6] gave a construction demonstrating that network coding could gain as large as $\Omega(\log(|V|))$ in throughput over routing. However, it remained unclear how large would be

the throughput gap in practical networks. Here, we compare the achievable throughput for network coding and tree packing using the network topologies of six commercial Internet Service Providers (ISPs). We use the graphs obtained from the Rocketfuel project at the University of Washington. For each ISP graph, we arbitrarily selected one sender and 20 receivers.

Figure 2 shows the achievable throughput on the six ISP graphs. The figure is partitioned into six sections, each corresponding to an ISP graph. Within each section, from left to right, the shown throughput results are for the maximum-rate distribution tree, repeated packing of maximum-rate distribution trees, greedy tree packing based on Lovasz's proof to Edmonds' Theorem, practical network coding, and the multicast capacity, respectively.

From Figure 2, it can be seen that packing multiple distribution trees offers a significant gain in throughput compared to using only one distribution tree (as in traditional IP multicast, for example). The achievable throughput for both network coding and greedy tree packing is observed to be reasonably close to the multicast capacity. Given the limited six tests, we are not yet ready to conclude that the small throughput gap would often be the case. The throughput gap between network coding and greedy tree packing may depend critically on the structure of the network. It would be an interesting future topic to characterize the structural features that determine the throughput gap.

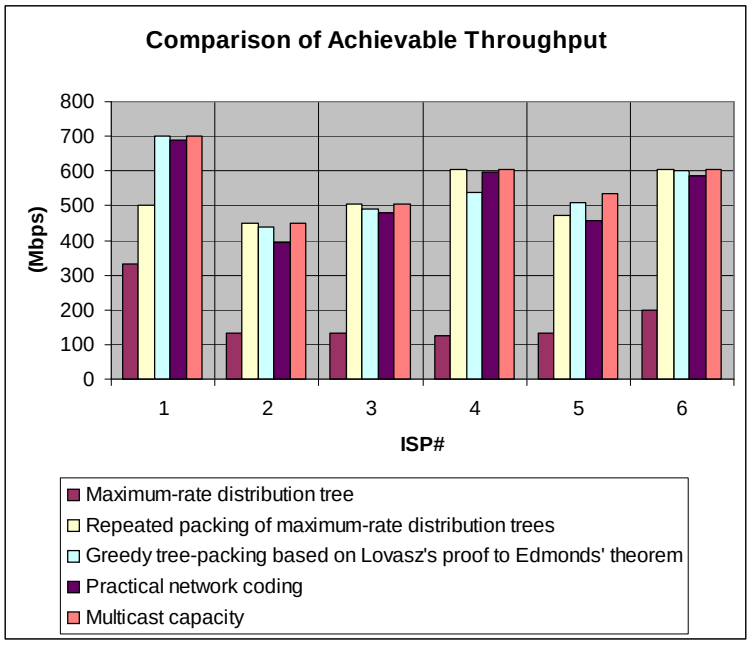

Figure 2: Comparison of achievable throughput on 6 ISP graphs.

Nevertheless, network coding offers additional benefits, including fewer network resources consumed, ease of management, and robustness to both ergodic losses (e.g., packet losses) and non-ergodic failures (e.g., node and link failures). For details, see [8]. Here, we investigate robustness to random link failures. The results are presented in Figure 3(a)-(f) for the six ISP graphs, respectively. Denote the original capacity graph by $G = (V, E, c)$. We run the algorithm that repeatedly packs maximum-rate distribution trees on G and obtains a sum of distribution trees F. We loop through the edges in E and evaluate the throughput after removing one edge at a time. For each edge $vw \in E$, three data points are collected. A first number, which we denote by $C_1(G - vw)$, is the

multicast capacity of $G - vw$, corresponding to the throughput potentially achievable by network coding. A second number, which we denote by $C_2(G - vw)$, is the throughput obtained by the repeated packing of distribution trees on $G - vw$. This result is labelled "adaptive tree packing", since it re-packs the trees after a link failure. A third number, which we denote by $C_3(F, vw)$, is for the "fixed tree packing", which evaluates the effect of removing vw from G on a given set of distribution trees F. If vw is not used in F, the throughput remains the same; otherwise, the offspring nodes on those trees containing edge vw will be affected.

In Figure 3(a)-(f), the horizontal axis is the index of the removed edge, sorted by the throughput with fixed tree packing. For visualization, only the first 200 edges are shown. It can be observed from these 6 figures that adaptive tree packing improves noticeably over fixed tree packing, and network coding offers some further enhancement over adaptive tree packing. Thus, the ability to adapt to changes may result in noticeable throughput advantages.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Information Theory*, IT-46(4):1204-1216, Jul. 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Information Theory*, IT-49(2):371-381, Feb. 2003.

[3] R. Koetter, and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782-795, Oct. 2003.

[4] S. Jaggi, P. A. Chou, and K. Jain, "Low complexity optimal algebraic multicast codes," In *Proc. IEEE Int'l Symp. Information Theory*, Yokohama, Japan, June 2003.

[5] P. Sander, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," In *ACM Symp. Parallel Algorithms and Architectures (SPAA)*, pp. 286-294, San Diego, CA, June 2003.

[6] S. Jaggi, P. Sander, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for network code construction," submitted to *IEEE Trans. Information Theory.*

[7] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *51st Allerton Conf. Communication, Control and Computing*, Oct. 2003.

[8] Y. Wu, P. A. Chou, and K. Jain, "Practical network coding," Technical report, Microsoft Research. In preparation.

[9] K. Jain, M. Mahdian, and M.R. Salavatipour, "Packing Steiner trees," *14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.

[10] J. Edmonds, "Edge-disjoint branchings," in: *Combinatorial Algorithms*, ed. R. Rustin, pp. 91-96, Academic Press, NY, 1973.

[11] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. American Math. Society*, 7:48-50, 1956.

[12] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical J.*, 36:1389-1401, 1957.

[13] L. Lovasz, "On two minimax theorems in graph theory," *J. Combin. Theory* B 21, pp. 96-103, 1976.

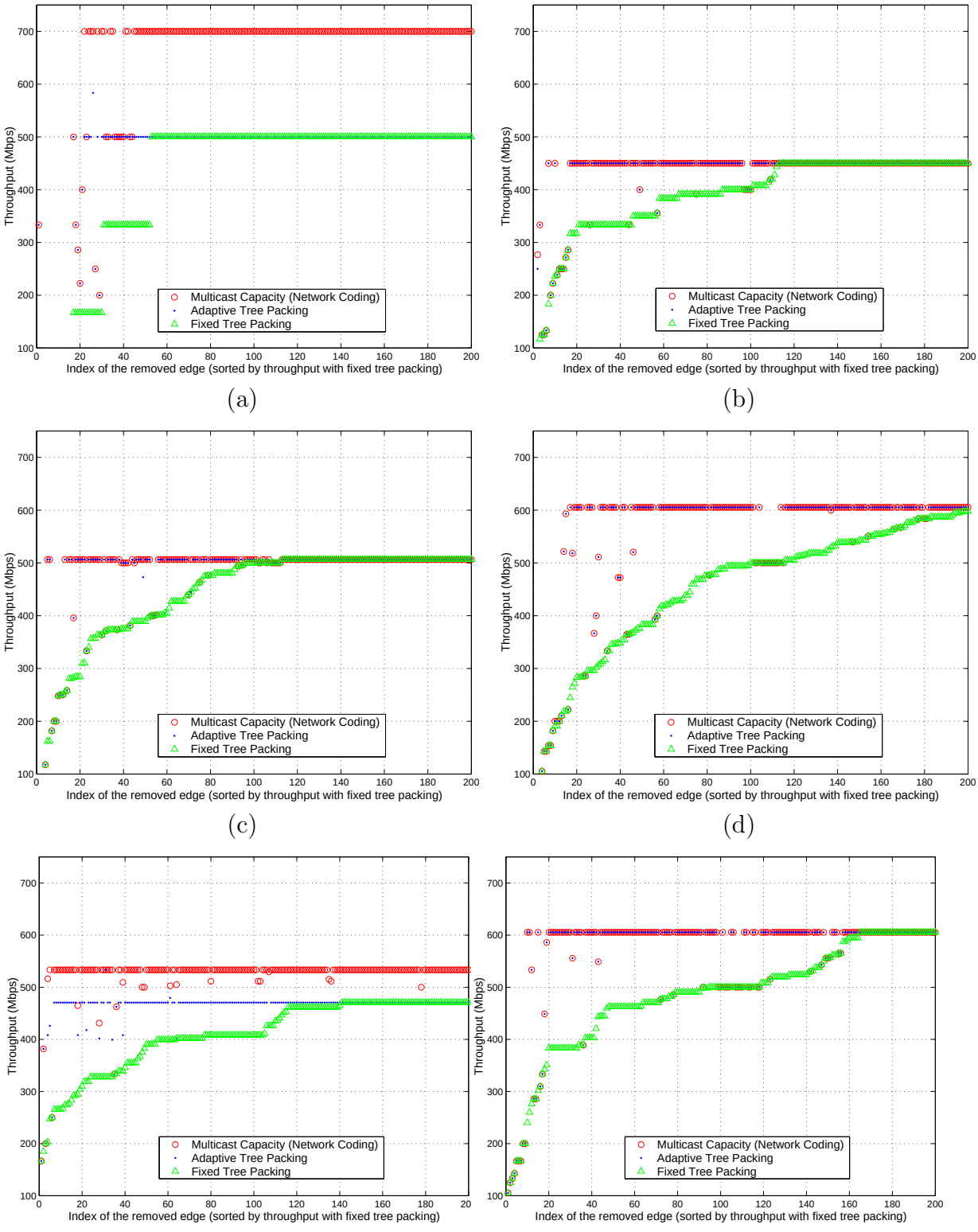

Figure 3: Comparison of throughput after removing one edge at a time. Edges are sorted by the throughput for the fixed tree packing.