

Lecture 5

1 Finite Fields (contd.)

We can construct a finite field $GF(q)$ where q is prime. The add and multiplication operation are module q add and module q multiplication respectively. We can also extend a finite field $GF(q)$ (q is prime) to $GF(q^m)$, where m is a positive integer. For example, to extend $GF(2)$ to $GF(2^2)$, we can interpret elements of $GF(2^2)$ as polynomials of degree at most 1 with coefficients in $GF(2)$. In this case the elements of $GF(4)$ are $\{0, 1, x, x + 1\}$. The operations on $GF(4)$ are performed modulo $(x^2 + x + 1)$. The polynomial $(x^2 + x + 1)$ is called a irreducible polynomial and is used to construct the finite field.

It is important to note that most computations over $GF(q^m)$ are equivalent to those in real field. For example, in the reals, suppose \mathbf{A} is a $n \times n$ matrix, \vec{x} and \vec{b} are $n \times 1$ column vectors, and

$$\mathbf{A}\vec{x} = \vec{b}.$$

Then we have

$$\vec{x} = \mathbf{A}^{-1}\vec{b}, \text{ as long as } \mathbf{A} \text{ is non-singular, or } \det(\mathbf{A}) \neq 0.$$

Exactly the same result holds the case when the elements of the matrix and vectors are from a finite field. Given $\mathbf{A}\vec{x} = \vec{b}$, it has a unique solution if and only if $\det(\mathbf{A}) \neq 0$, where the determinant is over the finite field.

The formalism of finite fields is useful in networking because a packet in a network that typically consists of a set of binary digits can be treated as a member of a finite field. For example, a 6-bit packet 010100 can be treated as an element belonging to $GF(2^6)$ by treating it as the polynomial $0x^5 + 1x^4 + 0x^3 + 1x^2 + 1x + 0$. The usage of finite fields provides us with a means of processing packets mathematically.

2 Network coding

We now introduce the concept of network coding. Consider a directed graph with unit capacity edges, as depicted in Fig.1 where there is one source node and one terminal node. Unit-capacity edges imply that each edge can transmit one symbol from $GF(2^m)$ per unit time (the choice of m is not fixed). In general a source node could be observing multiple sources (note the distinction between a source node and a source) as depicted in Fig. 2. We also assume that each such source is unit-rate, i.e. each source produces a symbol from a finite field e.g. $GF(2^m)$ at each time instant. If a source node is observing a source of higher rate, we shall split that source into multiple unit-rate sources. e.g. consider that the source node wants to transmit a high-resolution video of a football game to multiple terminals and suppose that at every instant of time (say per milli-second) it generates an image of size 1MB. Also assume that each terminal has a max-flow from the source node of at least 1MB per milli-second but the capacity of each link in the network is only 0.1 MB per milli-second. In this case the source would split the 1MB image into 10 sources of rate 0.1 MB per milli-second. Thus, the unit-capacity edge in this network would refer to an edge of capacity

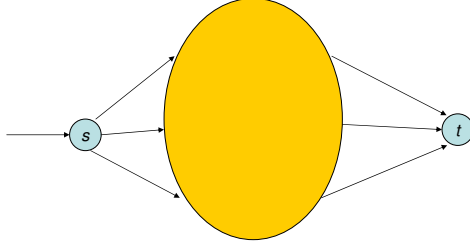


Figure 1: A network model for network coding.

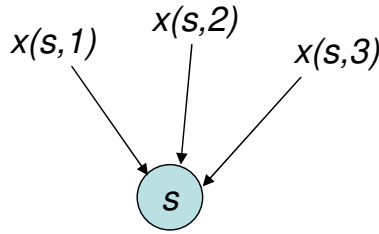


Figure 2: A source node is observing 3 sources $x(s, 1), x(s, 2), x(s, 3)$.

0.1 MB per milli-second and there would be 10 sources being injected into the network. More generally, a source node could be observing multiple sources as depicted in Fig.2

Let $x(s, k)$ denote the k^{th} source observed by the source node s . Thus, given $G = (V, E)$, we can specify the source inputs corresponding to each sources.

$$\begin{aligned}
 \{x(s_1, 1), x(s_1, 2)\} & : s_1 \\
 \{x(s_2, 1), x(s_2, 2), x(s_2, 3)\} & : s_2 \\
 \dots & \\
 \{x(s_k, 1)\} & : s_k
 \end{aligned}$$

We let $\mu(v)$ denote the number of sources observed at node v and the total number of sources by μ . In the above example we have $\mu(s_1) = 2$. The terminals specify their demands by requesting a subset of the sources as shown below. A connection is said to be feasible if the demands of each terminal can be satisfied.

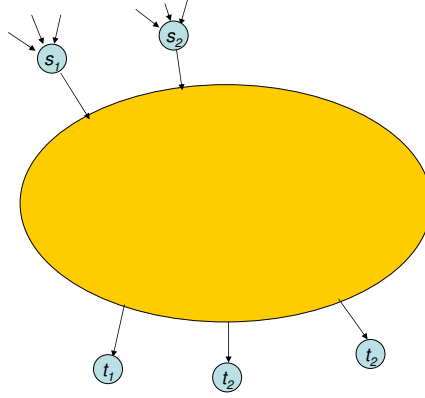


Figure 3: A general example, the source nodes are observing many sources and the terminal nodes is requesting a subset of the sources.

$$\begin{aligned}
 \{x(s_1, 1), x(s_2, 1)\} & : t_1 \\
 \{x(s_1, 2), x(s_2, 2), x(s_3, 2)\} & : t_2 \\
 \dots & \\
 \{x(s_m, n)\} & : t_k
 \end{aligned}$$

Similarly we denote the number of outputs requested at node v by $\nu(v)$ and the total number of outputs by ν .

Definition 1. *Delay free F_{2^m} -linear network.* Let $G = (V, E)$ be a delay-free communication network. It is said to be F_{2^m} -linear if for all edges the signal $y(e)$ on an edge $e \in E$ satisfies

$$y(e) = \sum_1^{\mu(v)} \alpha_{l,e} x(v, l) + \sum_{e': \text{head}(e') = \text{tail}(e)} \beta_{e'e} y(e')$$

where the $\alpha_{l,e}$'s and $\beta_{e'e}$'s are elements of F_{2^m} .

Fig. 4 shows an example of a linear operation in the network at a node v . The adjective delay-free means that if the inputs appear on the edges coming to node v (from other nodes and sources), outputs appear on outgoing edges of v immediately. This is an idealization and we shall discuss its implications later on.

Typically we shall be interested in finding a set of $\{\alpha_{l,e}\}$'s and a set of $\{\beta_{e'e}\}$'s such that the demands of the terminals are met. From the received signals on the inputs of a terminal, it should be able to uniquely recover the signals that it is interested in. We now consider an example with one source node, with three sources and a terminal node that wants to recover all the sources. The example is shown in Fig.5. We have three sources at v_1 and three paths from v_1 to v_4 :

- i) $e_1 \rightarrow e_5$.

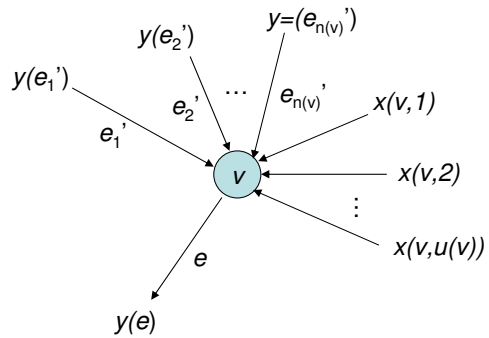


Figure 4: A node in a network with edges incoming from other nodes and sources, and one edge outgoing from it.

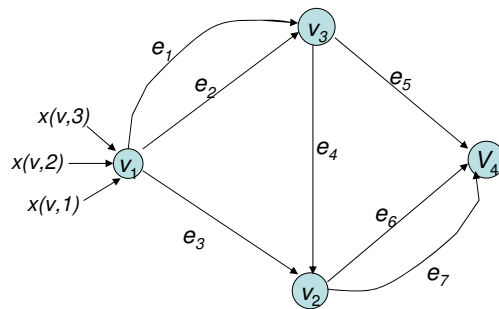


Figure 5: An example for building transfer matrix.

ii) $e_2 \rightarrow e_4 \rightarrow e_6$.

iii) $e_3 \rightarrow e_7$.

Let $y(e_i)$ denote the symbol transmitted on edge e_i . We now determine the input-output relationship of this linear network. The symbols on the outgoing edges from node v_1 , i.e. e_1, e_2, e_3 , are linear combinations of the sources $x(v, 1), x(v, 2), x(v, 3)$:

$$y(e_1) = \alpha_{1e_1}x(v, 1) + \alpha_{2e_1}x(v, 2) + \alpha_{3e_1}x(v, 3) \quad (1)$$

$$y(e_2) = \alpha_{1e_2}x(v, 1) + \alpha_{2e_2}x(v, 2) + \alpha_{3e_2}x(v, 3) \quad (2)$$

$$y(e_3) = \alpha_{1e_3}x(v, 1) + \alpha_{2e_3}x(v, 2) + \alpha_{3e_3}x(v, 3) \quad (3)$$

let α be a 3×3 matrix, equations above can be written as matrix form:

$$\begin{bmatrix} y(e_1) \\ y(e_2) \\ y(e_3) \end{bmatrix} = \alpha \begin{bmatrix} x(v, 1) \\ x(v, 2) \\ x(v, 3) \end{bmatrix}$$

Similarly, $y(e_4), y(e_5), y(e_6), y(e_7)$ are the linear combinations of symbols in their incoming edges:

$$y(e_4) = \beta_{e_1e_4}y(e_1) + \beta_{e_2e_4}y(e_2) \quad (4)$$

$$y(e_5) = \beta_{e_1e_5}y(e_1) + \beta_{e_2e_5}y(e_2) \quad (5)$$

$$y(e_6) = \beta_{e_3e_6}y(e_3) + \beta_{e_4e_6}y(e_4) \quad (6)$$

$$y(e_7) = \beta_{e_3e_7}y(e_3) + \beta_{e_4e_7}y(e_4) \quad (7)$$

At the sink v_4 , the received symbols are $y(e_5), y(e_6), y(e_7)$, they can be represented in a matrix form:

$$\begin{bmatrix} y(e_5) \\ y(e_6) \\ y(e_7) \end{bmatrix} = \beta \cdot \alpha \cdot \begin{bmatrix} x(v, 1) \\ x(v, 2) \\ x(v, 3) \end{bmatrix}$$

where β and α are given by

$$\beta = \begin{bmatrix} \beta_{e_1e_5} & \beta_{e_2e_5} & 0 \\ \beta_{e_1e_4}\beta_{e_4e_6} & \beta_{e_2e_4}\beta_{e_4e_6} & \beta_{e_3e_6} \\ \beta_{e_1e_4}\beta_{e_4e_7} & \beta_{e_2e_4}\beta_{e_4e_7} & \beta_{e_3e_7} \end{bmatrix}$$

and

$$\alpha = \begin{bmatrix} \alpha_{1e_1} & \alpha_{2e_1} & \alpha_{3e_1} \\ \alpha_{1e_2} & \alpha_{2e_2} & \alpha_{3e_2} \\ \alpha_{1e_3} & \alpha_{2e_3} & \alpha_{3e_3} \end{bmatrix}$$

We wish to recover $[x(v, 1) \ x(v, 2) \ x(v, 3)]^T$ from the received symbols $[y(e_5) \ y(e_6) \ y(e_7)]^T$. Note that as long as the square matrix $\beta \cdot \alpha$ is invertible, the unique decoding is possible. The condition is equivalent to

$$\det(\beta \cdot \alpha) \neq 0,$$

which is in turn equivalent to $\det \alpha \neq 0$ and $\det \beta \neq 0$. It is important to note that $\det \alpha$ is a multivariate polynomial in the $\alpha_{l,e}$'s. For example,

$$\det \begin{bmatrix} \alpha_{1e_1} & \alpha_{2e_1} \\ \alpha_{1e_2} & \alpha_{2e_2} \end{bmatrix} = \alpha_{1e_1}\alpha_{2e_2} - \alpha_{2e_1}\alpha_{1e_2}.$$

The question of whether there exist assignments to $\alpha_{l,e}$'s and $\beta_{e'e}$'s such that $\det \alpha \neq 0$ and $\det \beta \neq 0$ is equivalent to the question of whether there exist $\alpha_{l,e}$'s and $\beta_{e'e}$'s such that the corresponding polynomial evaluates to a non-zero value.

In general this is not a trivial question since there are examples of polynomial over a fixed finite field that would evaluate to zero over all points in the field. As an example consider the polynomial $p(z) = z^2 + z$ over $GF(2)$. It evaluates to 0 when $z = 0$ (since $0^2 + 0 = 0$) and when $z = 1$ (since $1^2 + 1 = 0$) i.e. over $GF(2)$ there does not exist any assignment to z such that it results in a non-zero value.

We now consider the following special case of a general network coding problem. Source node s has sources

$$\{x(s, 1), \dots, x(s, n)\}$$

and there is a set of terminals T . The transfer matrix to each terminal is denoted by M_1, \dots, M_i so that

$$\vec{y}_{T_i} = M_i \vec{x} \text{ for all } T_i$$

. Each terminal wants to decode the vector \vec{x} . This implies that

$$\det(M_i) \neq 0 \forall i,$$

which is equivalent to

$$\prod_{i=1}^{|T|} \det(M_i) \neq 0.$$

It should be clear that the problem of checking whether each terminal's demand can be satisfied is equivalent to checking whether there exist an assignment of $\alpha_{l,e}$'s and $\beta_{e'e}$'s such that the product of determinants is non-zero. There are two cases that need to be considered before this question can be addressed satisfactorily.

- a) Is the product a zero polynomial? i.e. regardless of the assignment of the indeterminates it always a polynomial with all zero coefficients.
- b) Suppose it is not a zero polynomial, then are we guaranteed that a valid assignment exists.

We shall first tackle the question (a). We demonstrate by an example that in some cases the polynomial can be the zero polynomial. Consider the network in Fig. 6 below. We first find the transfer matrix from source node v_1 to terminal node v_5 . The max-flow of this network from v_1 to v_5 is 2. The symbols on edges e_1 through e_7 have their linear combination representation in (1)-(7). Now we consider symbols on edges e_8, e_9 and e_{10} denoted by $z(e_8), z(e_9)$ and $z(e_{10})$. They are linear combinations of $y(e_3)$ and $y(e_4)$ and can be expressed as,

$$\begin{bmatrix} z(e_8) \\ z(e_9) \\ z(e_{10}) \end{bmatrix} = \begin{bmatrix} \gamma_{e_3e_8} & \gamma_{e_4e_8} \\ \gamma_{e_3e_9} & \gamma_{e_4e_9} \\ \gamma_{e_3e_{10}} & \gamma_{e_4e_{10}} \end{bmatrix} \begin{bmatrix} y(e_3) \\ y(e_4) \end{bmatrix}$$

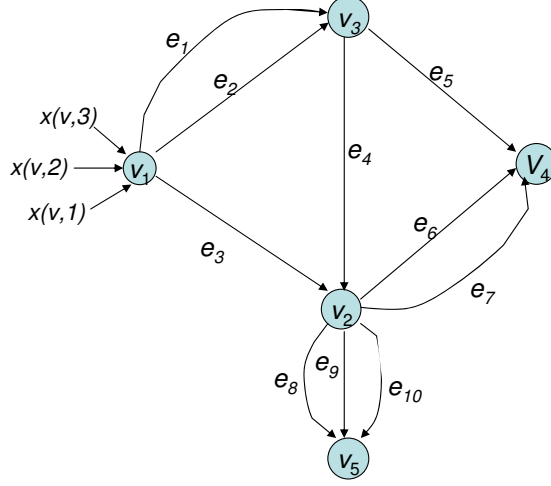


Figure 6: An example of network where the transfer matrix of v_5 is not invertible under any assignment of coefficients.

From (4), we know that

$$y(e_4) = \beta_{e_1e_4}y(e_1) + \beta_{e_2e_4}y(e_2).$$

Using this expression in above equations we obtain,

$$\begin{bmatrix} z(e_8) \\ z(e_9) \\ z(e_{10}) \end{bmatrix} = \begin{bmatrix} \gamma_{e_4e_8}\beta_{e_1e_4} & \gamma_{e_4e_8}\beta_{e_2e_4} & \gamma_{e_3e_8} \\ \gamma_{e_4e_9}\beta_{e_1e_4} & \gamma_{e_4e_9}\beta_{e_2e_4} & \gamma_{e_3e_9} \\ \gamma_{e_4e_{10}}\beta_{e_1e_4} & \gamma_{e_4e_{10}}\beta_{e_2e_4} & \gamma_{e_3e_{10}} \end{bmatrix} \begin{bmatrix} y(e_1) \\ y(e_2) \\ y(e_3) \end{bmatrix}.$$

Further, we represent $[y(e_1) \ y(e_2) \ y(e_3)]^T$ in terms of $[x(v,1) \ x(v,2) \ x(v,3)]^T$, and denote the matrix above by $\boldsymbol{\mu}$. Then

$$\begin{bmatrix} z(e_8) \\ z(e_9) \\ z(e_{10}) \end{bmatrix} = \boldsymbol{\mu} \cdot \boldsymbol{\alpha} \cdot \begin{bmatrix} x(v,1) \\ x(v,2) \\ x(v,3) \end{bmatrix}$$

so that $\boldsymbol{\mu} \cdot \boldsymbol{\alpha}$ is the net transfer matrix. We will show that $\det(\boldsymbol{\mu}) = 0$ under any assignment of coefficients, which means that the determinant of transfer matrix equals to 0 under any assignment of coefficients. In fact, if we observe the first two columns in $\boldsymbol{\mu}$, they could be written as

$$\beta_{e_1e_4} \begin{bmatrix} \gamma_{e_4e_8} \\ \gamma_{e_4e_9} \\ \gamma_{e_4e_{10}} \end{bmatrix} \text{ and } \beta_{e_2e_4} \begin{bmatrix} \gamma_{e_4e_8} \\ \gamma_{e_4e_9} \\ \gamma_{e_4e_{10}} \end{bmatrix}$$

i.e. they are multiple of each other since $\beta_{e_1e_4}$ and $\beta_{e_2e_4}$ are elements from a finite field. If we multiply $\beta_{e_2e_4}$ with the element $\beta_{e_1e_4}\beta_{e_2e_4}^{-1}$, the result is $\beta_{e_1e_4}$. This implies that $\det \boldsymbol{\mu} = 0$ regardless of the assignments of the variables.

On inspection one realizes that the reason for this is that there aren't enough disjoint paths going into the terminal v_5 from v_1 . Both symbols $y(e_1), y(e_2)$ pass through edge e_4 to reach v_5 , which causes that two columns in $\boldsymbol{\mu}$ share the common coefficients.

Note that in this case the max-flow to v_5 is equal to two and it is clear that no processing can ever result in a higher flow to v_5 . Therefore the result makes sense. However, if the max-flow is high enough i.e. in this case if it was three, then the disjoint paths algorithm would yield a set of paths over which the symbols could simply be routed. This would correspond to a 0 – 1 assignment of the indeterminates $\alpha_{l,e}$'s and $\beta_{e'e}$'s and $\gamma_{e'e}$'s i.e. we would have at least one assignment of the variable such the determinant is non-zero. This would ensure that the determinant is not the zero polynomial.