# Lecture 10

In the previous lecture, we talked about the preprocessing steps for the Linear Information Flow (LIF) algorithm and the local and global coding vectors. In this lecture, we continue with LIF algorithm.

As a matter of convention, we will number the source edges $\{-1, -2, ... - h\}$. As we talked in last lecture, LIF algorithm tries to find a valid global coding vector for each edge in $G'$, in this process, LIF algorithm maintains a frontier edge set and a frontier edge matrix for each terminal. Global coding vectors are assigned to edges one by one in the topological order. For the purposes of this algorithm we shall assume that the edges have been topologically sorted i.e. there exists a path from edge $e_i$ to $e_j$ only if $i < j$. We shall use $e_i < e_j$ to indicate that $e_i$ precedes $e_j$ in the numbering. Let $t$ denote the counter. This indicates the number of the current edge being assigned a coding vector. Recall that the set of paths from the source to terminal $T_i$ was denoted $\mathcal{P}_i$ and each of the $h$ paths was denoted $P_{ij}, j = 1, \ldots, h$.

**Definition 1** *Frontier edge set $F_i^t$. The frontier edge set for terminal $T_i$ at time $t$, $F_i^t$ is a set of $h$ edges, $e_1^{(i)}, ...e_h^{(i)}$ such that $e_j^{(i)} \in P_{ij}$ and $e_j^{(i)} < e_t$. It is initialized at $t = 0$ to be $\{e_{-1}$, $e_{-2}$,.....$e_{-h}\}$ for all $T_i$ .*

**Definition 2** *Frontier edge matrix $M_i^t$. The frontier edge matrix for $T_i$ at time $t$ is an $h \times h$ matrix denoted by $M_i^t$ whose rows are the global coding vectors of the edges in $F_i^{(t)}$. The frontier edge matrices are initialized to be $h \times h$ identity matrix for all the terminals.*

The LIF algorithm maintains the frontier edge sets and frontiers edge matrices as explained below.

At step $t$, the algorithm considers edge $e_t \in E'$.

- It identifies which paths $e_t$ participates in. Note that because of the preprocessing step $e_t$ has to belong to certain paths from the source to some of the terminals. It also identifies the coding vectors of the edges incoming into $e_t$. Note that because the edges are visited in a topological order, coding vectors have already been assigned to the incoming edges into $e_t$.
- The algorithm now identifies a *good* global coding vector belonging to the span of the coding vectors of the incoming edges into $e_t$ and updates the frontier edge sets and frontier edge matrices for all the terminals such that all the new frontier edge matrices remain full rank. We will explain the update procedure for the frontier edge sets below and also show that as long as the field size is large enough we can always find such *good* global coding vectors.

We first explain the frontier edge set update procedure. First realize that because of the preprocessing step, the incoming edges into $e_t$ will be members of the frontier edge sets $F_i^t$ for some or all the terminals $T_i \in T$. We call these predecessor frontier edges of $e_t$ Once the coding vector for $e_t$ has been determined, we remove these predecessor frontier edges from the corresponding $F_i^t$'s and replace them with $e_t$.

Suppose that the coding vector assigned to $e_t$ is denoted $\eta(e_t)$. We need to ensure that the updated frontier edge matrices $M_i^t, T_i \in T$ remain full-rank. That we can always find such a suitable coding vector is shown in Lemma 1 below.

# 1 An Example of the LIF Algorithm

We now show an example of the LIF algorithm. Consider the network shown in Figure 1. Suppose that Figure 1 is the $G'$ obtained from the preprocessing steps for LIF algorithm. There is one source $S$ and two terminals $T_1$ and $T_2$ in the network. Source $S$ produces two source messages and both terminals want to receive the two source messages. From the figure, we can see that the path sets from $S$ to both terminals are as follows.

$$\mathcal{P}_1 = \{\{e_{-1}, e_0, e_2\}, \{e_{-2}, e_1, e_4, e_6, e_7\}\} \text{ } and \text{ } \mathcal{P}_2 = \{\{e_{-1}, e_0, e_3, e_6, e_8\}, \{e_{-2}, e_1, e_5\}\}$$

Initially at time 0, both frontier edge sets $F_i^0$ for $i = 1, 2$ were initialized to be $\{e_{-1}, e_{-2}\}$. Both frontier edge matrices $M_i^0$ for $i = 1, 2$ were initialized to be the $2 \times 2$ identity matrix. The global coding vectors for source edges $\{e_{-1}, e_{-2}\}$ were initialized to be $[1, 0]$ and $[0, 1]$ respectively.

At time 0, we consider assigning a global coding vector for $e_0$ in the graph which should be from the span of the coding vectors of the incoming edges and at the same time make the matrix $M_i$ invertible. Let us assign a global vector $[1, 0]$ for $e_0$. Now we update all $F_i$ for $i = 1, 2$ by substituting the predecessor $e_{-1}$ for $e_0$. At the same time, by examining the path set $P_1$ and $P_2$, we see that $e_0$ exists in both edge sets $P_1$ and $P_2$, so we need to update both $M_1$ and $M_2$. The coding vector $\eta(e_t)$ enters the $M_i^{(1)}$ for both $i = 1, 2$. Note that here $\eta(e_t) = [1\ 0]$ which belongs to the span of the removed row vectors. The condition that $M_1^{(1)}$, $M_2^{(1)}$, remain full rank holds here since they continue to be the identity matrices. The algorithms continues as explained in the previous section.

In Figure 1 the global vectors for all edges in our example are labeled besides corresponding edges. The update process for the frontier edge sets and the frontier edge matrices of the two terminals are shown in Figure 2.

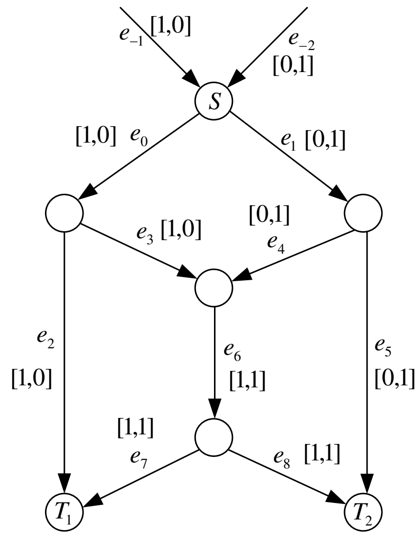


Figure 1: Example network in LIF algorithm

| steps | $F_1$ | $F_2$ | $M_1$ | $M_2$ |
|---|---|---|---|---|
| 0 | $\{e_{-1}, e_{-2}\}$ | $\{e_{-1}, e_{-2}\}$ | $I_{2\times 2}$ | $I_{2\times 2}$ |
| 1 | $\{e_0, e_{-2}\}$ | $\{e_0, e_{-2}\}$ | $I_{2\times 2}$ | $I_{2\times 2}$ |
| 2 | $\{e_0, e_1\}$ | $\{e_0, e_1\}$ | $I_{2\times 2}$ | $I_{2\times 2}$ |
| 3 | $\{e_2, e_1\}$ | $\{e_0, e_1\}$ | $I_{2\times 2}$ | $I_{2\times 2}$ |
| 4 | $\{e_2, e_1\}$ | $\{e_3, e_1\}$ | $I_{2\times 2}$ | $I_{2\times 2}$ |
| 5 | $\{e_2, e_4\}$ | $\{e_3, e_1\}$ | $I_{2\times 2}$ | $I_{2\times 2}$ |
| 6 | $\{e_2, e_4\}$ | $\{e_3, e_5\}$ | $I_{2\times 2}$ | $I_{2\times 2}$ |
| 7 | $\{e_2, e_6\}$ | $\{e_6, e_5\}$ | $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ |

Figure 2: Updating process for frontier edge sets and frontier edge matrix

We now show that it is always possible to choose suitable coding vectors for $e_t$. We are given a set of full rank $h \times h$ matrices $M_i^t, T_i \in T$. The coding vector for $e_t$ needs to be chosen from the span of a set of at most $|T|$ vectors where each one of those vectors belongs to a different $M_i^t$. Once the coding vector $\eta(e_t)$ has been determined we replace the original vectors by $\eta(e_t)$ and ensure that the new matrices are still full rank. The lemma below establishes that as long as the field size is large enough, this is always possible.

**Lemma 1** *Let $M_i, i = 1, 2, ..., k'$ be arbitrary $n \times n$ nonsingular matrices over GF(q). Let $v_i$ denote a row vector of $M_i$, $\mathcal{L}\ \{v_1, v_2, ..., v_{k'}\}$ denote the span of the set of vectors $\{v_1, v_2, ..., v_{k'}\}$ and $\mathcal{L}\ \{M_i - \{v_i\}\}$ denote the span of the row vectors of $M_i$ excluding $v_i$. If $k' < q$, then there exists a vector $v \in \mathcal{L}\ \{v_1, v_2, ..., v_{k'}\}$, such that the matrix $\{M_i - \{v_i\}\} \cup \{v\}$ are nonsingular for $i = 1, ....k'$.*

*Proof.* Let us define the dimension of vector set $\{v_1, v_2, ..., v_{k'}\}$ denoted $\dim\{v_1, v_2, ..., v_{k'}\}$ in this way:

$$\dim\{v_1, v_2, ..., v_{k'}\} = \text{number of linearly independent column vectors in the set}$$

Suppose that $\dim\{v_1, v_2, ..., v_{k'}\} = k$ where $k \leq k'$. Then the following statement holds.

$$\text{The total number of vectors in } \mathcal{L}\ \{v_1, v_2, ..., v_{k'}\} = q^k. \tag{1}$$

We know that any linear combination of vectors in $\mathcal{L}\ \{v_1, v_2, ..., v_{k'}\}$ that has a non-zero coefficient associated with $v_i$ does not belong to $\mathcal{L}\ \{M_i - \{v_i\}\}$. This is because by assumption each $M_i$ is full rank. This implies that $\{M_i - \{v_i\}\}$ has rank $n-1$ and does not contain any linear combination that has a non-zero coefficient associated with $v_i$. Therefore if a vector belongs to both $\mathcal{L}\ \{M_i - \{v_i\}\}$ and $\mathcal{L}\ \{v_1, v_2, ..., v_k'\}$ must have a zero coefficient for $v_i$ and the number of such vectors in $\mathcal{L}\ \{v_1, v_2, ...v_k'\}$ is at most $q^{k-1}$. Thus, we have established that the number of non-zero vectors in $\mathcal{L}\ \{M_i - \{v_i\}\} \cap \mathcal{L}\ \{v_1, v_2, ...v_k'\} \leq q^{k-1} - 1$. Now, the total number of vectors that belongs to one of these intersections $\mathcal{L}\ \{M_i - \{v_i\}\} \cap \mathcal{L}\ \{v_1, v_2, ..., v_k'\}$ for $i = 1, ...k'$ is at most $k'(q^{k-1} - 1) < k'q^{k-1} < q^k$ if $k' < q$.

So, we conclude that the number of vectors that do not belong to any of these intersection is at least $q^k - k'q^{k-1} > 0$ for $k' < q$ which means that a vector that does not belong to any of these intersections $\mathcal{L}\ \{M_i - \{v_i\}\} \cap \mathcal{L}\ \{v_1, v_2, ...v_{k'}\}$ can be found.

In our problem, $n = h$ and $k' =$ number of terminals. It means that if $q > |T|$, then we can find a suitable coding vector for each edge $e_t$ for all $t$. This implies that a multicast network code can be constructed using the LIF algorithm.