Linear Network Coding: Introduction and Application

Presented by 唐崇實

2/16/2006

Outline

- Introduction to Linear Network Coding
- Linear Information Flow (LIF) Algorithm
- Application of Network Coding
- Overlay Network Monitoring
- Summary and Discussion

Linear Network Coding (1/2)

- Encoding
  - Original packets $M^1, \ldots, M^n$ generated by one or several sources
  - Each packet contains *encoding vector* $g = (g_1, \ldots, g_n)$ in $\mathbf{F}_{2^s}$ and *information vector* $X = \sum_{i=1}^{n} g_i M^i$
  - The summation has to occur for every symbol position, i.e., $X_k = \sum_{i=1}^{n} g_i M_k^i$, M_k^i and X_k is the kth symbol of M^i and X
  - The encoding vector is used by recipients to decode the data, ex: $e_i = (0,..,0,1,0,..,0)$ means M^i
  - Encoding can be performed recursively to already encoded packets

Linear Network Coding (2/2)

- Decoding
  - A node has received the set $(g^1, X^1), \ldots, (g^m, X^m)$
  - In order to retrieve the original packets, it needs to solve the system $\{X^j = \sum_{i=1}^{n} g^j_i M^i\}$ — linear systems with m equations and n unknowns, where the unknowns are M^i
  - $m \geq n$ is needed to have a chance of recovering all data

Network Code Design

- The problem of network code design is to select what linear combinations each node performs
  - Simple algorithm: each node select uniformly at random the coefficients over the field $\mathbf{F}_{2^s}$, in a completely independent and decentralized manner → the probability of failing to decode at each destination node is 1/ |F|
  - Polynomial-time algorithm for multicasting: using the Linear Information Flow (LIF) algorithm

Polynomial Time Coding

- The algorithm is for centralized design of optimal network multicast codes
- The algorithm consists of two stages
  - A flow algorithm to find, for each sink $t \in T$, a set f^t of h edge-disjoint paths from s to t
  - A greedy algorithm that visits each edge in turn and designs the linear coding employed for that edge → the goal in designing the encoding for $e=(v, w)$ is to choose a linear combination of the inputs to node v that ensures all downstream sinks obtain h linearly independent combinations of the original source symbols $b_1, \ldots, b_h$

Linear Information Flow Algorithm

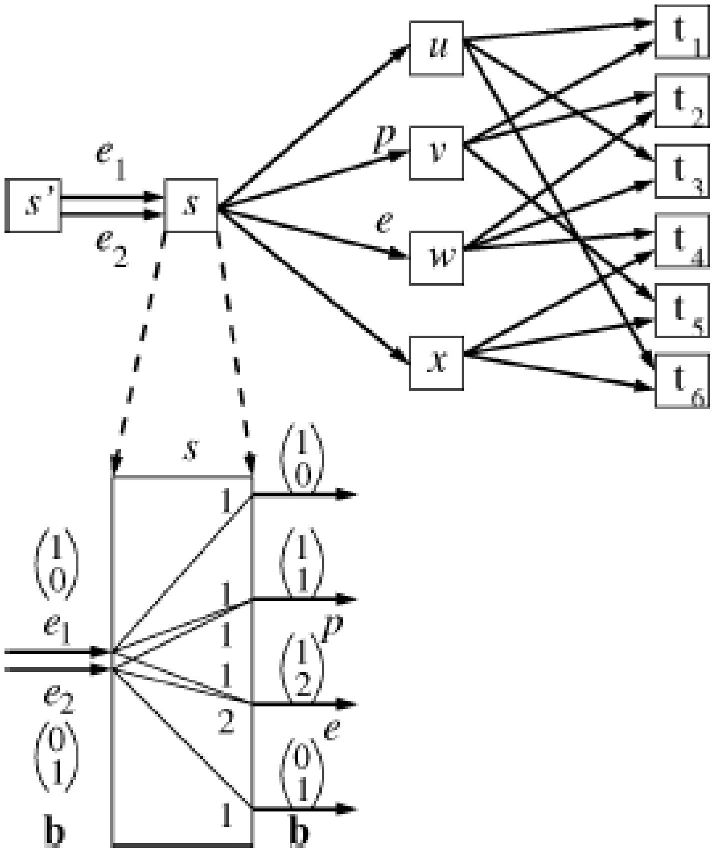
- Notation
  - An acyclic, unit capacity network G=(V,E)
  - $s \in V$ is the source node; T $\subseteq$ V is the set of sink nodes
  - h is the size of smallest min-cut separating s from any $t \in T$
  - $\Gamma_I(v)$ and $\Gamma_O(v)$ denotes the set of edges feeding into and leaving node v, respectively
  - *T(e)* denotes the set of sinks using in some flow f^t
  - $P(e) = \{f^t_{\leftarrow}(e) : t \in T(e)\}$ denotes the set of predecessor edges
- Define a *local coding vector* $\mathbf{m}_e$ for each edge e, the symbol carried by edge e is $y(e) = \sum_{p \in \Gamma_I(\text{start}(e))} m_e(p) y(p)$
- Objective: to determine the coefficients $\mathbf{m}_e(p)$ such that all sinks can reconstruct the original information

Multicasting with Linear Coding

- Multicast example from s to $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$
- $\boldsymbol{b}$: global coding vector

$$\boldsymbol{b}(e) = \sum_{g \in P(e)} m_e(g)\boldsymbol{b}(g)$$

- Assume $\mathbf{F}$=GF(3)
- $\Gamma_I(t_2) = \{(v, t_2); (w, t_2)\}$
- start(e) = s, P(e)=$\{e_1, e_2\}$
- $T(e) = \{t_2, t_3, t_4\}$
- $f^{t_4}_{\leftarrow}(e) = e_1$ $f^{t_3}_{\leftarrow}(e) = e_2$



LIF with Linear Independence Testing

Function LIF(V, E, s, T)

$h := \min_{t \in T} \min\{|C| : C \text{ is } s\text{-}t \text{ cut}\}$ — $= \min_{t\in T} |\text{max flow from } s \text{ to } t|$

find max flow

insert a new source s' into V — help to establish the invariant

insert h parallel edges $\{e_1, \ldots, e_h\}$ from s' to s into E

add artificial edges

let f^t denote a set of h edge disjoint paths from s to t — the chosen flow from s to t

(* We use the notation $f^t_{\leftarrow}(e)$, $T(e)$, and $P(e)$ to access the flows. *)

let $\mathbb{F}$ be a finite field of a size satisfying the conditions of Theorem 3

define s→t path

forall i: $\mathbf{b}(e_i) := [0^{i-1}, 1, 0^{h-i}]$ — the i-th unit vector of $\mathbb{F}^h$

forall $t \in T$ **do**

- $C_t := \{e_1, \ldots, e_h\}$ — t is supplied through C_t
- $B_t := \{\mathbf{b}(e_1), \ldots, \mathbf{b}(e_h)\}$ — the coding vectors span $\mathbb{F}^h$
- **forall** $c \in C_t$: $\mathbf{a}_t(c) := \mathbf{b}(c)$ — inverse vectors

set initial vectors that span $\mathbf{F^h}$

foreach vertex $v \in V \setminus \{s'\}$ in topological order **do**

- **forall** outgoing edges e of v **do**
  - (* Invariant: $\forall t \in T : |C_t| = h$ and $\forall c, c' \in C_t : \mathbf{b}(c) \cdot \mathbf{a}_t(c') = \delta_{c,c'}$ *)
  - choose a linear combination $\mathbf{b}(e) = \sum_{p \in P(e)} m_e(p)\mathbf{b}(p)$ such that — (*)
    - $\forall t \in T(e) : (B_t \setminus \{\mathbf{b}(f^t_{\leftarrow}(e))\}) \cup \{\mathbf{b}(e)\}$ is linearly independent

find resulting coding vector **b(e)**, including testing for linear independence

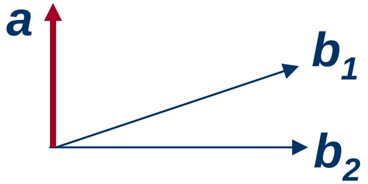
  - **forall** $t \in T(e)$ **do**
    - $C'_t := (C_t \setminus \{f^t_{\leftarrow}(e)\}) \cup \{e\}$ — advance the set of edges C_t,
    - $B'_t := (B_t \setminus \{\mathbf{b}(f^t_{\leftarrow}(e))\}) \cup \{\mathbf{b}(e)\}$ — update B_t correspondingly, and
    - $\mathbf{a}'_t(e) := (\mathbf{b}(e) \cdot \mathbf{a}_t(f^t_{\leftarrow}(e)))^{-1}\, \mathbf{a}_t(f^t_{\leftarrow}(e))$ — update $\mathbf{a}_t$ correspondingly
    - **forall** $c \in C_t \setminus \{f^t_{\leftarrow}(e)\}$: $\mathbf{a}'_t(c) := \mathbf{a}_t(c) - (\mathbf{b}(e) \cdot \mathbf{a}_t(c))\mathbf{a}'_t(e)$
    - $(C_t, B_t, \mathbf{a}_t) := (C'_t, B'_t, \mathbf{a}'_t)$

form new set of spanning vector

return $(h, \{m_e : e \in E\}, \{(C_t, \mathbf{a}_t) : t \in T\}, \mathbb{F})$.

Testing for Linear Independence

- Idea: testing whether a vector is linearly dependent on an h-1 dimensional subspace can be done by testing the dot-product of the vector with the vector representing the orthogonal complement of the subspace.
- Maintain the invariant that for each sink $t \in T$ there is a set C_t of h edges such that the set of global coding vectors $B_t = \{\boldsymbol{b}(c): c \in C_t\}$ forms a basis of $\mathbf{F}^h$
- Maintain vectors $\boldsymbol{a}_t(c)$ for each sink t and edge $c \in C_t$ that can be used to test linear dependence on $B_t \setminus \{\boldsymbol{b}(c)\}$



Application of Network Coding

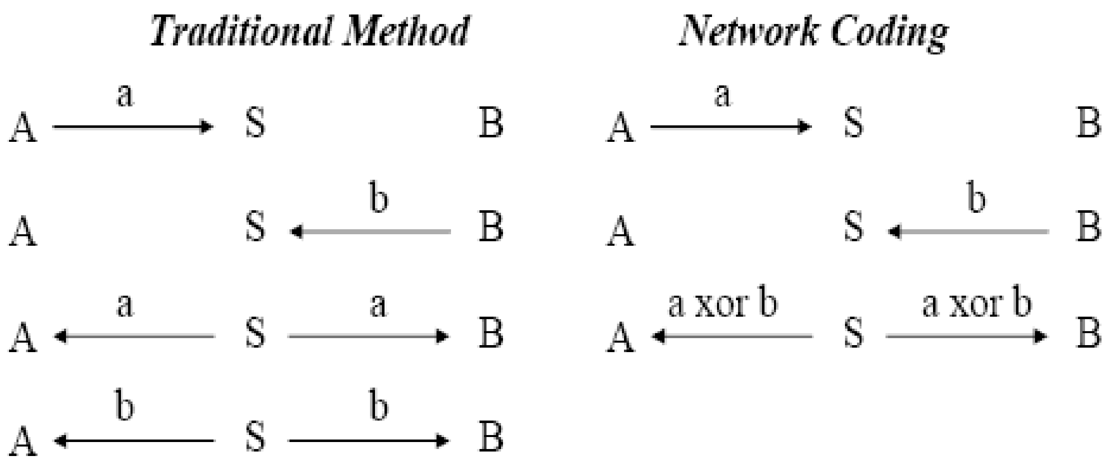
- P2P File Distribution
- Wireless Networks
- Ad-hoc Sensor Networks
- Network Tomography
- Network Security

P2P File Distribution

- Avalanche
  - A server splits a large file into a number of blocks
  - The blocks sent out by the server and peers are random linear combinations of all original blocks
  - A node can either
    - determine how many innovative blocks it can transmit to a neighbor by comparing its own and the neighbor's matrix of decoding coefficients, or
    - simply transmit coded block until the neighbor receives the first non-innovative block
- Network coding helps in
  - Minimizing download times
  - More robust in early-leaving server or high churn rate
  - Small performance penalty under incentive mechanisms

Wireless Networks

- Network coding can improve throughput when two wireless nodes communicate via a common base station



- Can be extended to the case of Multi-hop routing in a wireless network (or any other network with physical layer broadcast) where
  - The traffic between two end nodes is bidirectional, and
  - Both nodes have a similar number of packets to exchange

Ad-hoc Sensor Networks

- Untuned radios in sensor networks
  - To replace the analog oscillator by a much simpler on-chip resonator → radio frequencies depend on manufacturing
  - In dense sensor networks, a multi-hop path between source and sink will most probably exist
  - With random network coding, it's possible to use these paths without having to "explicitly find them" and without excessive overhead of flooding
- Data gathering in sensor networks
  - Nodes have storage for one single packet
  - Overheard packets from neighboring nodes are multiplied with a random coefficient and added to the existing one
  - A sink can reconstruct n data packets with a high probability by contacting only n sensor nodes

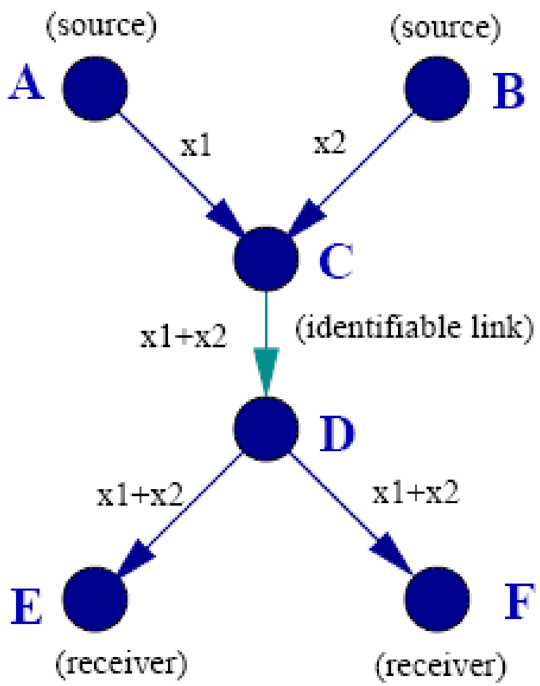
Network Tomography

- Active monitoring
  - Conventional active probing, packets are usually multicast to several receivers.
  - The receivers experience the same loss event in the underlying multicast tree
  - Using network coding to infer the loss rates of links in an overlay network
- Passive network monitoring
  - A random (but fixed) network code allows receivers to determine which coefficients are expected under normal condition
  - When the obtained coefficients differ, the receiver can draw out the failure pattern

Network Security

- Secure network codes for wiretap networks
  - The source combines the original data with random information and designs a network code in a way that only the receivers can decode
- Weak security
  - With network coding, nodes can only decode packets if they have received a sufficient number of linearly independent information vectors
- Protection against modified packets
  - In the case of network coding, an attacker can’t control the outcome of decoding process at the destination, without knowing all other coded packets the destination will receive

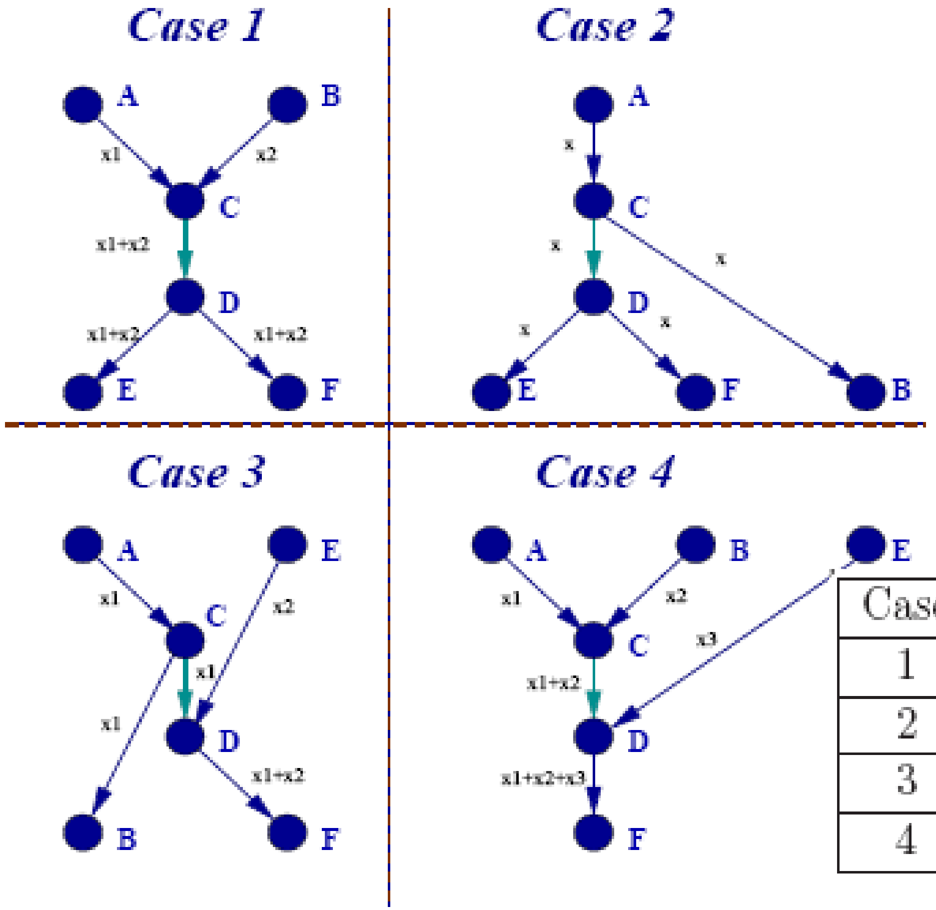
Network Monitoring Example



$x_3 = x_1 \oplus x_2$

Received at		Is link ok?				
E	F	AC	BC	CD	DE	DF
0	0	Multiple possible events				
x_1	–	1	0	1	1	0
x_2	–	0	1	1	1	0
x_3	–	1	1	1	1	0
–	x_1	1	0	1	0	1
x_1	x_1	1	0	1	1	1
–	x_2	0	1	1	0	1
x_2	x_2	0	1	1	1	1
–	x_3	1	1	1	0	1
x_3	x_3	1	1	1	1	1

Network Coding Improve Identifiability



Identifiable links
for the four cases

Case	Network Coding	Multicast Probes
1	all links	DE, DF
2	all links	all links
3	all links	AC, CB
4	all links	no links

Summary and Discussion

- Network coding is an efficient and effective technique in many applications
- Two categories of research on network coding
  - To propose more efficient coding and decoding algorithms
  - To apply network coding technique on specific network fields of interest

References

[1] Christina Fragouli, et. al. “Network Coding: An Instant Primer,” LCA-REPORT-2005-010.

[2] Peter Sanders, et. al., “Polynomial Time Algorithms for Network Information Flow,” ACM Symposium on Parallel Algorithms and Architectures, 2003.

[3] Sidharth Jaggi, et. al., “Polynomial Time Algorithms for Multicast Network Code Construction,” IEEE Trans. on Information Theory, June 2005.

[4] Christina Fragouli and Athina Markopoulou, “A Network Coding Approach to Overlay Network Monitoring,” In Allerton Conference, Sept. 2005.

[5] Christos Gkantsidis and Pablo Rodriguez Rodriguez, “Network Coding for Large Scale Content Distribution,” IEEE Infocom 2005