Network coding for security and robustness

Outline

- Network coding for detecting attacks
- Network management requirements for robustness
- Centralized versus distributed network management

Byzantine security

- Robustness against faulty/malicious components with arbitrary behavior, e.g.
  - dropping packets
  - misdirecting packets
  - sending spurious information
- Abstraction as Byzantine generals problem [LSP82]
- Byzantine robustness in networking [P88,MR97,KMM98,CL99]

Byzantine detection with network coding [HLKMEK04]

Distributed randomized network coding can be extended to detect Byzantine behavior

- Small computational and communication overhead
  - small number of hash bits included with each packet, calculated as simple polynomial function of data
- Require only that a Byzantine attacker does not design and supply modified packets with complete knowledge of other nodes' packets

Byzantine modification detection scheme

- Suppose each packet contains θ data symbols $x_1, \ldots, x_\theta$ and $\phi \leq \theta$ hash symbols $y_1, \ldots, y_\phi$

- Consider the function $\pi(x_1, \ldots, x_k) = x_1^2 + \cdots + x_k^{k+1}$

- Set

$$\begin{aligned} y_i &= \pi(x_{(i-1)k+1}, \ldots, x_{ik}) \quad \text{for} \quad i = 1, \ldots, \phi - 1 \\ y_\phi &= \pi(x_{(\phi-1)k+1}, \ldots, x_\theta) \end{aligned}$$

where $k = \left\lceil \frac{\theta}{\phi} \right\rceil$ is a design parameter trading off overhead against detection probability

Detection probability

[HLKMEK04] If the receiver gets s genuine packets, then the detection probability is at least $1 - \left(\frac{k+1}{q}\right)^s$.

- E.g. With 2% overhead ($k = 50$), code length=7, $s = 5$, the detection probability is 98.9%.

- with 1% overhead ($k = 100$), code length=8, $s = 5$, the detection probability is 99.0%.

Analysis

- Let M be the matrix whose i^{th} row $\underline{m_i}$ represents the concatenation of the data and corresponding hash value for packet i

- Suppose the receiver tries to decode using
  - s unmodified packets, represented as $C_a\,[M|I]$, where the i^{th} row of the coefficient matrix C_a is the vector of code coefficients of the i^{th} packet
  - $r-s$ modified packets, represented by $[C_bM + V|C_b]$, where V is an arbitrary matrix

Analysis (cont'd)

- Let $C = \left[\frac{C_a}{C_b}\right]$

- Decoding is equivalent to pre-multiplying the matrix

$$\left[\begin{array}{c|c} C_a M & C_a \\ \hline C_b M + V & C_b \end{array}\right]$$

  with C^{-1}, which gives

$$\left[\begin{array}{c|c} M + C^{-1}\left[\frac{0}{V}\right] & I \end{array}\right]$$

- For any C_b and V, since receiver decodes only with a full rank set of packets, possible values of C_a are s.t. C is non-singular

Analysis (cont'd)

We can show that

- for each of $\geq s$ packets, the attacker knows only that the decoded value will be one of $q^{\text{rank}(V)}$ possibilities

$$\left\{ \underline{m}_i + \sum_{j=1}^{\text{rank}(V)} \gamma_{i,j} \underline{v}_j \;\middle|\; \gamma_{i,j} \in \mathbb{F}_q \right\}$$

- at most $k+1$ out of the q vectors in a set $\{\underline{u} + \gamma \underline{v} | \gamma \in \mathbb{F}_q\}$, where $\underline{u} = (u_1, \ldots, u_{k+1})$ is a fixed length-$(k+1)$ vector and $\underline{v} = (v_1, \ldots, v_{k+1})$ a fixed nonzero length-$(k+1)$ vector, can satisfy the property that the last element of the vector equals the hash of the first k elements.

Network mgt for link failure recovery [HMK02, HMK03]

- Structured schemes for link failure recovery, e.g. end-to-end path protection, loopback, generalized loopback

- Network coding admits any solution feasible on surviving links

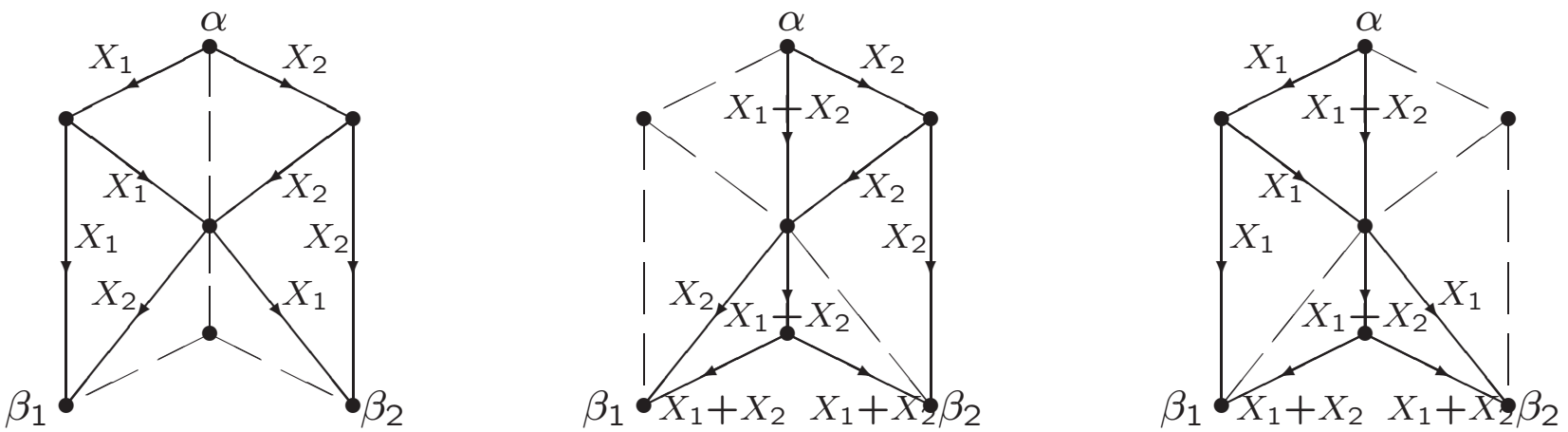
- Network management information directs network's response to different link failures

- Questions:
  -How to quantify fundamental amount of information needed

to direct recovery?
-How do different types of recovery schemes compare in management overhead?

A theoretical framework for network management

- Network management information can be quantified by the log of the number of different behaviors (codes) used [tbh]



- Allowing general network coding solutions gives fundamental limits on management information required

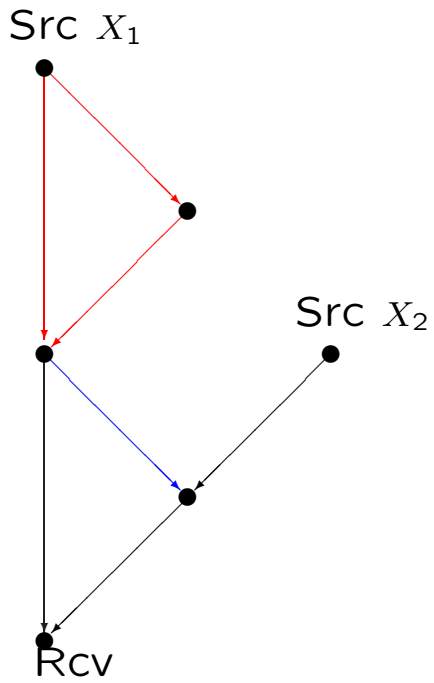
Classes of failure recovery schemes considered

- Receiver-based schemes: only receivers change behavior under different failure scenarios
- Network-wide schemes: any node may change behavior, includes receiver based schemes as a special case
- Linear schemes: linear operations at all nodes
- Nonlinear receiver-based schemes: nonlinear decoding at receivers

Need for network management

- A link h is called *integral* if there exists some subgraph of the network on which the set of source-receiver connections is feasible if and only if h has not failed.

- For any network connection problem with at least one integral link whose failure is recoverable, no single linear code can cover the no-failure scenario and all recoverable failures


Src X_1
Src X_2
Rcv

Bounds on network management

Network management for single recoverable link, using network parameters

- r, number of source processes transmitted in network;
- m, the number of links in a minimum cut between the source nodes and receiver nodes;
- d, the number of receiver nodes;
- t_{min}, the minimum number of terminal links among all receivers.

Some bounds

- Tight lower bounds on no. of linear codes for general case:

receiver-based	$\left\lceil \frac{m}{m-r} \right\rceil$
network-wide	$\left\lceil \frac{m+1}{m-r+1} \right\rceil$

- Tight upper bounds on no. of linear codes for the single-receiver:

receiver-based	$\begin{cases} r+1 & \text{for } r = 1 \text{ or } m-1 \\ r & \text{for } 2 \le r \le m-2 \end{cases}$
network-wide	$\begin{cases} r+1 & \text{for } r = 1,\ r = 2 = m-1 \\ r & \text{for } r = 2 \le m-2, \\ & \quad r = 3,\ r = m-1 \ge 3 \\ r-1 & \text{for } 4 \le r \le m-2 \end{cases}$

- Upper bound on no. of linear codes for multicast: $(r^2+2)(r+1)^{d-2}$

- Tight lower bounds for nonlinear receiver-based codes for multicast:

$$\begin{cases} r & \text{for } 1 < r = t_{\min} - 1 \\ 1 & \text{for } r = 1 \text{ or } r \leq t_{\min} - 2 \end{cases}$$