COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 7, No. 4, pp. 353-358, 2007

003

AVALANCHE: A NETWORK CODING ANALYSIS

RAYMOND W. YEUNG*

Abstract. In this paper, we study the application of random network coding in peer-to-peer (P2P) networks. The system we analyze is based on a prototype called *Avalanche* proposed in [8] for large scale content distribution on such networks. We present the necessary techniques for analyzing the system and show that random network coding provides the system with both maximum bandwidth efficiency and robustness. We also point out that the model for random network coding in P2P networks is very different from the one that has been studied extensively in the literature.

1. Introduction. For quite some time, BitTorrent [2] and other peer-to-peer applications (e.g., eDonkey [3] and BitComet [4]) have dominated the Internet traffic [5][6]. In 2005, Microsoft announced a prototype called *Avalanche* [7][8] for large scale content distribution on peer-to-peer (P2P) networks that uses network coding [1][10][11], specifically random linear network coding [12], as the core technology. Most recently, Avalanche has been further developed into Microsoft Secure Content Distribution (MSCD) and has been trialed on the Internet for software distribution [9].

What distinguishes Avalanche from BitTorrent and other similar systems is the application of network coding. The performance of Avalanche has been studied by simulation in [8]. A mathematical analysis of a simplified version of such a system can be found in [14].

Potentially, Avalanche can be incorporated in future versions of Windows, so that hundreds of millions of PCs on the Internet will form a gigantic P2P network. On such a network, software updates, patches, entertainment contents, etc, can be delivered efficiently to a large number of users in a scalable manner. In this paper, we analyze the performance of Avalanche from the network coding perspective.

The rest of the paper is organized as follows. In Section 2, we describe how the system works. In Section 3, we present a network coding analysis of the system. Concluding remarks are in Section 4.

2. How the System Works. In this section, we give a brief introduction to Avalanche. We will only focus on those aspects of the system which are relevant to our analysis. For further details on the system, we refer the reader to [8].

Consider the distribution of a file originally residing on a single server to a large number of users on a P2P network. In such a system, the server does not upload the file to each individual user. Rather, it divides the file into k data blocks, $B_1, B_2, \ldots, B_k$,

*Department of Information Engineering, The Chinese University of Hong Kong, N.T., Hong Kong, E-mail: whyeung@ie.cuhk.edu.hk

and uploads possibly coded versions of these blocks to different users at random. These users again help distributing the file by uploading blocks to other users in the network. By means of such repeated operations, a logical network is formed by the users as the process evolves. On this logical network, henceforth referred to as the network, information can be dispersed very rapidly, and the file is eventually delivered to every user in the network. Note that the topology of the network is not known ahead of time.

In the system, new users can join the network as a node at any time as long as the distribution process is active. Upon arrival, a new user will contact a designated node called the *tracker* that provides a subset of the other users already in the system, forming the set of neighboring nodes of the new user. Subsequent information flow in the network is possible only between neighboring nodes.

What distinguishes Avalanche from BitTorrent or any other similar system is the application of network coding, which was introduced in [1]. Compared with routing, network coding allows coding at the nodes within the network, and it offers the benefit of achieving the maximum possible throughput when information is multicast in a point-to-point network. It was further shown in [10][11] that linear network coding suffices to achieve the maximum throughput when a single message (e.g., a file) is to be multicast, which is the scenario under consideration. Random linear network coding proposed in [12] further enables the application of network coding when the network topology is unknown. For a tutorial on network coding, we refer the reader to [15].

For the purpose of coding, the data blocks $B_1, B_2, \ldots, B_k$ are represented as blocks of symbols in a large finite field F referred to as the *base field*, whose size is of the order 2^{16}. At the beginning of the distribution process, a Client A contacts the server and receives a number of *encoded blocks*. For example, the server uploads two encoded blocks E_1 and E_2 to Client A, where for $i = 1, 2$,

$$E_i = c_1^i B_1 + c_2^i B_2 + \ldots + c_k^i B_k, \tag{1}$$

with c_j^i, $1 \le j \le k$ being chosen randomly from the base field F. Note that each E_1 and E_2 is some random linear combination of $B_1, B_2, \ldots, B_k$. In other words, instead of choosing two particular uncoded data blocks to upload, the server forms two encoded blocks by applying random network coding to all the blocks it possesses and uploads them to Client A.

Associated with each encoded block E_i ($i \ge 1$) is a *coefficient vector* $\mathbf{u}^i = [u_j^i]_{j=1}^k$ that gives the necessary information to construct E_i from $B_1, B_2, \cdots, B_k$. We have explained how the blocks E_1 and E_2 are formed, and it is readily seen from (1) that for $i = 1, 2$, $\mathbf{u}^i = [c_j^i]$.

In general, whenever a node needs to upload an encoded block to a neighboring node, the block is formed by taking a random linear combination of all the blocks possessed by that node. Continuing with the above example, when Client A needs to upload an encoded block E_3 to a neighboring Client B, we have

$$E_3 = c_1^3 E_1 + c_2^3 E_2, \tag{2}$$

where c_1^3 and c_2^3 are randomly chosen from F. Substituting (1) into (2), we obtain

$$E_3 = \sum_{j=1}^{k} (c_1^3 c_j^1 + c_2^3 c_j^2) B_j \tag{3}$$

$$= \sum_{j=1}^{k} (c_1^3 u_j^1 + c_2^3 u_j^2) B_j \tag{4}$$

Thus the coefficient vector $\mathbf{u}^3$ is given by $c_1^3 \mathbf{u}^1 + c_2^3 \mathbf{u}^2$.

The exact strategy for downloading encoded blocks from the neighboring nodes so as to avoid receiving redundant information depends on the implementation, and two such strategies were proposed in [8]. The main idea is that downloading from a neighboring node is necessary only if the neighboring node has at least one block not in the linear span of all the blocks possessed by that particular node. Upon receiving enough linearly independent encoded blocks, a node is able to decode the whole file.

Intuitively, the application of network coding can reduce the file download time because a coded block uploaded by a node contains information about every block possessed by that node. Moreover, in case some nodes leave the system before the end of the distribution process, it is more likely that the remaining nodes have the necessary information to recover the whole file if network coding is used. These issues have been discussed in [8]. In the next section, we give a quantitative analysis to substantiate these claimed advantages of using network coding.

3. Model and Analysis. Let V be the set of all nodes in the system. In a real implementation, blocks of data are transmitted between neighboring nodes in an asynchronous manner, and possibly at different speeds. To simplify the analysis, we assume that every transmission from one node to a neighboring node is completed in an integral number of time units. Then we can unfold the network of nodes in discrete time into a graph $G^* = (V^*, E^*)$ with the node set

$$V^* = \{(i,t) : i \in V \text{ and } t \geq 0\}, \tag{5}$$

where node $(i,t) \in V^*$ corresponds to node $i \in V$ at time t. The edge set E^* specified below is determined by the strategy adopted for the server as well as for all the other nodes in V to request uploading of data blocks from the neighboring nodes. Specifically, there are two types of edges in E^*:

1. There is an edge with capacity m from node (i,t) to node (j,t'), where $t<t'$, if m blocks are transmitted from node i to node j, starting at time t and ending at time t'.
2. For each $i\in V$ and $t\geq 0$, there is an edge with infinite capacity from node (i,t) to node $(i,t+1)$.

An edge of the second type models the assumption that the blocks, once possessed by a node, are retained in that node indefinitely over time. Without loss of generality, we may assume that all the blocks possessed by nodes $(i,l), l\leq t$ are transmitted on the edge from node (i,t) to node $(i,t+1)$.

An illustration of the graph G^* up to $t=3$ is given in Figure 1, where the edges with infinite capacities are lightened for clarity. Note that the graph G^* is acyclic because each edge is pointed in the positive time direction and hence a cycle cannot be formed.

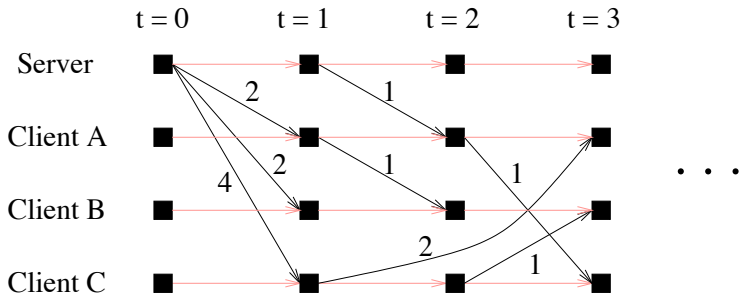

FIG. 1. *A illustration of the graph G^*.*

Denote the server node by s and regard node $(s,0)$ in G^* as the source node generating the whole file consisting of k data blocks and multicasting it to all the other nodes in G^* via random linear network coding. Note that random linear network coding is applied on G^*, not the logical network formed by the user nodes. Thus the model for random network coding studied here is very different from the one that has been studied extensively in the literature.

We also note that in order to simplify our description of the system, we have omitted the necessity of delivering the coefficient vectors associated with the encoded blocks for the purpose of decoding. We refer the reader to [12][13] for this implementation detail.

We are now ready to determine the time it takes for a particular node $i\in V$ to receive the whole file. Denote the value of a max-flow from node $(s,0)$ to a node $v\in G^*$ other than $(s,0)$ by $\mathrm{maxflow}(v)$. When the base field is sufficiently large, with probability close to 1, those nodes (i,t) with

$$\mathrm{maxflow}((i,t))\geq k \tag{6}$$

can receive the whole file [12]. In other words, with high probability, the time it takes a node $i \in V$ to receive the whole file is equal to t^*, the minimum t that satisfies (6). Obviously, this is a lower bound on the time it takes a node $i \in V$ to receive the whole file, and it is achievable with high probability by the system under investigation. In the rare event that node i cannot decode at the time t^*, it can eventually decode upon downloading some additional encoded blocks from the neighboring nodes.

When some nodes leave the system before the end of the distribution process, an important question is whether the remaining nodes have the necessary information to recover the whole file. To be specific, assume that a subset of users $U^c \subset V$ leave the system after time t, and we want to know whether the users in $U = V \backslash U^c$ have sufficient information to recover the whole file. If they do, by further exchanging information among themselves, every user in U can eventually receive the whole file (provided that no more nodes leave the system). Toward this end, again consider the graph G^*. Let

$$U(t) = \{(u, t) : u \in U\}$$

and denote the value of a max-flow from node $(s, 0)$ to the set of nodes $U(t)$ by $\text{maxflow}(U(t))$. If

$$\text{maxflow}(U(t)) \geq k, \tag{7}$$

then the users in U with high probability would have the necessary information to recover the whole file. This is almost the best possible performance one can expect from such a system, because if $\text{maxflow}(U(t)) < k$, it is simply impossible for the users in U to recover the whole file even if they are allowed to exchange information among themselves.

4. Conclusion. In this paper, we have presented a network coding analysis of Avalanche, showing that it achieves the theoretical lower bound on the file download time with respect to the strategy adopted for the centralized server as well as for the nodes in the network to request uploading of data blocks from neighboring nodes. However, the actual performance of the system is affected by how the strategy is chosen, which involves consideration of computational efficiency, incentive mechanisms, etc. We refer the reader to [8] for the details. Further research along this line may include performance optimization of the system based on the analytical framework presented in this paper.

Acknowledgment. The author thanks Alan Lam and Pui Wing Kwok for their useful inputs.

REFERENCES

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

[2] B. Cohen, *Incentive build robustness in BitTorrent*, in: P2P Economics Workshop, Berkeley, CA, 2003.

[3] eDonkey, `http://www.edonkey2000.com`

[4] BitComet, `http://www.bitcomet.com`

[5] NetworkWorld, `http://www.networkworld.com/news/2005/082905-p2p.html`

[6] Linux Reviews, `http://linuxreviews.org/news/2004/11/05_p2p/`

[7] Avalanche, `http://research.microsoft.com/camsys/avalanche/`

[8] C. Gkantsidis and P. Rodriguez, *Network Coding for Large Scale Content Distribution*, INFOCOM 2005, Miami, FL, Mar 13-17, 2005.

[9] Microsoft Secure Content Distribution (MSCD), `http://research.microsoft.com/news/featurestories/publish/MSCD.docx.aspx?0hp=n1`

[10] S.-Y. R. Li, R. W. Yeung and N. Cai, *Linear network coding*, IEEE Trans. Inform. Theory, IT-49(2003), pp. 371-381.

[11] R. Koetter and M. Médard, *An algebraic approach to network coding*, IEEE/ACM Transactions on network coding, 11(2003), pp. 782-795.

[12] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, *The benefits of coding over routing in a randomized setting*, 2003 IEEE International Symposium on Information Theory, Yokohama, Japan, Jun 29-Jul 4, 2003.

[13] P. A. Chou, Y. Wu, and K. Jain, *Practical network coding*, in: The 41st Allerton Conference on Communication, Control, and Computing, Monticello, IL, Oct 2003.

[14] S. Acedanski, S. Deb, M. Médard, and R. Koetter, *How Good is Random Linear Coding Based Distributed Networked Storage?*, NetCod 2005, Riva del Garda, Italy, Apr 7, 2005.

[15] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network coding theory*, Foundations and Trends in Communications and Information Theory, 2:4-5(2005), pp. 241-381, 2005. Downloadable version available at
`http://www.nowpublishers.com/product.aspx?product=CIT&doi=0100000007I` and
`http://www.nowpublishers.com/product.aspx?product=CIT&doi=0100000007II`