Network Coding

An Instant Primer

Kanat Tangwongsan

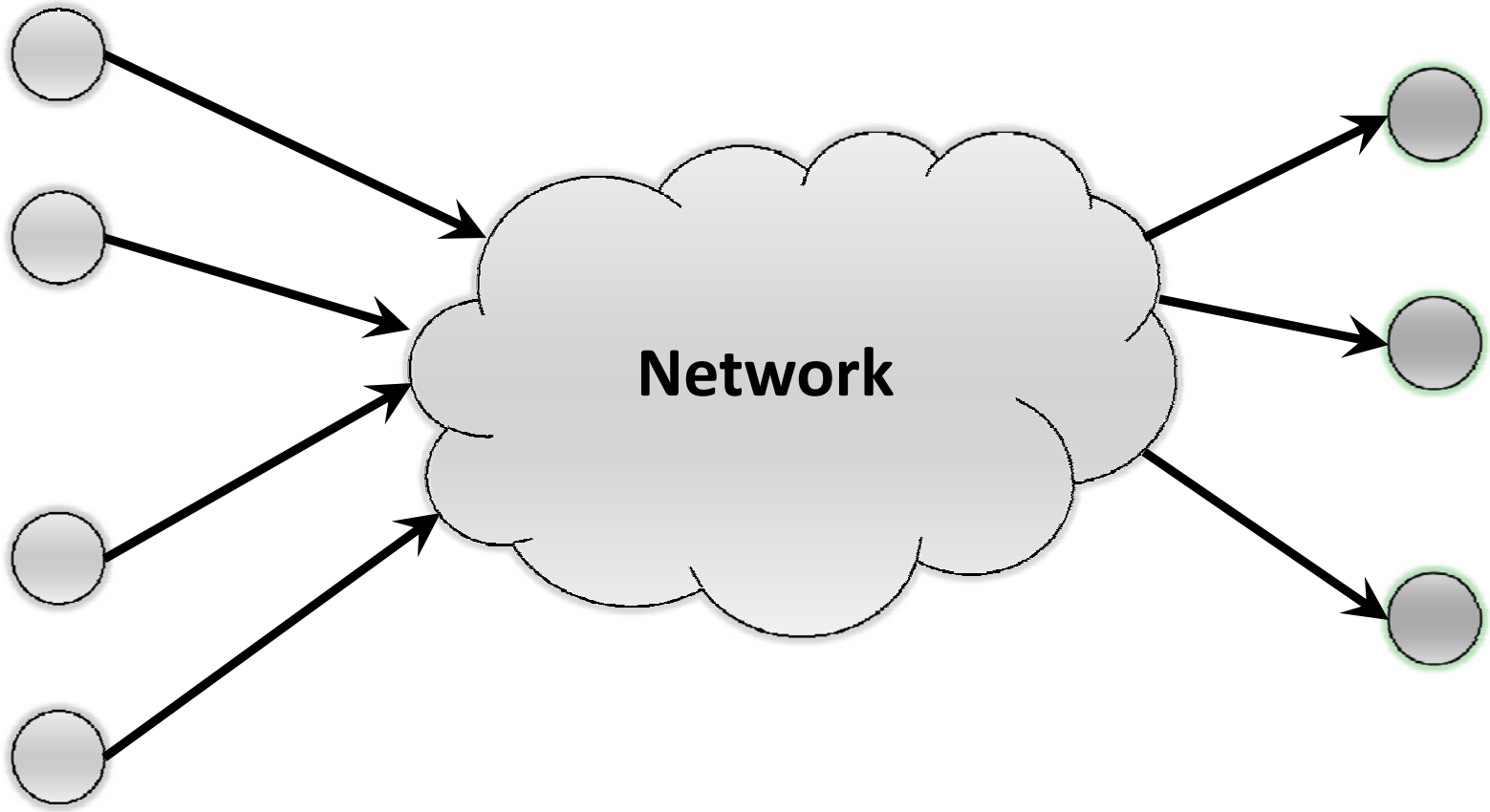
ktangwon@cs.cmu.edu

Sources of Materials

Links from Course Web Page

- Network Coding: An Instant Primer
  - Fragouli, Boudec, and Widmer.
- Network Coding – an Introduction
  - Koetter and Medard
- On Randomized Network Coding
  - Ho, Medard, Shi, Effros, and Karger
- An Algebraic Approach to Network Coding
  - Ralf Koetter and Medard

Basic Goal:
Throughput + Robustness



Today's Network:
Data as rigid objects

Network Coding:
Nodes can **recombine** packets

Example: A sends ‘a’ to B, B sends ‘b’ to A

Why is Network Coding interesting?

(1) Enormous Throughput Improvement

Thm [Ahlswede, Cai, Li, and Yeung, 2000]

There exist multicast problems such that the gap between routing and network coded strategies is *arbitrarily* large.

Why is Network Coding interesting?

(2) Robustness for free

Thm [Deb and Medard, 2004]

Rumor propagation on n nodes and $O(n)$ messages take $\Theta(n)$ rounds.

$\Theta(n \log n)$ rounds in typical decentralized se

Are we too optimistic?

Thm [Li and Li, 2004]

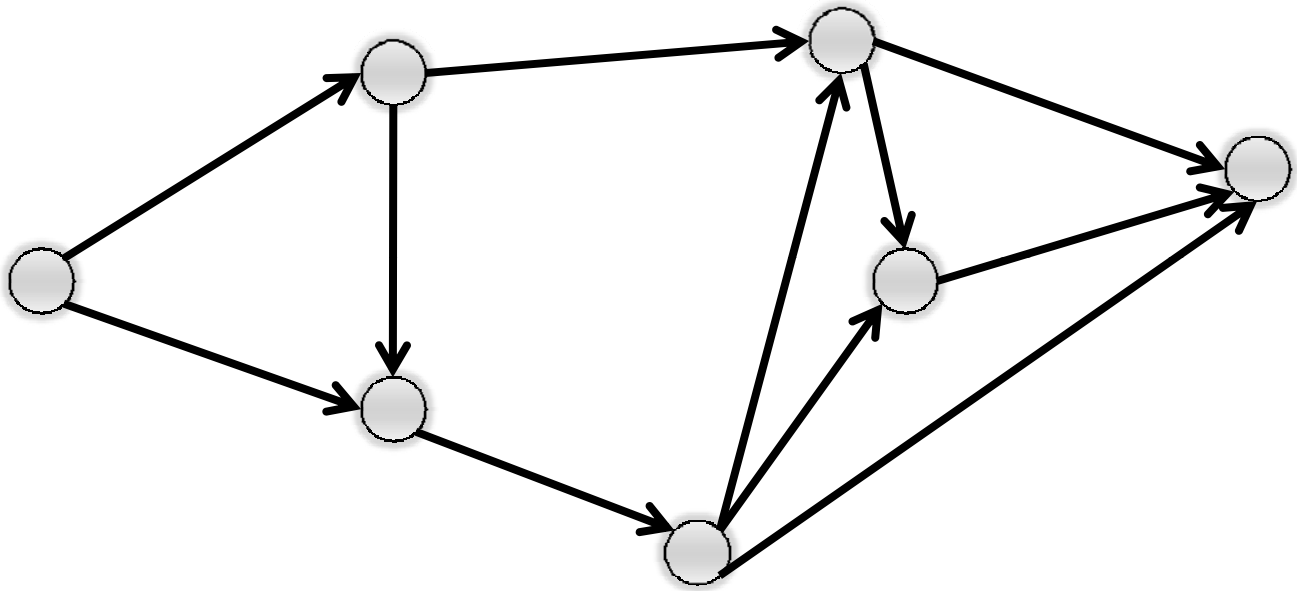
The throughput gain in *undirected* settings is at most 2.

Roadmap

- Motivations, High-level Picture, Goofing Around, …
- Algebraic Foundations of Network Coding
- Decentralized/Randomized Construction
- Practical Considerations

Part II
Algebraic Foundations of Network Coding

Problem Formulation



Setting: Directed Graph with edge capacity $C(e)$

Problem Formulation (cont.)

- Input random processes:

$$\chi(v) = \{X(v,1), X(v,2), \ldots, X(v, \mu(v))\}$$

- A connection from *v* to *v'*:
  - Replicate a subset of random processes of *v*.

$$c = (v, v', \chi(v, v'))$$

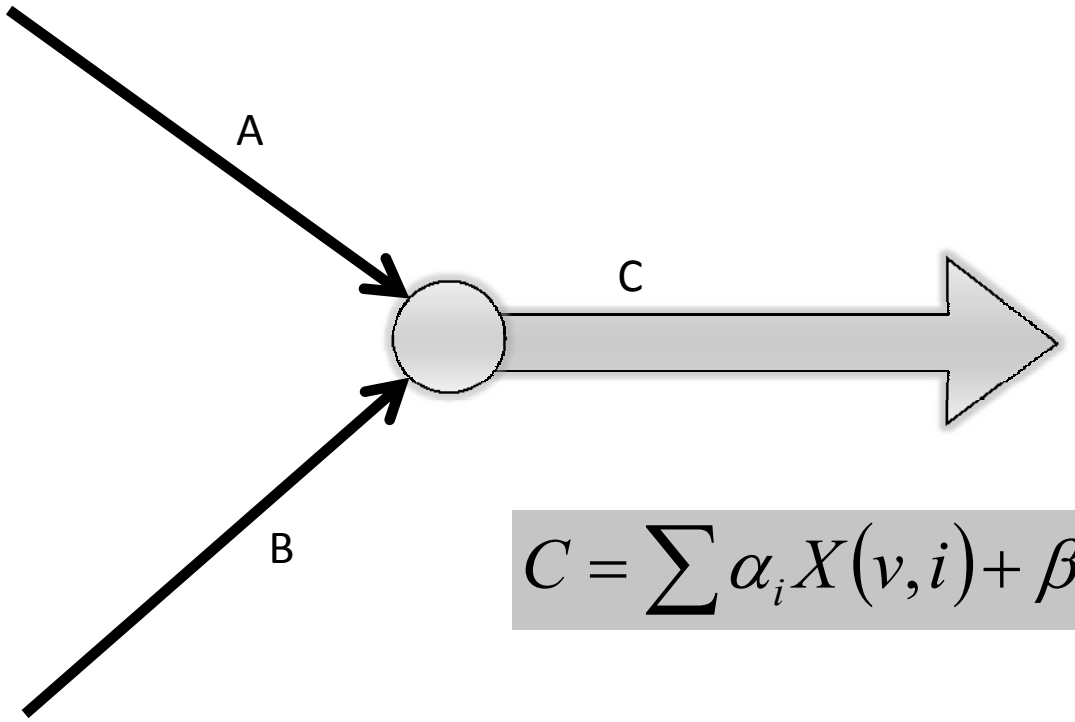
- A pair of graph and set of connections defines a network coding problem.

Basic Model: Linear Network Codes

- Links have the same capacity. I.e., $C(e) = 1$
- Sources have the same rate. I.e., $H(X(v, i)) = 1$
- The "data" $X(v, i)$ are mutually independent (across v and i).

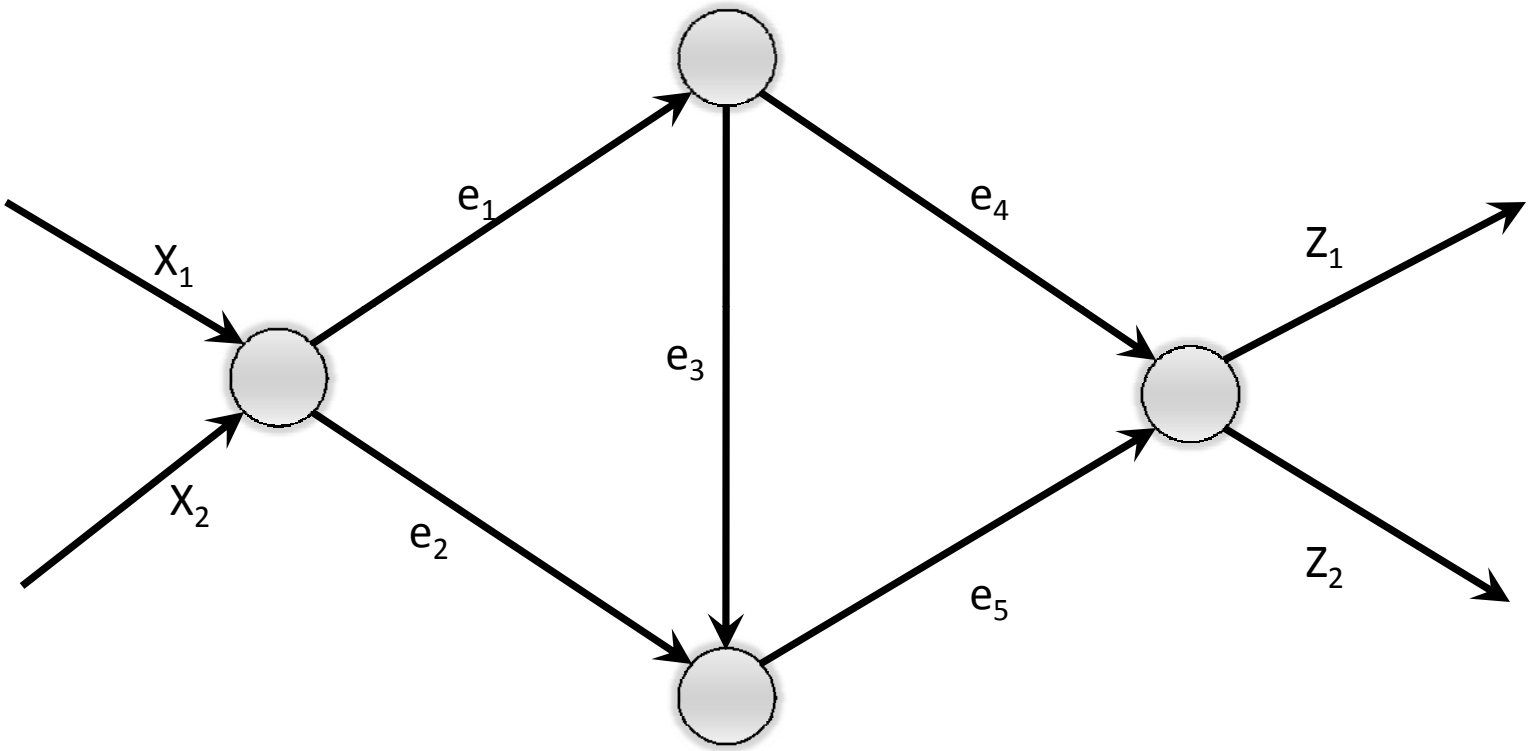
- All operations at network nodes are linear :)

Linear Network Codes



$$C = \sum \alpha_i X(v,i) + \beta_{\mathbf{a}} A + \beta_b B$$

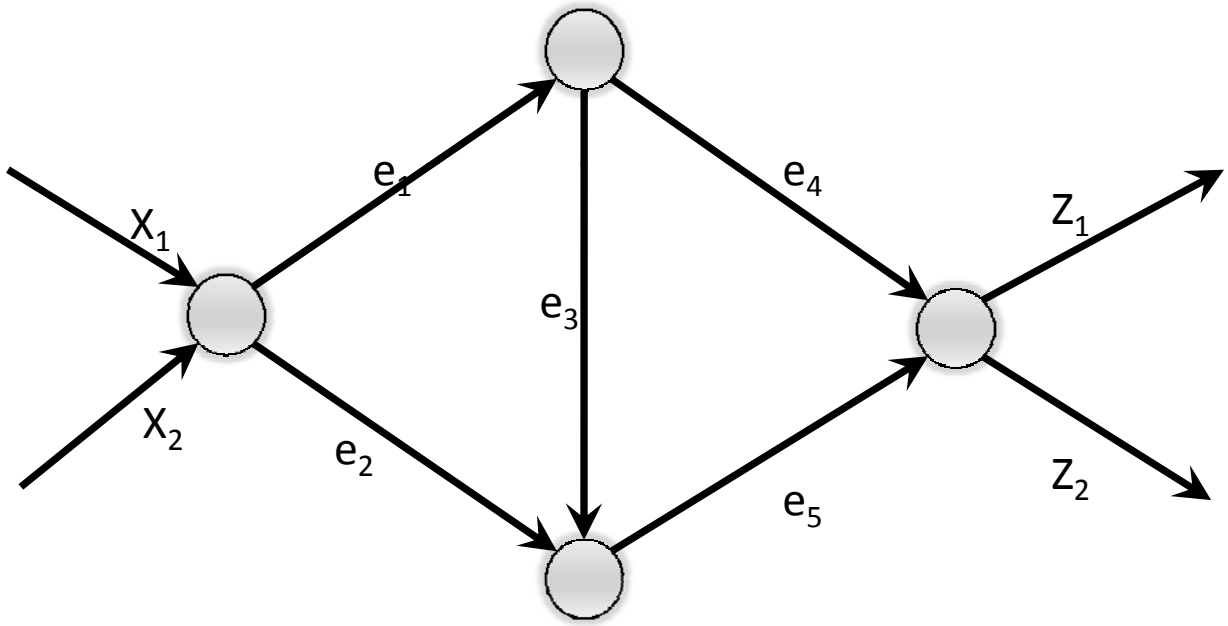
Linear Network: A Simple Example



$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = \begin{pmatrix} \varepsilon_{e_4,1} & \varepsilon_{e_5,1} \\ \varepsilon_{e_4,2} & \varepsilon_{e_5,2} \end{pmatrix} \begin{pmatrix} \beta_{e_1,e_4} & 0 \\ \beta_{e_1,e_3}\beta_{e_3,e_5} & \beta_{e_2,e_5} \end{pmatrix} \begin{pmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} \\ \alpha_{2,e_1} & \alpha_{2,e_2} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

The Transfer Matrix

Let F be an $|E| \times |E|$ matrix, where $f_{i,j} = \beta_{i,j}$.



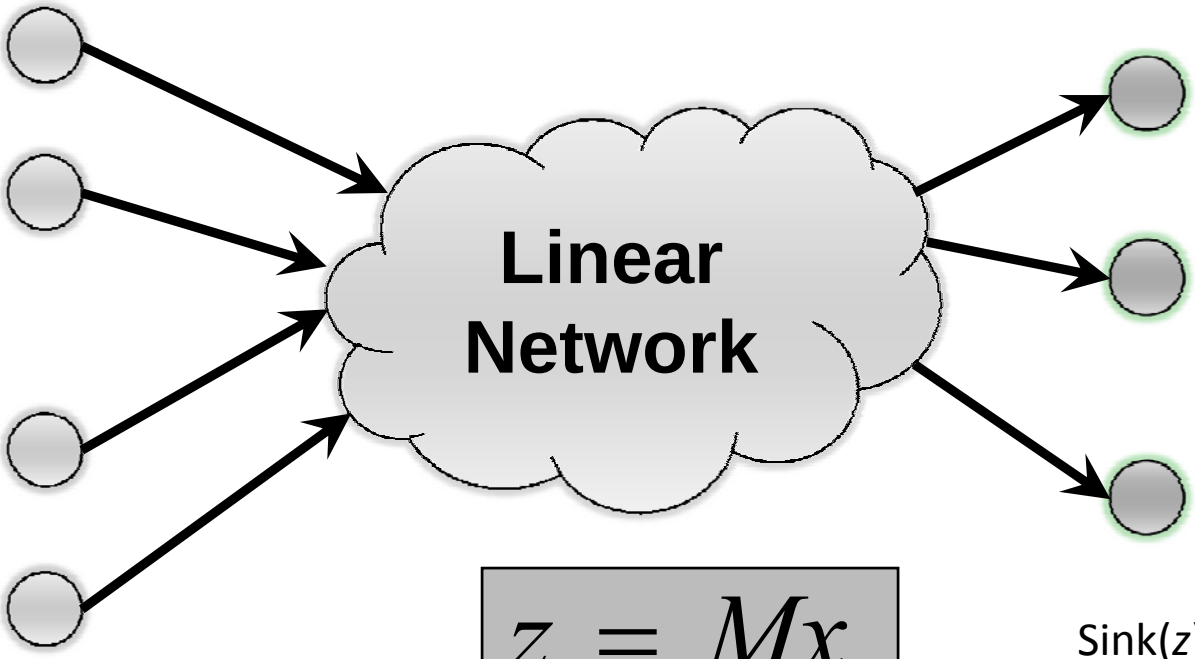
Let $G = I + F + F^2 + \ldots = (I - F)^{-1}$, and define M = BGA

A Linear System

Transfer Matrix

Let F be an $|E|$x$|E|$ matrix, where $f_{i,j} = \beta_{i,j}$.

Let $G = I + F + F^2 + \ldots = (I - F)^{-1}$, and define $M = BGA$



$$z = Mx$$

Source (x)

Sink(z)

Connections to Max-Flow Min-Cut

- **Theorem** (Max-Flow Min-Cut)
  - In a network where the only desired connection is *c*, the network problem is solvable if and only if the rate of the connection *R*(*c*) is less than or equal to the minimum value of all cuts between the source and the sink.

What about Multicast?

- **Theorem**
  - There exists a linear network coding solution for a network problem over a finite field of 2^m elements for some large enough *m* if and only if there exists a flow of sufficient capacity between the source and each sink individually.
  - Together with Max-Flow Min-Cut theorem, gives a criterion for when a network problem is solvable.

So far…

- Networks have no cycles, and no delays.
  - Easy to add delays, cycles: $G = (I - DF)^{-1}$
- We don't know how to construct the matrices *A*, *B*, and *M* yet.
  - That's the topic of our next section.
- How big is the field size (2^m)?
  - Typically not too bad.
  - Multicast: $O(T)$

Roadmap

- Motivations, High-level Picture, Goofing Around, …
- Algebraic Foundations of Network Coding
- Decentralized/Randomized Construction
- Practical Considerations

How do we construct these magic matrices?

Many centralized methods:

- Direct algebraic solution [KM01]
- Subgraph/flow solutions [SET03, JCJ03]
- etc.

Part III
Distributed Randomized Coding

Randomized Coding: Idea
[Ho, Medard, Shi, Effros and Karger, 2003]

- Interior network nodes independently choose random linear mappings from inputs to outputs.
- Coefficients of aggregate effect communicated to receivers.
- => Receivers can decode if they receive as many independent linear combinations as the number of source processes.

Randomized Coding: Main Theorem
[Ho, Medard, Shi, Effros and Karger, 2003]

- For a feasible (multicast) connection problem on a (possibly cyclic) network, a network code constructed by the previously mentioned scheme has a success probability at least $1 - (1 - d/q)^n$ for $q > d$, where d is the number of receivers and n is the number of links carrying random source processes.

- **Proof Idea:** Pretty much follows from the two lemmas we are about to see.

Lemma 1: Network as Edmonds matrix

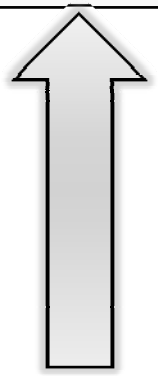
- For an arbitrary (possibly cyclic) network, the transfer matrix $A(I - F)^{-1}B$ is non-singular if and only if the corresponding Edmonds matrix E is non-singular, where

$$E = \begin{bmatrix} A & 0 \\ I - F & B \end{bmatrix}$$

Proof of Lemma 1

- Note that

$$\begin{bmatrix} I & -A(I-F)^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A & 0 \\ I-F & B \end{bmatrix} = \begin{bmatrix} 0 & -A(I-F)^{-1}B \\ I-F & B \end{bmatrix}$$



The matrix has determinant 1.

By a simple expansion, we know that

$$\det \begin{bmatrix} A & 0 \\ I-F & B \end{bmatrix} = c \det\left(A(I-F)^{-1}B\right)\det(I-F)$$

Since det ($I - F$) is non-zero, the lemma follows.

Lemma 2

- Let P be a polynomial in F [Y_1, Y_2,…] of degree at most dn, in which the largest exponent of any variable is at most d. If values for Y's are chosen independently and uniformly at random from $F_q \subseteq F$, then the probability that P equals to zero is at most $1 - (1 - d/q)^n$ for $d < q$.

- **Proof Idea**: Recursive applications of Schwartz-Zippel.

Thm [Schwartz-Zippel]

Let P be a polynomial of degree d over a field F. Let S be a finite subset of F. The probability that P is 0 when evaluated at randomly selected points from S is <= $d/|S|$.

Proof Sketch of Lemma 2

- By Schwartz-Zippel, we know that

$$\Pr[P = 0] \leq \Pr[P_1 \neq 0]\frac{d_1}{q} + \Pr[P_1 = 0]$$

where d_1 is the largest exponent of Y_1 in P,
and P_1 is such that $P = Y_1^{d_1} P_1 + R_1$

- Inductively, we find that

$$\Pr[P_{k'} = 0] \leq \left(1 - \frac{d_{k'+1}}{q}\right)\Pr[P_{k'+1} \neq 0] + \frac{d_{k'+1}}{q}$$

Proof Sketch of Lemma 2 (cont.)

- Combining the results:

$$\Pr[P=0] \le \frac{\sum_{i=1}^{k} d_i}{q} - \frac{\sum_{i \neq j} d_i d_j}{q^2} + \ldots + (-1)^{k-1} \frac{\prod_{i=1}^{k} d_i}{q^k}$$

where $0 \le dn - \sum_{i=1}^{k} d_i$

- Seek d_i's that maximize the probability.
  - Idea: Relax an integer optimization program (straightforward but tedious)
  - Result: we know $d_i^* \in \{0, d\}$ and $\sum_{i=1}^{dn} d_i^* = dn$
- Q.E.D.

Part IV
Practical Considerations

Okay, I want to implement network coding today…

- What do you say?
  - Good luck (with an evil smile)
  - It is actually pretty simple…

Linear Coding Summary

- Original Packets: $M^{(1)}, M^{(2)}, \ldots, M^{(n)}$
- Each round, randomly pick $g_1, g_2, \ldots, g_n$ and send out

$$\left(\vec{g}, X\right) = \left(\left\langle g_1, \ldots, g_n \right\rangle, \sum_{i=1}^{n} g_i M^{(i)}\right)$$

- Intermediate nodes dream up $h_1, h_2, \ldots, h_m$ used for combining packets.

$$X' = \sum_{i=1}^{m} h_j X^j$$

Linear Coding Summary (cont.)

- Decoding
  - Assume you receive $\left\{\left(g^i, X^i\right)\right\}_{i=1}^{m}$
  - Basically solve the linear system

$$X^j = \sum_{i=1}^{n} g_i^j M^{(i)}$$

  - Experiments show that Gaussian elimination seem to suffice.

Real-World Use of Network Coding

P2P Content Distribution

Many-to-Many Broadcast

Data Gathering in
Ad-hoc Sensor Networks

Avalanche: A P2P Distribution

- Microsoft Research
- Goal: Efficient content distribution network
  - Get large files (e.g., Windows update) to everyone
- Network Coding:
  - Minimizes download times (optimal packet scheduling is almost impossible; complex network)
  - Outperforms traditional forwarding
  - More robust when servers leave early (DoS/bombarding)
  - Works okay when incentive mechanisms are implemented.


Thank You