ECE 158A - Multicast and coding for erasures

Multi-cast traffic

For multi-cast traffic the shortest tree to reach all destinations is **not** the tree of shortest paths and routing algorithms must be modified accordingly. Once this tree is established there is an additional difficulty due to errors.

Retransmissions to make multicast reliable are not practical. If a packet gets lost near the root of the multicast tree, then too many users will need to notify the sender for retransmission. A better strategy is to deal with this possibility in advance, through Forward Error Correction (FEC). The idea is to add redundancy to the transmitted packet to deal with possible errors. In a sense, we trade bandwidth for reliability.

Network coding can also be used to improve the overall throughput of the multi-cast tree.

How to implement multicast efficiently on the Internet is still a topic of discussion and even its implementation at the application level or at the networking level is debated. There is a strong interest in this due to streaming and content distribution applications.

0.1 Example

We want to multicast N packets $p_1, \ldots, p_N$ across a link, but unfortunately packets can get lost. We decide to implement a FEC strategy, which consists in generating and transmitting $M > N$ packets $c_1, \ldots, c_M$. Each packet c_i is obtained as a combination (modulo-2 sum) of a subset of $p_1, \ldots, p_N$. For example, fixed $a_{i,1}, a_{i,2}, \ldots, a_{i,N} \in \{0, 1\}$

$$c_i = \begin{bmatrix} a_{i,1} & a_{i,2} & \ldots & a_{i,N} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \ldots \\ p_N \end{bmatrix}$$

means that c_i is obtained as the (bit-by-bit) modulo-2 sum of the packets p_j corresponding to all coefficients $a_{i,j}$ that are equal to 1.

Out of all M transmitted packets, $M' \leq M$ are received. Let these packets be $c_{i_1}, c_{i_2}, \ldots, c_{i_{M'}}$, where $i_1, \ldots, i_{M'}$ are distinct indices in $\{1, \ldots, M\}$. The received packets $c_{i_1}, \ldots, c_{i_{M'}}$ and the original packets $p_1, \ldots, p_N$ are related as

$$\begin{bmatrix} c_{i_1} \\ c_{i_2} \\ \cdots \\ c_{i_{M'}} \end{bmatrix} = \begin{bmatrix} a_{i_1,1} & a_{i_1,2} & \cdots & a_{i_2,N} \\ a_{i_2,1} & a_{i_2,2} & \cdots & a_{i_2,N} \\ \vdots & \vdots & & \vdots \\ a_{i_{M'},1} & a_{i_{M'},2} & \cdots & a_{i_{M'},N} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_N \end{bmatrix}$$
$$= A \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_N \end{bmatrix}.$$

Note that A is a $M' \times N$ matrix. We have that $p_1, \ldots, p_N$ can be recovered from $c_{i_1}, \ldots, c_{i_{M'}}$ if and only if A has rank N. If A has rank N, we can select N linearly independent rows of it and obtain a $N \times N$ matrix A' of rank N (note that A' is therefore invertible).

We now ask the following questions:

1. What is the minimum M' below which it is not possible to recover $p_1, \ldots, p_N$? We need to receive $M' \geq N$ packets (so the minimum is N). If $M' < N$ we cannot build a $N \times N$ matrix A' which is invertible and allows to recover the original packets. If $M' \geq N$, then the packets can be recovered if and only if there is a $N \times N$ submatrix A' of A (obtained choosing N linearly independent rows of A) that is invertible.

2. How can we use A' to recover $p_1, \ldots, p_N$ from the received packets $c_{j_1}, \ldots, c_{j_N}$ corresponding to the chosen rows of A'?

   ANSWER: Let $\mathbf{c}$ be the vector of the N received packets corresponding to the selected row in A' (which is invertible and has inverse $(A')^{-1}$). Let $\mathbf{p}$ be the vector of the original packets. Then $\mathbf{p}$ can be recovered as

   $$\mathbf{p} = (A')^{-1}\mathbf{c}.$$

3. Assume now that we want to transmit packets p_1, p_2, p_3 and that we choose to adopt a FEC strategy sending 6 packets $c_1, \ldots, c_6$ obtained as follows

   $$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}.$$

If we receive c_1, c_2, c_4, can we recover p_1, p_2, p_3? If so, explain how, if not, explain why.

ANSWER: The matrix A corresponding to the received packets c_1, c_2, c_4 is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

The matrix has rank 2, and thus it is not invertible. Therefore, p_1, p_2, p_3 cannot be recovered.

4. If we receive c_1, c_2, c_4, c_6, can we recover p_1, p_2, p_3? If so, explain how, if not, explain why.

   ANSWER: If we receive c_1, c_2, c_4, c_6, can we recover p_1, p_2, p_3? If so, explain how, if not, explain why.

   The matrix A corresponding to the received packets c_1, c_2, c_4, c_6 is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

   The matrix has rank 3. We can choose the rows of A the received packets c_1, c_2, c_6, obtaining

$$A' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

   which is, of course, invertible, and can be user to recover p_1, p_2, p_3.

0.2 One step further: network coding

A similar idea has been proposed to improve network throghput in multi-cast transmissions. The butterfly network in Section 5.5.4 of Walrand's book illustrate the idea. This can be extended to random linear combinations at intermediate nodes. As long as the receivers get enough independent linear combinations of the data, they will be able to invert the system and decode correctly. Further reading is provided in the handout on network coding on the web page.