Lecture 28

Note Title 4/25/2012

* Def: A feasible network code that sends K symbols $Y_1, \cdots, Y_K \in GF(2^b)$ from a single source s to a single desti. d is defined by M_e : the coded message M on edge e s.t.

$$M_{(s,u)} = \underbrace{f_{(s,u)}}_{\text{A function indexed by } (s,u)}(Y_1, \cdots, Y_K) \in GF(2^b)$$

For $v \neq s$

$$M_{(v,w)} = f_{(v,w)}\left(M_{(u,v)} : (u,v) \in In(v)\right)$$

$\boxed{\text{decoding function}}$

$$(\hat{Y}_1, \cdots, \hat{Y}_K) = f_{DEC}\left(M_{(v,t)} : (v,t) \in In(t)\right)$$

s.t. $\forall k$, $\hat{Y}_k = Y_k$ for all $Y_1, \cdots, Y_K$ values.

* The optimization question is: Given the underlying network topology, what is the largest K^* info symbols of a <u>feasible</u> network code by arbitrarily design the f_e & f_{DEC} functions.

* It can be generalized to a multicast session that sends $Y_1, \cdots, Y_k$ from s to multiple destination t_i.

The only changes are (all of t_i requested the same # of symbols.)

$$(\hat{Y}_{i,1}, \cdots, \hat{Y}_{i,k}) = f_{i,DEC}\left(M_{(v,t_i)} : (v,t_i) \in In(t_i)\right)$$

$\hat{Y}_{i,k} = Y_k \ \forall k$, & $Y_1 \cdots Y_k$ values

* The goal is again to maximize K.

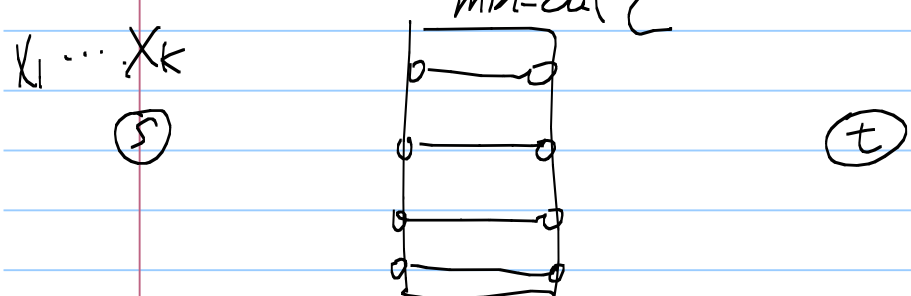
* We first answer the question for a single-source s, single destination t scenario.

Theorem:

* For a single (s,t) unicast session, the largest K^* is equal to the min (s,t)-cut value.

Pf: $K \leq \min (s,t)\text{-cut} = |C^*|$ for any choice of f_e & f_{DEC}

min-cut C^*



Since C^* is a cut

$$(\hat{Y}_1, \cdots, \hat{Y}_k) = f_{DEC}(M_{(v,t)} : (v,t) \in E)$$

$$= f'(M_{(u,v)} : (u,v) \in C^*).$$

Since there are $(2^b)^{|C^*|}$ possible vector values of $\{M_{(u,v)} : (u,v) \in C^*\}$, there are at most $|2^b|^{|C^*|}$ possible values of $(\hat{Y}_1, \cdots, \hat{Y}_k)$.

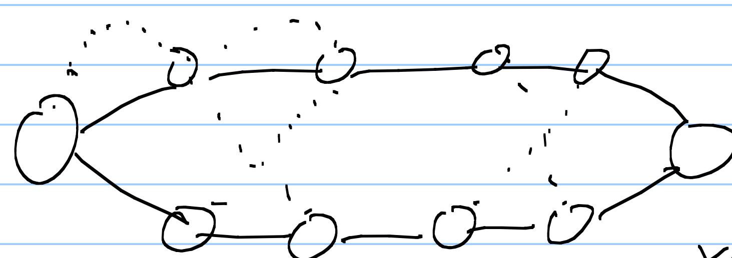
Nonetheless, the feasibility condition requires that there are $(2^b)^K$ different values of $(\hat{Y}_1, \cdots, \hat{Y}_k)$

$$\Rightarrow |2^b|^K \leq |2^b|^{|C^*|} \Leftrightarrow K \leq |C^*|$$

✓

pf: $K^* \geq |C^*|$.

By the mc/MF theorem, $\exists\ |C^*|$ edge disjoint paths from s to t. We can then construct our f_e functions that "relay" $|C^*|$ symbols directly along the $|C^*|$ edge-disjoint paths. Those edges not in the edge-disjoint paths simply remain silent.



f_{DEC} just reproduces $\hat{Y}_1 \cdots \hat{Y}_{|C^*|}$ from the received $Y_1 \cdots Y_{|C^*|}$

$\Rightarrow K^* = |C^*|$

Corollary:
For an (s,t) unicast sessions, one can achieve the optimal min-cut capacity by multipath routing without any pkt mixing.

A more interesting scenario is the $(s, \{t_i\})$ multicast sessions. (All t_i are interested in the same amount of info.)

Theorem: Suppose we can multicast K symbols from s to all destinations $\{t_i\}$.

then the largest K^* satisfies

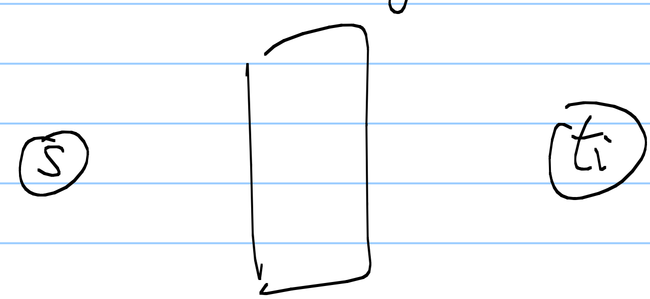
$$K^* = \min_{\{t_i\}} \left(\min\ (s, t_i)\text{-cut value} \right)$$

when the underlying $GF(2^b)$ $2^b \to \infty$

Pf: $K \le \min_{\{t_i\}} \left(\min\ (s, t_i)\text{-cut value} \right)$

for any f_e & $f_{i,DEC}$

By the same argument



$\Rightarrow K \le \min\ (s, t_i)\text{-cut value}$ for all t_i

$$\Rightarrow K \le \min_{\{t_i\}} \left(\min\ (s, t_i)\text{-cut value} \right)$$

pf: $K^* \geq \min_{t_i} \left(\min (s, t_i)\text{-cut value} \right)$

when $GF(2^b)$ is sufficiently large.

By construction of linear network codes.

* Linear network codes.

Each $f_{(v,w)}\left(M_{(u,v)} : (u,v) \in In(v)\right)$

$$= \sum_{(u,v) \in In(v)} \alpha_{(u,v),(v,w)} M_{(u,v)}$$

where $\alpha_{(u,v),(v,w)} \in GF(2^b)$ is the linear coefficient for $M_{(u,v)}$

Ex: $f_{e_3}(M_{e_1}, M_{e_2}) = M_{e_1} + 2M_{e_2}$

$f_{i,DEC}\left(M_{(u,t_i)} : (u,t_i) \in In(t_i)\right)$

$= \beta^{[i]} \underbrace{\left[M(u,t_i) : (u,t_i) \in In(t_i) \right]}_{\text{a column vector of dim } |In(t_i)|}$

$\beta^{[i]}$ = a $K^*_{[i]}$ by $|In(t_i)|$ matrix

$= (\beta_{k,e})$

pf: $K^* \geq \min_{t_i} \left(\min (s, t_i)\text{-cut value} \right)$

when $GF(2^b)$ is sufficiently large.

By construction of linear network codes.

* Linear network codes.

Each $f_{(v,w)}\left(M_{(u,v)} : (u,v) \in In(v)\right)$

$$= \sum_{(u,v) \in In(v)} \alpha_{(u,v),(v,w)} M_{(u,v)}$$

where $\alpha_{(u,v),(v,w)} \in GF(2^b)$ is the linear coefficient for $M_{(u,v)}$

Ex: $f_{e_3}(M_{e_1}, M_{e_2}) = M_{e_1} + 2M_{e_2}$

$f_{i,DEC}\left(M_{(u,t_i)} : (u,t_i) \in In(t_i)\right)$

$= \beta^{[i]} \underbrace{\left[M(u,t_i) : (u,t_i) \in In(t_i)\right]}$ a column vector of dim $|In(t_i)|$

$\beta^{[i]}$: a $K^*_{[i]}$ by $|In(t_i)|$ matrix

$= (\beta_{k,e})$

* A linear network code is decided by its $\alpha_{(u,v),(v,w)}$ & $\beta_{k,e}^{[i]}$ coefficients.

Once α's are fixed, each $M_e = \sum_{k=1}^{K} \gamma_{e,k} Y_k$ by concatenating the ENC function.

$\gamma_{e,k}$ is termed the <u>global coding coefficients</u> for the M_e coded symbol.

* The feasibility of a network code

$\Leftrightarrow \exists B^{[i]}$ for all i

$$\Leftrightarrow \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_K \end{pmatrix} = B^{[i]} \left[M_{(v,t_i)} : (v,t_i) \in In(t_i) \right]$$

$$= B^{[i]} \begin{pmatrix} \gamma_{e_1,1} & \cdots & \gamma_{e_1,K} \\ \gamma_{e_2,1} & & \\ \gamma_{e_3,1} & & \\ \vdots & & \\ \gamma_{e_{|In(t_i)|},1} & & \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_K \end{pmatrix}$$

$$= \begin{pmatrix} Y_1 \\ \vdots \\ Y_K \end{pmatrix} \quad \text{by feasibility}$$

$\Leftrightarrow$ the global coding coeff. on $In(t_i)$

$$\boxed{\begin{pmatrix} \gamma_{e,1} & & \gamma_{e_1,K} \\ \gamma_{e_2,1} & & \vdots \\ \vdots & & \vdots \end{pmatrix}}$$

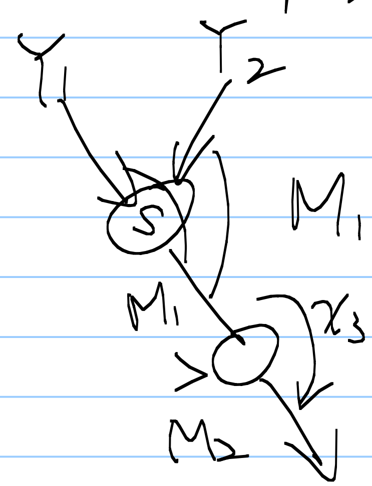
is of full column rank.

Without loss of generality, we can assume $|In(t_i)| = k$, then we require

To have full rank (invertible)

* A good notation (though a bit confusing in the first place) is to write $\alpha_{(u,v),(v,w)}$ as $x_{(u,v),(v,w)}$ to emphasize that $\alpha_{(u,v),v}$ is a design variable. We use $\vec{x}$ as shorthand for $\{x_{(u,v),(v,w)} : \forall (u,v), (v,w) \in E\}$

* $\Rightarrow$ Each $\gamma_{e,k}$ is a polynomial of $\vec{x}$. Ex:



$$M_1 = x_1 Y_1 + x_2 Y_2$$

$$\begin{aligned} M_2 &= x_3 M_1 \\ &= \underbrace{x_1 x_3}_{\gamma_{2,1}} Y_1 + \underbrace{x_2 x_3}_{\gamma_{2,2}} Y_2 \end{aligned}$$

* If we focus on

the global coding coeff. on $In(t_i)$

$$\Delta_i = \begin{pmatrix} \gamma_{e,1} & & \gamma_{e_1,K} \\ \gamma_{e_2,1} & & \vdots \\ \vdots & & \vdots \end{pmatrix},$$ since

each entry is a polynomial of $\vec{x}$

$\Rightarrow$ $\det(\Delta)$ is also a polynomial of

$\vec{x}$: $\det(\Delta_i) = f_{\Delta_i}(\vec{x})$ | each $\vec{x}$ is a network code

$\Rightarrow$ Our goal is to design one $\vec{x}$ s.t

$$\det(\Delta_i) = f_{\Delta_i}(\vec{x}) \neq 0 \text{ for}$$

all t_i.

Or equivalently, our goal is to

design one $\vec{x}$ s.t.

$$f_{overall}(\vec{x}) \triangleq \left[\prod_{t_i} f_{\Delta_i}(\vec{x})\right] \neq 0$$

Case 1: $f_{overall}(\vec{x})$ is a zero polynomial. We are doomed in this case. So we need to show that case 1 is impossible provided

$$K^* \le \min_{t_i} \left(\min (s,t_i)\text{-cut values}\right)$$

Case 2: $f_{overall}(\vec{x})$ is not a zero polynomial. Q: does not being a zero-polynomial automatically implying that $\exists \vec{x}$ s.t. $f_{overall}(\vec{x}) \neq 0$?

Ans: Not necessarily when we focus on a finite field.

Ex: x is a scalar from $GF(2)$

i.e. $x \in \{0,1\}$.

A non-zero polynomial: $f(x) = x(x-1)$

$f(x) = 0$ for all $x \in \{0,1\}$.

Fortunately when the $GF(2^b)$ is large we have $|2^b|^{\dim(\vec{x})}$ choices of $\vec{x}$. Since $f_{overall}(\vec{x})$ is non-zero and has a finite degree $\deg(f_{overall})$ ($\because$ the graph G is finite), it has at most $\deg(f_{overall})$ zeros

by choosing a large $GF(2^b)$, we have

$$\deg(f_{overall}) < |2^b|^{\dim(\vec{x})}$$

$\Rightarrow \exists$ at least one $\vec{x}$ s.t.

$f_{\text{overall}}(\vec{x}) \neq 0$. Case 2 is proven.

Q: How to prove that $f_{overall}$ must not be a zero polynomial?

A: We note that for any two polynomials $g_1(x)$ & $g_2(x)$. $f(\vec{x}) = g_1(\vec{x}) \cdot g_2(\vec{x})$ is not zero if both $g_1(\vec{x})$ & $g_2(\vec{x})$ are non-zero.

$\Rightarrow$ We only need to show that $f_{\Delta i}(\vec{x})$ is a non-zero. polynomial

Q: How to show $f_{\Delta i}(\vec{x})$ being non-zero?

Ans: Construct any $\vec{x}$ s.t. t_i can receive/decode the $Y_1, \cdots, Y_k$ info symbols.

Observation: this construction of $\vec{x}$ does not need to take care of all destinations $\{t_i\}$, but only a single t_i since we are focusing on $f_{\Delta i}(\vec{x})$ rather than $f_{overall}(\vec{x})$

$\Rightarrow$ This observation drastically simplify the construction as given

$$K = \min_{t_i} \left(\min (s, t_i)\text{-cut value} \right)$$

$$\leq \min (s, t_i)\text{-cut value},$$

We can route/relay the K info pkts along the $|C_i^*|$ edge-disjoint paths.

$\Rightarrow$ The corresponding "relaying" choice of $\vec{x}_{relay,i}$ suffices to communicate K info pkts to t_i $\Rightarrow f_{\Delta i}(\vec{x}_{relay,i}) \neq 0$

$\Leftrightarrow f_{\Delta i}(\cdot)$ is non-zero.

A summary: The joint feasibity of all destinations t_i with sufficiently large $GF(2^b)$

$\Leftrightarrow$ $f_{\text{overall}}(\vec{x}) = \prod_{t_i} f_{\Delta_i}(\vec{x}) \neq 0$

$\Leftrightarrow$ $f_{\Delta_i}(\vec{x}) \neq 0 \quad \forall i$

$\Leftrightarrow$ The feasibility for individual destinations as if all other destinations does not exist.

$\Leftrightarrow$ $K \leq$ min(s,t_i)-cut value for all i. is sufficient

$\Rightarrow K^* = \min_{t_i} (\min (s,t_i)\text{-cut value})$