Network Coding for Large Scale Content Distribution

Christos Gkantsidis

Abstract

Multicast routing, i.e. the distribution of information from a source node to a large number of destination nodes over a communication network, has attracted a lot of attention for more than a decade (e.g. native IP multicast, CDNs, and, recently, peer-to-peer networks). A fundamental problem in large scale distribution is the optimal scheduling of the data streams. Recently, network coding proposed a novel solution to the scheduling problem by encouraging the network nodes to mix the transmitted data. In this talk, we will review network coding, study its advantages over the traditional store-and-forward paradigm, and describe the design of Avalanche, which is a peer-to-peer system that uses network coding.

Short Description

Content distribution using peer-to-peer networks is one of the most important applications in the Internet today. In fact, recent studies show that almost 60% of the Internet traffic is generated by peer-to-peer applications, such as eMule, BitTorrent, KaZaA, and others. Typically, large files, such as movies and software distributions, need to be distributed from one or very few servers to a large number of users. The objective of the system is to distribute the content as fast and as reliable as possible.

The problem of efficient content distribution can be modelled as a graph-theoretic flow problem. The nodes of the network can be modelled as vertices in a graph, the connections between nodes as edges, and the maximum rate that a node can send information to an adjacent node can be modelled as a capacity constraint on the corresponding edge. It is straight-forward to find the maximum rate from the source to each of the receivers *independently*; the minimum of those maximums is an upper bound of the throughput of the system. Is it possible to find a routing from the server to each of the receivers *simultaneously* that achieves optimal throughput?

The theory of network coding shows that achieving the optimal throughput is possible, and, moreover, computationally efficient (i.e. there are polynomial algorithms for computing the optimal routing). The basic idea is to send *encoded* information along the edges of graph; the nodes of the network process the information they receive and produce encoded packets that they forward to their neighbours. In comparison, the traditional store-and-forward paradigm, in which the nodes send bits of the original file without further processing, cannot achieve the optimal throughput; moreover, computing the maximum possible rate using store-and-forward is a known difficult problem (NP hard).

In this talk, we will discuss the basic ideas behind network coding and examine its advantages over the store-and-forward paradigm. We will also describe how at Microsoft Research at Cambridge we have used the basic ideas of network coding to design and implement a peer-to-peer system that outperforms current peer-to-peer systems both in terms of achievable performance, and also in terms of increased reliability.

Reading List

Here is the list of papers that we will cover in the discussion. (For each paper, I am giving a very short overview of the ideas presented in the paper that I am planning to cover in the lecture; each of the papers covers other topics as well.)

- R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, pp. 1204-1216, 2000. (http://personal.ie.cuhk.edu.hk/~pwkwok4/Yeung/1.pdf)

  This is the first paper on network coding. The paper introduces the concept of network information flow, and, in doing so, relates the problem of multicasting (sending from one node to a subset of the nodes in a graph) to flow problems. It shows that the classical approach to optimal multicasting (using many multicast trees) is not optimal. This paper also shows that with coding we can achieve the optimal rate allowed by the topology.

- S.-Y. R. Li, R. W. Yeung, and N. Cai. "Linear network coding". *IEEE Transactions on Information Theory* , Februray, 2003 (http://personal.ie.cuhk.edu.hk/~pwkwok4/Yeung/2.pdf)

  The paper shows that linear coding is sufficient to achieve the optimal multicast rate. Observe that the previous work of Ahlswede et al. assumed arbitrary coding functions.

- P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," A*llerton Conference on Communication, Control, and Computing*, Monticello, IL, October 2003. (http://research.microsoft.com/users/pachou/pubs/ChouWJ03.pdf)

  This paper shows how to construct a decentralized scheme that uses network coding by pre-pending the encoding coefficients to the encoded packets.

- Christos Gkantsidis, Pablo Rodriguez, "Network Coding for Large Scale Content Distribution", *IEEE/INFOCOM 2005*, Miami. March 2005. (http://www.research.microsoft.com/~pablo/papers/nc_contentdist.pdf)

  In this paper we show how to use and extend the ideas of network coding to design a peer-to-peer system for content distribution. Network coding enables us to improve both the speed of content distribution and, also, the reliability of the system (in the event of node failures).

Optional/extra material:

- P. Chou's tutorial on network coding: http://research.microsoft.com/users/pachou/pubs/ChouTutorial04.ppt
- Network coding home page (with a lot of reading material): http://tesla.csl.uiuc.edu/~koetter/NWC/
- R. Yeung, S.R. Li, N. Cai, Z. Zhang, "*Theory of Network Coding*" (manuscript, pre-print). (`http://iest2.ie.cuhk.edu.hk/~whyeung/post/netcode/main.pdf`)
- For an introduction to flow problems please take a look at the following URL and the references therein: http://en.wikipedia.org/wiki/Maximum_flow_problem.

Exercises/problems

Problem 1: Network coding in node-capacitated networks

Typically network coding literature studies edge-capacitated graphs. In this exercise we shall study how network coding applies in node-capacitated graphs. (Hint: There is a standard transformation from node capacitated graph to edge capacitated graphs. This transformation does NOT work with network coding. Why?)

The study of node-capacitated graphs is particularly important for understanding how to use network coding to construct overlay networks. Overlay networks are network constructed between end-hosts (computers): each end-host is a node and each edge is a connection between a pair of computers. Many applications, including peer-to-peer networks like KaZaA, eDonkey, and BitTorrent, use overlay networks. Observe that the connections in overlay networks are virtual and not "real" physical links. In overlay networks the capacities are at the nodes (the connection of the computer to the Internet) rather than in the paths between computers.

Given a node capacitated graph G=(V,E), such that the input flow at any node $v \in V$ is at most c_in(v) and the output flow from any node $v \in V$ is at most c_out(v), a special source node $s \in V$ and a set of destinations $t \subseteq V$, formulate the problem of finding the maximum rate from source s to all destinations in t using network coding as a linear programming problem.

Problem 2: Field Size

A) Consider the network of Figure 1. The source node has id 0, and the nodes 5-10 are the receiving nodes. Assume that each edge has capacity 1 bit. What is the maximum rate from the source to each of the receivers? What is the maximum rate that network coding can achieve? Is it possible to achieve the maximum rate using XOR operations only? If yes, show how to perform the encoding. If not, explain why not and give an encoding that achieves the maximum rate. (In either case you need to describe how to construct the information transmitted at each outgoing edge as a function of the information arriving at the incoming edges.)

B) **The $\binom{n}{r}$ combination network:** Assume a network similar to that of Figure 1, but now there are n nodes in the intermediate layer and $\binom{n}{r}$ receivers (bottom layer). Each receiver receives information from r intermediate nodes. There are no two receivers that connect exactly to the same set of intermediate nodes. All edges have capacity 1. What is maximum flow from the server to each receiver (ignoring all the other receivers)? What is the maximum rate that network coding can achieve? How big should the field size be to achieve the optimal rate?

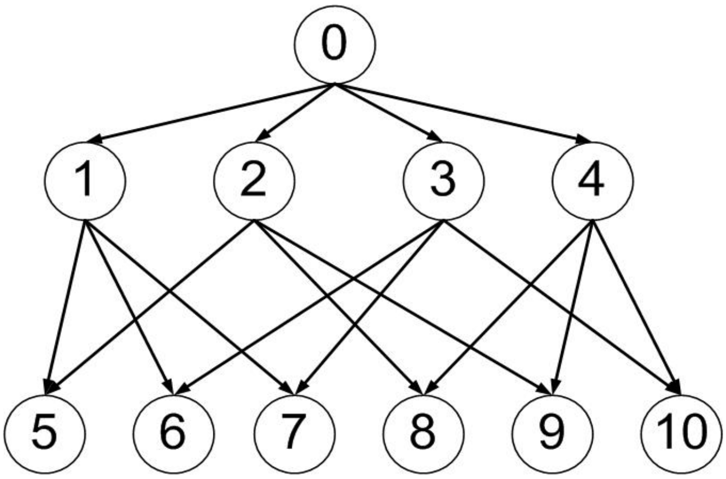

Figure 1

Problem 3: Constructing Linear Codes (programming)

In this exercise you will implement an algorithm for constructing a linear network code for acyclic graphs. In other words, your program will assign linear codes to each of the edges of the graph such that the flow of information from a source node to each of the receivers is maximized.
Your program will get as input a) the description of an acyclic graph, b) a list of receiving nodes. We will assume that the identity of the source node is 0. The syntax of the input file and an example are given in Figure 2.
You may assume that the number of nodes and edges are relatively small (say, 100 nodes and 1000 links at most).
In this project you will need to perform arithmetic on finite fields. There are many available packets that can use:

- http://www.swox.com/gmp/,
- http://www.cs.utk.edu/~plank/plank/gflib/,
- Etc.

If you do not feel comfortable using finite field arithmetic, you may also use real numbers.
You may assume that the capacities of the edges are integers.

<number of nodes> <number of edges> <number of receivers>
<receiver 1> <receiver 2> … <receiver n>
<edge 1 starting node> <edge 1 ending node> <capacity of edge 1>
<edge 2 starting node> <edge 2 ending node> <capacity of edge 2>
…

Example:
7 9 2
5 6
0 1 1
0 2 1
1 3 1
2 3 1
1 5 1
2 6 1
3 4 1
4 5 1
4 6 1

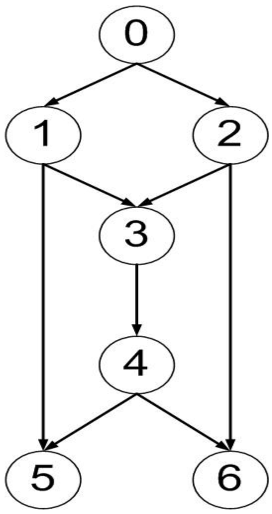

Figure 2

You will need to describe in words or using pseudo-code the algorithm you have used for assigning network code to each of the edges. Make sure that you mention all the assumptions you have made and also a brief justification of the correctness of your algorithm. What is the complexity of your algorithm?
You will also submit a printed listing of your code and the output of your program for the example of Figure 2.

[You can use whichever language/programming system you feel more comfortable with.]

Further questions/problems

1. If every node of the graph is also a receiver, then does the use of network coding increase the maximum achievable broadcasting rate? (Hint: think of the problem of packing Steiner trees.) Answer: No. Is there any other advantage of using network coding? (Hint: think of the complexity of packing Steiner trees.)
2. Assume that you have a set of nodes with capacities at the nodes. Assume also that you can form the topology (graph) as you like, as long as the maximum degree of a node is smaller than a (small) constant d. How can you design a topology that maximizes the rate achieved with network coding? Are there distributed algorithms/heuristics that achieve a high rate? (Distributed in the sense that each node can use constant memory.)
3. In network coding each intermediate node generates “new” blocks (by performing the coding operations). This posses extra difficulties in verifying that the encoded

blocks are correct. Indeed, the source node cannot sign the encoded blocks that are generated by a random intermediate host. As a result, malicious nodes can introduce corrupted blocks and destroy the communication easily. How can an intermediate node verify that a received encoded block is correct? [Hint: Using homomorphic hash functions maybe a solution, however, these functions are slow to compute and do not work in Galois Fields.]

Bio

Christos is a researcher in the Systems and Networking Research Group at Microsoft Research in Cambridge, UK. Prior to joining Microsoft, Christos had been a Ph.D. student at the College of Computing at Georgia Institute of Technology, GA, USA. His current research interests include the areas of content distribution networks, peer-to-peer technologies, analysis and modelling of complex communication networks.
Christos is a member of the IEEE and the ACM.