Can Network Coding Help in P2P Networks?

Dah Ming Chiu, Raymond W Yeung, Jiaqing Huang*and Bin Fan
Chinese University of Hong Kong

Abstract

In this paper, we compare the maximum achieveable throughput using network coding with routing in P2P networks. Our analysis is based on a simple star network where there is no multicast and network coding can only be applied at the peers. This model captures the essential elements of a P2P network, yet allows simple analysis under both schemes. The conclusion is that there is no coding advantage. We then discuss the applicability of this result to a real P2P content distribution system which may operate at lower throughput due to various other factors. Finally, in addition to yielding insights to the present case of P2P networks, we believe this type of non-multicast network models can lead to other new results for network coding in general.

1 Introduction

The throughput of *network multicast* is limited to the bottleneck of the multicast tree. Recently, it is shown that *network coding* combined with network multicasting can boost multicast throughput to be the minimum of the min-cut from the source to the multicast receivers [1, 2]. This is best illustrated by the *butterfly* network example in Figure 1 from [3].

A special form of network coding - *random linear network coding* (RNC) - is applied to multicast networks [4, 5], and is shown to asymptotically achieve the maximum multicast capacity of a network with probability 1 when the code alphabet is large. The randomization removes the practical difficulty of comingcup with the network codes and placing them at specific network nodes according to the given network topologies.

For a variety of reasons, network multicast has not been widely deployed in the Internet. Instead, scalable content distribution - a major intended application for network multicast - has been realized through peer-to-peer (P2P) networks. One of the most well-known example is BitTorrent [6]. Multicast via P2P netwoks is also known as application layer multicast [7, 8]. One of the important advantages of appllication layer multicast is the flexibility to use multiple spanning trees simultaneously to improve the achievable throughput [9].

More recently, the theoretical work on network coding is considered for implementation in practical P2P content distribution systems [10]. In particular, a well publicized study [11] applied RNC to a BitTorrent-like system for large scale content distribution. As in BitTorrent, a large file is split into many pieces; but instead of helping the server to distribute the pieces intact, the peers apply random linear coding to the pieces that they have before forwarding. The project is code-named Avalanche. In [11], based on a simulation study, the authors claim that Avalanche has a performance (throughput) gain of 20-30% over coding at the server only, and 2-3 times over

[*] Jiaqing Huang is normally affiliated with the Department of Electronics and Information Engineering of Huazhong University of Science and Technology. He contributed to this work while he was a postdoc at CUHK.

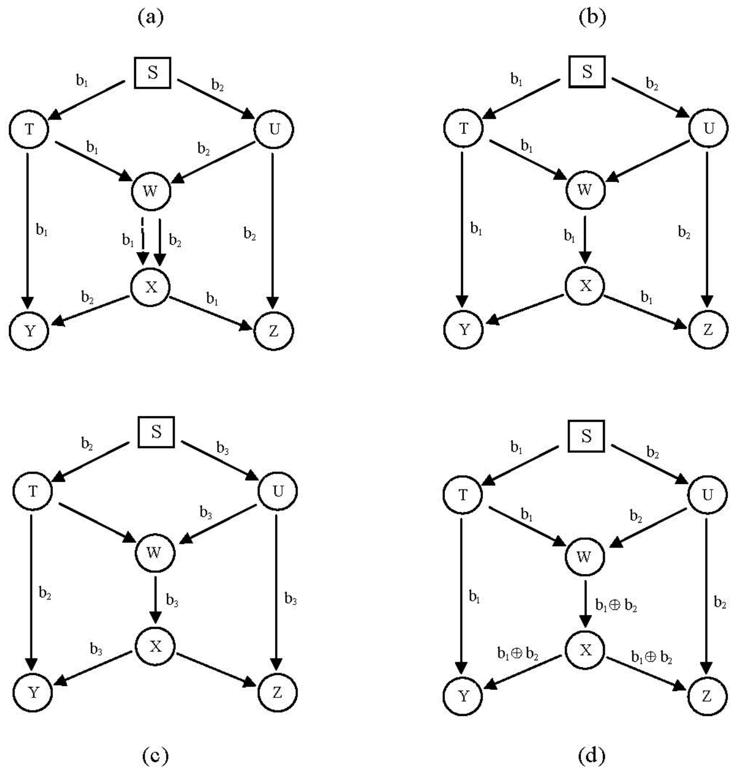

Figure 1: Example for network coding. *(a)* Source S sends bits b_1 and b_2 into the multicast network where each intermedia node replicates the received bit to its output links for destination Y and Z. With two channels between W and X, multicast throughput of 2 (bits per unit time) is achieved. *(b-c)* If one of the channels between W and X is removed, then the highest throughput that can be achieved is to send 3 bits, b_1, b_2 and b_3 into the multicast network in two time slots as shown in (b) and (c). A throughput of 1.5 is achieved. *(d)* In this case, the source S sends two bits into the multicast network with network coding applied before the bottleneck link is traversed. The single channel from W to X is used to transmit $b_1 \oplus b_2$, which is then replicated by X for destination Y and Z. Again, a throughput of 2 is achieved, but this time with a single channel between W and X.

no coding. This stirred up some heated debate on whether the simulation study in [11] presented enough evidence for its performance claims, see [12] for example.

In this short paper, we discuss the potential benefits of networking coding in P2P content distribution by considering a simple *star network* that, we believe, captures the important features of a P2P network. Asymptotic bounds on achievable throughput based on routing is given in [13]. We argue that this network is also appropriate for deriving a theoretical throughput bound for best possible performance under network coding (including RNC) applied at the peers. We show that for this network model and in terms of the maximum throughput bound, there is no benefit from network coding over routing. There are many prior results comparing network multicast capacity using network coding versus using multiple spanning trees, notably [14, 15]. The conclusion is that there is generally an advantage using network coding, and the advantage gained is often referred to as the *coding advantage*. For the single source case, the coding advantage is shown to be small in many practical networks [15]. All these results hitherto on coding advantage considered network coding applied in the network where nodes are capable of supporting network multicast. Our result points out that if coding is applied at the peers and there is no multicast in the network (which is the case for a P2P network), it is provable that there is no coding advantage.

Of course in practical P2P systems such as BitTorrent, the actual obtainable maximum throughput may be far from this achievable bound due to scheduling difficulties, and the fact that the P2P network is continuously changing. Random network coding may help deal with these problems for various network topologies and sizes. Our analysis in this paper does not model such engineering details of practical P2P systems. Therefore, we believe it is important to distinguish the comparison at the theoretical level versus the engineering level. In the concluding section, we briefly discuss the factors that come into the comparison for the practical P2P systems.

2 The Star Network Model for P2P Content Distribution

Our star network model is no different than the *uplink sharing* model from [13], as shown in Fig. 2. A server and n peers are connected to the network, each with an uplink and a downlink. The server's downlink has capacity C_0. (Since the server does not upload, its uplink capacity is irrelevant). Each peer i ($i = 1, \ldots, n$) has a uplink of capacity C_i and a separate downlink of sufficiently large capacity that can be assumed to be infinite.

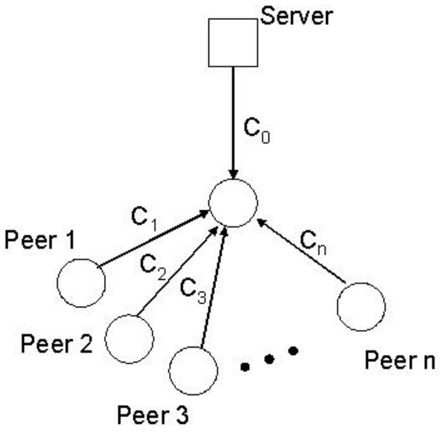

Figure 2: Star model of P2P networks

Let us make the same *fluid assumption* as in [13], that is let the content from the server be infinitely divisible and the file be infinitely large so that the server continuously send content to the peers. The throughput of the system is defined as the amount of content received by *all* peers per unit time. The peers can help the server by forwarding what they received from the server to other peers. However, one important difference between the peers and the intermediate nodes in Figure 1 is that a peer cannot multicast (replicate) content to multiple other peers at the same time. Rather, it must send its content to others one piece at a time over its uplink. So it is possible to think of how information flows from the server to each peer as consisting of different paths each traversing zero or more other peers. Since each piece of the content must be sent to all peers, the composition of these paths must form a spanning tree. In general, the strategy is to use multiple spanning trees to deliver the content to all the peers. Following [13], a two-hop strategy is one where the spanning tree is of depth at most two-hops.

In this network, it is fairly obvious that only the downlink of the server and the uplinks of the peers are possibly constraining the throughput of content delivery to all peers. One of the main results of [13] is restated as follows.

Theorem 1. *Given the star network and the fluid workload, the maximum system throughput is:*

$$R = min\{C_0, \frac{C_0 + \sum_j C_j}{n}\}. \tag{1}$$

And there exists a two-hop strategy that achieves this throughput.

The proof is not given in [13], but should eventually be available from the full version of that paper. We derive the proof independently here, since it is helpful in our later discussion of the case when network coding is applied.

Let each spanning tree k be identified by a unique normalized resource usage ratio given by $(s_0(k), s_1(k), \dots, s_n(k))$ where each $s_i(k)$, an integer, represents the usage of link C_i relative to all other links $(C_0, C_1, \dots, C_n)$, and $\sum_i s_i(k) = n$. The implication of this definition is that the system throughput of using each spanning tree is normalized to unity. For example, the one-hop spanning tree is $(n, 0, 0, \dots, 0)$, and the n-hop spanning tree when the server has its content forwarded by the peers in order is $(1, 1, 1, \dots, 1)$. In each case, the system throughput is 1, but the resource (server downlink and peer uplink) usage pattern is precisely given by the vector.

Let S be the set of different spanning trees used, and λ_k be the rate that the content is sent to the spanning tree k, then the following resource constraints must be satisfied:

$$\sum_{k \in S} \lambda_k s_i(k) \le C_i \;\; \forall i \tag{2}$$

The maximum throughput of the system is then determined by the S and $\lambda_k, k \in S$ that maximizes $\sum_{k \in S} \lambda_k$ yet satisfying the constraints in Equation 2.

In the simple star network, since the network is not a bottleneck, all the uplinks of the peers are equivalent and can be used in serving other peers interchangeably. We can think of all the peers as aggregated together into another server (server 2) of total uplink capacity $C = \sum_{j=1}^n C_j$, as shown in Figure 3. This allows us to aggregate all the spanning trees with the same r_0 into the same class. There are therefore a total of n classes of spanning trees, corresponding to the following resource usage patterns

$$(i, n-i) \;\; i = 1, 2, \dots, n$$

where r_0 is the resource usage at the server (as before) and r_1 is the resource usage at server 2. Let the corresponding traffic intensity of the i^{th} class of spanning tree be denoted $\lambda_{\{i\}}$.

The throughput optimization problem thus reduces down to:

$$\begin{array}{rc} \text{maximize} & \sum_{i=1}^n \lambda_{\{i\}} \\ \text{subj to} & \sum_{i=1}^n i\lambda_{\{i\}} \le C_0 \\ & \sum_{i=1}^n (n-i)\lambda_{\{i\}} \le C \end{array} \tag{3}$$

This is a very simple linear program. The solution may occur at one of the two vertices given by the constraints, depending on whether C_0 or C is more constraining. The maximum throughput is:

$$R = \begin{cases} C_0 & \text{if } C_0 < \frac{C}{n-1}, \\ \frac{C_0 + C}{n} & \text{if } C_0 \ge \frac{C}{n-1}. \end{cases}$$

This proves the first part of the theorem. For the existence of a two-hop strategy, it can be given by construction as follows.

There are two cases:

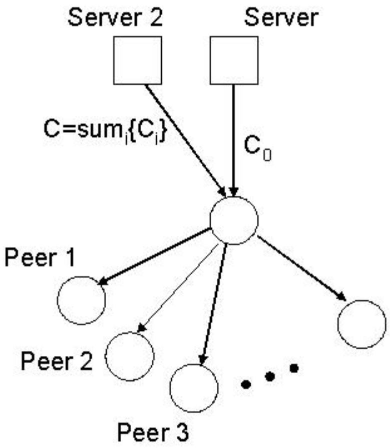

Figure 3: The equivalent star network

- Case 1: $C_0 < \frac{C}{n-1}$. In this case, only those spanning trees with $r_0 = 1$ are used. Namely, the server sends each piece of content to only one peer and it is then forwarded by that peer to the rest of the peers. The two-hop trees S_2 and their corresponding traffic intensity $\{\lambda_k | k \in S_2\}$ that can be used to achieve the maximum throughput are:

  $$\text{The } k^{th} \text{ spanning tree} = (1, e_k), \quad \lambda_k = C_0 \frac{C_i}{C}$$

  Here, e_k is used to denote the vector with the k^{th} element equal to 1 and the rest of the elements equal to zero. It is easy to verify from the problem definition 3 that the total throughput is $\sum_{i=1}^{n} \lambda_k = C_0$. Furthermore, since each peer k needs to send what it receives to $n-1$ other peers, its total upload rate is $(n-1)\lambda_k$ which is less than C_k given the assumption for this case.

- Case 2: $C_0 \geq \frac{C}{n-1}$. In this case, the server uses both the two-hop spanning trees as in case 1 and the one-hop spanning tree. The rate for the two-hop spanning trees are:

  $$\text{The } k^{th} \text{ 2-hop spanning tree} = (1, e_k), \quad \lambda_k = (C_0 - \frac{C}{n-1})\frac{C_i}{C}$$

  and the rate for the unique one-hop spanning tree is

  $$\text{The one-hop spanning tree} = (n, 0, 0, \dots, 0), \quad \lambda_{1-hop} = \frac{C_0 - \frac{C}{n-1}}{n}$$

  Again, it is straightforward to verify that $\lambda_{1-hop} + \sum_{i=1}^{n} \lambda_k = \frac{C_0 + C}{n}$, and that both of the constraints in 3 are satisfied.

Since all the spanning trees that share the same r_0 are equivalent, the two-hop strategy above that achieves maximum throughput is not unique. There are combinatorially many multi-hop spanning trees that satisfy the uplink capacity constraints which can also achieve the maximum throughput. This is a subtle, and important point for our discussion later.

The above theorem, and the ensuing discussion covered the routing solutions, what is the maximum throughput using the routing solutions, and a concrete construction of a specific routing solution that achieves the maximum throughput. The main goal of this paper, however, is to establish whether peers can use network coding to achieve a better throughput bound. The routing bound has two cases. In case 1, it achieves C_0 throughput for all peers. This happens to the *minimum cut* for each peer as well, so we know network coding cannot help in improving this bound. In the other case, when $C_0 \geq \frac{C}{n-1}$, routing achieves a throughput of $\frac{C_0+C}{n}$ which we know is less than C_0 (the minimum cut). The natural question is whether networking coding can help improve this case.

Theorem 2. *Given the star network with no coding or multicast (replication) in the network, then any coding applied at the peers cannot improve the throughput bound given by Theorem 1.*

Proof. Due to the assumptions of the star network, it turns out the proof for this theorem is very simple, without any special knowledge of network coding.

C_0 is the min-cut to all the peers, which is a well-established bound on maximum throughput.

In general, in order to achieve an information throughput of X, a peer must be receiving content at least at the rate of $Y \geq X$ from the server or other peers. This is the case whether network coding is applied or not. Network coding only helps to ensure Y contains non-redundant information so you can deduce X. In the star network we are considering, the network does not do multicasting, but only forwarding; hence it is not generating any distinct information. So the total capacity (of sourcing information) to satisfy all the peers is $C_0 + C$. If this quantity is less than nC_0, then at most we can split this capacity amount the n peers, and get a maximum throughput of $\frac{C_0+C}{n}$. Therefore the maximum throughput must also satisfy this bound network coding is applied at the peers or not. □

The star network is arguably a suitable model of P2P networks. It is particularly interesting that for this model we can enumerate the spanning trees, easily derive the maximum throughput under fluid traffic assumptions, and show that network coding will not achieve a better bound than routing. In general, we believe this class of networks - star networks or more general topologies with shared links and no multicasting - provides a new direction for network coding analysis.

3 Discussion and Conclusions

The analysis and conclusions in the previous section is based on an idealized model of a P2P network that focus on the achievable maximum multicast throughput. In reality, a real P2P content distribution system will achieve no where near this idealized system capacity. We briefly discuss some of the reasons.

- The idealized model depend on distributing content from the server along a number of spanning trees which may share certain links. The analysis therefore assumes the peers will do the equivalent of time-division multiplexing of the flows sharing the common links. This would not happen in the datagram-based Internet, and throughput would be less than the maximum due to head-of-line blocking.

- The above analysis assumes a static network environment. In a real-life P2P system, peers arrive at different times and have various departure behaviours. This means the network topology is constantly changing, and it would be very hard for the network to adapt perfectly to the topology continuously.

- The analysis assumes perfect information (network topology, peers, link capacity etc) to compute a optimal suite of spanning trees to use. In a real P2P network, such information would be hard to come by.

- Even if we assume the server can gather all the needed information to compute a suite of spanning tree to achieve the maximum throughput, it may not be conformant to the incentives of the peers to comply. An optimal suite of spanning trees inevitably require the peers with more uplink capacity to serve more, but the *rich* peers may want to selfishly look for ways to finish quicker and not having to provide as much service called upon them.

Because of these reasons, given a peer i that needs a piece of content x and another peer j that has x and some spare uplink capacity, peer j may not serve i for any of the reasons above, hence we would not achieve the system capacity.

A P2P network equipped with random network coding at the peers would need to struggle with the same set of issues as a spanning tree routing scheme. The difference is the RNC-based system can be more opportunistic in seeking downloads. The upside is that it is more likely to overcome the scheduling difficulties. The downside may be more coding (compute) overhead, and potentially some wasted bandwidth due to redundant transmissions. The result would be quite complicated to model, and is beyond the scope of this paper.

In conclusion, we present a simple model for P2P content distribution networks and show that there is no coding advantage. This theoretical result does not, unfortunately, settle the debate between the proponents and competitors of Avalanche. We present some discussions of the issues that need to be address in modeling the practical P2P systems which, hopefully, shed some light on that problem.

References

[1] R. Ahswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, 2000.

[2] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, 2003.

[3] R. Yeung, S. Li, N. Cai, and Z. Zhang, "Theory of network coding," *submitted to Foundation and Trends in Communications and Information Theory*, 2005.

[4] T. Ho, M. Medard, M. Effros, J. Shi, and R. Koetter, "Toward a random operation of networks," *IEEE Transactions of Information Theory*, 2004.

[5] T. Ho, B. Leong, M. Medard, R. Koetter, Y. Chang, and M. Effros, "On the utility of network coding in dynamic environments," in *Proc. International Workshop on Wireless Ad-hoc Networks (IWWAN)*, 2004.

[6] B. Cohen, "Incentives build robustness in bittorrent," in *Workshop on Economics of P2P Systems*, 2003.

[7] P. Francis, "Yoid: Extending the internet multicast architecture," 2000, http://www.icir.org/yoid/docs/.

[8] Y. Chu, S. Rao, and H. Zhang, "A case for end-system multicast," in *Proc. ACM Sigmetrics*, 2000.

[9] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowston, and A. Singh, "Splitstream: High-bandwidth multicast in cooperative environments," in *Proc. SOSP*, 2003.

[10] P. Chou, Y. Wu, and K. Jain, “Practical network coding,” in *Proc. Proc Annual Allerton Conference on Communication, Control and Computing*, 2003.

[11] C. Gkantsidis and P. Rodriguez, “Network coding for large scale content distribution,” in *Proc. Infocom*, 2005.

[12] Http://www.livejournal.com/users/bramcohen/20140.html.

[13] J. Mundinger, R. Weber, and G. Weiss, “Analysis of peer-to-peer file dissemination amongst users of different upload capacities,” in *Poster presentation at IFIP Performance*, 2005.

[14] K. Jain, M. Mahdian, and M. Salavatipour, “Packing steiner trees,” in *14th ACMSIAM Symposium on Discrete Algorithms*, 2003, to appear.

[15] Y. Wu, A. Chou, and K. Jain, “A comparison of network coding and tree packing,” in *Proc. IEEE International Symposium on Information Theory (ISIT 2004)*, 2004.