ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

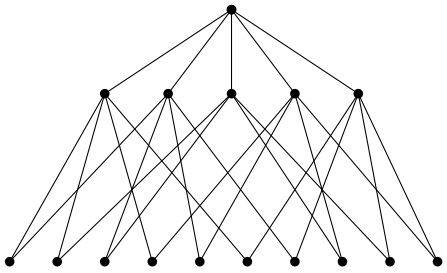
Project **Graph Theory Applications**

Date: April 18, 2013 Spring 2013

Here are the rules for the project.

1. You work in groups of up to 3 people. Please announce your group by April 25th.
2. The due date of the project is Thursday May 30th.
3. Hand in a written report, one per group suffices. We will not make a distinction between the members of your group. Every group member gets the same grade.
4. The report does not need to be long. But it should explain what it is that you did and give answers to the posed questions.
5. As for your graded homework, if you used any resources (books, Wikipedia, etc.) list these sources at the beginning of your report. This does not influence your grade. But if we discover outside influences which are not listed, you will get zero points on your project.

Here is the description of the project. Consider the following graph.



The node on top is the *source* node and the 10 nodes at the very bottom are *receiver* nodes. The graph represents a network. The edges are directed and they are all directed from top to bottom. The basic task is to send information from the source to the receivers.

Each edge has a "capacity" of one bit per time unit. I.e., time is clocked, and in each time unit you can push exactly one bit through each edge. We will explore two different ways of how we can send information from the source to the receivers. The information we want to send is the same to each receiver. So this is a so-called *multicast* problem. In order to solve this problem, you might want to read ahead and look at the chapter which describes the max-flow min-cut problem and the algorithms to solve it.

1. Assume first that the source wants to send information to only a single receiver at a time. We want to use *routing*. This means that at the intermediate nodes, information can be

routed (switched) from any incoming edge to any outgoing edge. I.e., every node acts like a switch connecting the various incoming edges to the various outgoing edges. In addition to switching, nodes can also duplicate information. I.e., if information comes in along e.g. one edge, it can be forwarded along e.g. two outgoing edges. For each of the receiver nodes determine how much information can be sent per unit time. We do not just want the numbers but we are looking for an explanation of how you arrived at this conclusion.

2. Assume now that we want to send common information from the source to all the 10 receivers. This means that we want to send the same information to each of the receivers. Again we want to use routing. What is the maximum amount of common information per unit time which we can send and successfully receive at all receivers? Note that we are allowed to use *time-varying* strategies. This means that for some time period we can use a particular routing scheme and then after some time we can switch to another scheme. What we are interested is the average rate at which all the receivers receive common information per unit time.

   Again, we are not just interested in a number but in your explanation of how you came up with it and why this is the best you can do.

3. Finally, we would like to explore a more sophisticated strategy, called *network coding*. This strategy requires that the nodes perform some algebraic operations, i.e., more work is needed than just routing. But in general, more common information can be multicast in this way.

Before you start, make sure to review some basic facts about linear algebra and modern algebra. In particular, this approach requires that you know what a finite field is, what finite fields exist, and how you can perform linear algebra using finite fields.

Here is how network coding works. Instead of sending single bits, imagine that we send vectors whose components are finite field elements. E.g., assume that we use the field F_{2^m} for some natural number m. Then each components has m bits, and if the vector is of length n, the whole vector can be described by mn bits. Such a vector can hence be transmitted by using mn consecutive time slots.

At the source we generate k "source vectors" uniformly at random. These vectors contain in their components the information we want to send.

Now at the source node as well as every intermediate node (but not the receiver nodes) we proceed as follows. We take the set of all incoming vectors. At the source node these are the k original information vectors. At all intermediate nodes these are the vectors along the incoming edges. For every outgoing edge we take a particular linear combination of the incoming vectors and send this linear combination, which is again a vector of length n. All the algebra is done over the fixed finite field which we have chosen.

So at a particular receiver, let us assume it has in-degree l, we get l incoming vectors. Each of these vectors is a particular combination of the original k information vectors. We assume that all the coefficients which are used at all nodes are known at the receiver. So the task of the receiver is to take the l incoming vectors and the knowledge of the coefficients and to reconstruct from it the k original vectors, i.e., to reconstruct the transmitted information.

The largest k for which we can construct such a scheme so that we can successfully reconstruct all the information vectors at all receivers is our maximum achievable rate. I.e., this is the maximum number of bits per unit time we can achieve with this scheme.

3. Continuing our scheme, determine for our particular graph the maximum rate at which you can transmit common information. Give a particular scheme, i.e., describe your choice of finite field, your choice of n, and your choice of coefficients at every node. Prove that your scheme works and show the workings by means of a particular example. In practice, we are interested in using small field sizes. What is the smallest field size you can get away with? Can you give a general characterization of the maximum achievable rate in terms of graph quantities?