Delays and cycles in networks

- Delay-free network coding implicitly assumes that coding occurs over an acyclic topology
- For networks with cycles, we can
  - do convolutional network coding, or
  - code over a time-expanded network, as in packet networks
    - source process <-> source packet
    - link <-> transmitted packet

Centralized multicast code construction

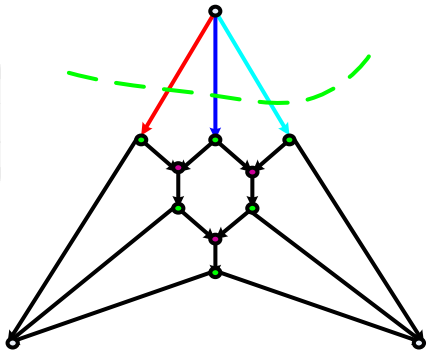
- Centralized polynomial-time construction for acyclic graphs
  - Choose a flow solution for each sink individually
  - Consider the links in the union of these flow solutions
  - Set the code coefficients of these links in topological order starting from the source, ensuring that at each step the "frontier set" for each sink has linearly independent coefficient vectors
  - Solution follows by induction

S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction,"IEEE Transactions on Information Theory. June 2005 .

Example

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_1}



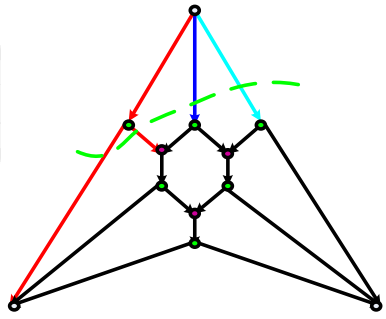
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_2}

Example

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_1}



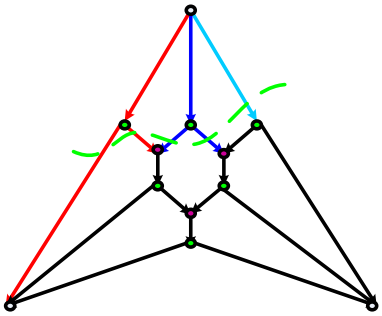
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_2}

Example

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_1}



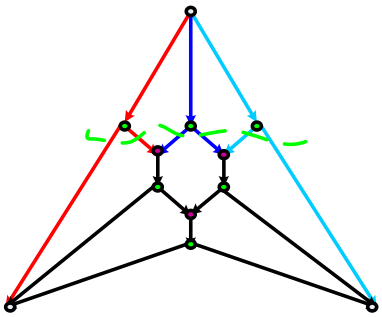
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_2}

Example

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_1}



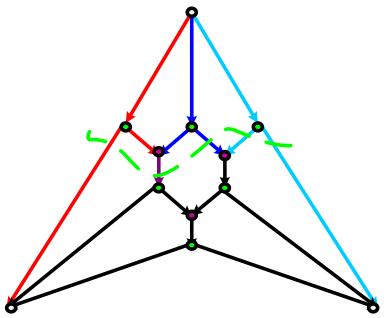
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_2}

Example

$$\begin{pmatrix} 1 & ? & 0 \\ 0 & ? & 0 \\ 0 & ? & 1 \end{pmatrix}$$

B_{t_1}



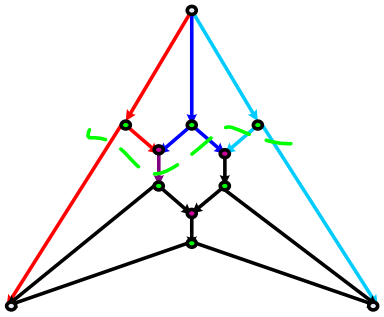
$$\begin{pmatrix} ? & 0 & 0 \\ ? & 1 & 0 \\ ? & 0 & 1 \end{pmatrix}$$

B_{t_2}

Example

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_1}



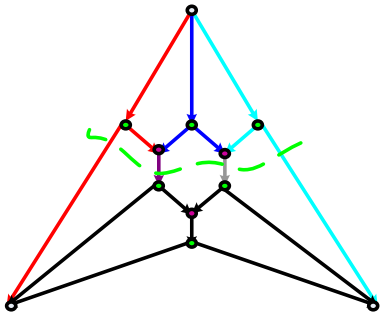
$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_2}

Example

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

B_{t_1}

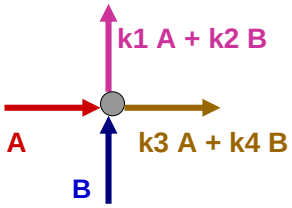


$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

t2 matrix

Distributed random network coding

- Random linear coding among packets of a single multicast or unicast session
- Nodes independently choose random linear mappings from inputs to outputs in some field
- To communicate transfer matrix to receivers, packet header contains a vector of code coefficients called the *global coding vector,* to which same linear mappings are applied



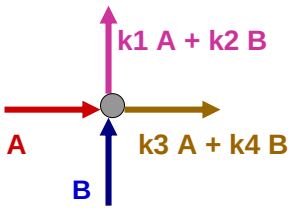
1	0	A
0	1	B

k1	k2	k1 A + k2 B
k3	k4	k3 A + k4 B

T. Ho, R. Koetter, M. Médard, D. R. Karger and M. Effros, ``The Benefits of Coding over Routing in a Randomized Setting'', International Symposium on Information Theory 2003.

Distributed random network coding

- Each sink can decode when it has received a full rank set of packets
- For any multicast subgraph which satisfies min-cut max-flow bound for each receiver, probability of failure over field F is decreases exponentially in symbol length log|F|
- Random network coding can be used to distribute a group of packets from any number of sources
- Advantages: decentralized, optimal throughput, robust to link failures / packet losses



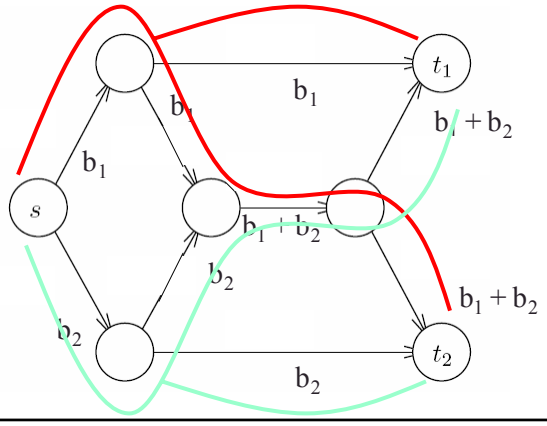
1 0	A
0 1	B

k1 k2	k1 A + k2 B
k3 k4	k3 A + k4 B

T. Ho, R. Koetter, M. Médard, D. R. Karger and M. Effros, ``The Benefits of Coding over Routing in a Randomized Setting", International Symposium on Information Theory 2003.

Multicast optimization

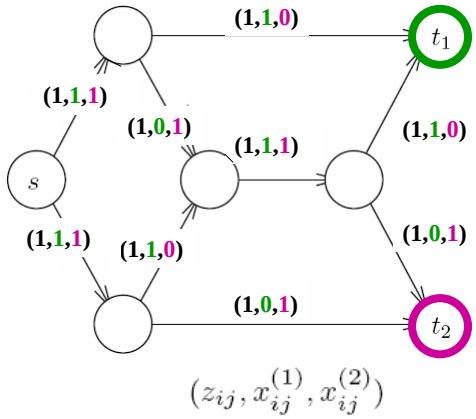
•Without network coding, maximum rate or minimum cost multicast optimization is NP-complete (involves Steiner trees)
•E.g., in the illustration, integrality constraint arises in time-sharing between trees



Minimum cost multicast optimization with network coding

- With network coding, minimum cost multicast optimization problem becomes a polynomial-complexity linear optimization

$$\begin{aligned}
&\text{minimize } f(z) \\
&\text{subject to } z \in Z \\
&z_{ij} \geq x_{ij}^{(t)} \geq 0, \\
&\sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} \\
&\quad = \begin{cases} R, & \text{if } i = s \\ -R, & \text{if } i = t \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$



D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *IEEE Trans. Inform. Theory*, 52(6):2608-2623, June 2006.

Minimum cost multicast optimization

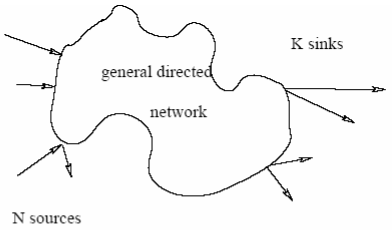
- The vector z is part of a feasible solution for the optimization problem if and only if there exists a network code that sets up a multicast connection in the graph G at *average* rate arbitrarily close to R from source s to terminals in the set T and that puts a flow arbitrarily close to z_{ij} on each link (i, j)
  - Proof follows from min-cut max-flow necessary and sufficient conditions
- For any convex cost functions, optimization can be solved in polynomial-time

D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *IEEE Trans. Inform. Theory*, 52(6):2608-2623, June 2006.

Multicast optimization with network coding

- Network coding multicast optimization problems can be solved distributedly using various optimization algorithms [Lun et al 05, Wu et al. 06, etc.]
- For multiple multicast sessions with intra-session network coding, a distributed back pressure approach can be used [Ho & Viswanathan 05]
  - Generalization of back pressure approach for multi-commodity routing of Tassiulas & Ephremides 92, Awerbuch & Leighton 93
  - Network coding greatly reduces complexity for multicast

Scalar linear network coding for non-multicast



$$\mathscr{C} = \{(v_i, u_j, \mathscr{X}(v_i, u_j))\}$$

$$M = \begin{pmatrix} M_{1,1} & M_{1,2} & \dots & M_{1,K} \\ M_{2,1} & M_{2,2} & & M_{2,K} \\ \vdots & & & \vdots \\ M_{N,1} & M_{N,2} & \dots & M_{N,K} \end{pmatrix}$$

$M_{i,j}$ corresponds to $c_{i,j} = (v_i, u_j, \mathscr{X}(v_i, u_j))$.

Scalar linear network coding for non-multicast

Thm: There exists a scalar linear network coding solution if and only if there exist values for the coding coefficients such that

1. $M_{i,j} = 0$ for all pairs (v_i, v_j) of vertices such that $(v_i, v_j, \mathscr{X}(v_i, v_j)) \notin \mathscr{C}$.

2. If $\mathscr{C}$ contains the connections $(v_{i_1}, v_j, \mathscr{X}(v_{i_1}, v_j)), (v_{i_2}, v_j, \mathscr{X}(v_{i_2}, v_j)), \ldots, (v_{i_\ell}, v_j, \mathscr{X}(v_{i_\ell}, v_j))$ the determinant of $\left[M^T_{i_1,j} M^T_{i_2,j}, \ldots, M^T_{i_\ell,j}\right]$ is nonzero.

R. Koetter and M. Medard, "An algebraic approach to network coding". *IEEE Transactions on Networking,* October 2003

Network coding for non-multicast

For the general case we need to find solutions to some system of polynomial equations!

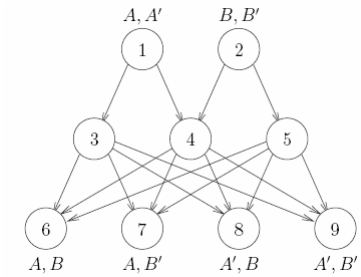
For the multicast case we need to find non solutions to some system of polynomial equations!

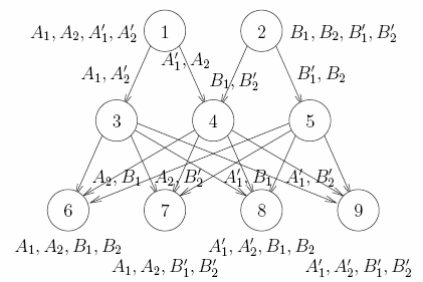
Another way to phrase this is: In a multicast setup everybody wants everything so the issue of interference is moot!

For the general case we may have carefully balanced solutions where some unwanted information cancels out in clever ways.....

Scalar linear coding may not suffice

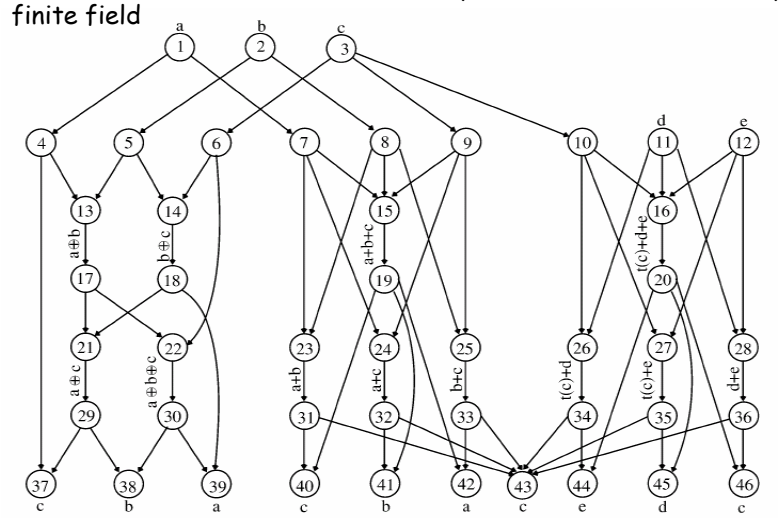
Network requiring a vector (specifically, time-sharing) solution





Linear coding may not suffice

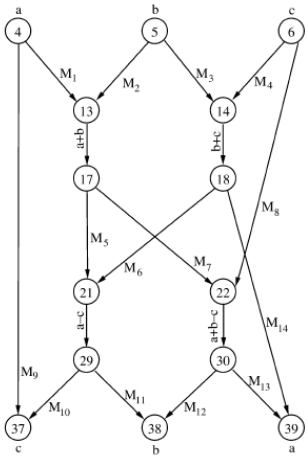
Network with no linear solution for any vector dimension over any finite field



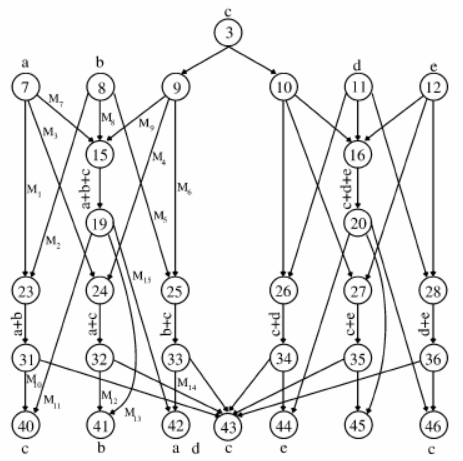
Dougherty, R. Freiling, C. Zeger, K., "Insufficiency of linear coding in network information flow", IEEE Transactions on Information Theory, Aug. 2005

Linear coding may not suffice

No linear solution for any vector dimension over a finite field with odd characteristic



No linear solution for any vector dimension over a finite field with characteristic 2



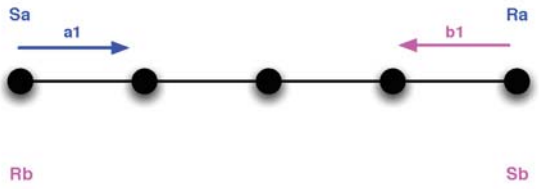
Dougherty, R. Freiling, C. Zeger, K., "Insufficiency of linear coding in network information flow", IEEE Transactions on Information Theory, Aug. 2005

Questions

- Given a non-multicast problem what rates are achievable?
  - In general, we only have inner and outer bounding techniques, and an implicit information theoretic characterization of the capacity region for acyclic networks
- When is coding advantageous in terms of throughput or cost and by how much?
- What types of codes are needed?
- How do we construct such codes?

Coding advantages

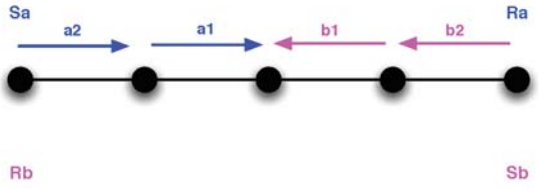
- information exchange between two nodes



Y. Wu, P. A. Chou, S.-Y. Kung, ``Information exchange in wireless networks with network coding and physical-layer broadcast,'' Microsoft Technical Report, MSR-TR-2004-78, Aug. 2004

Coding advantages

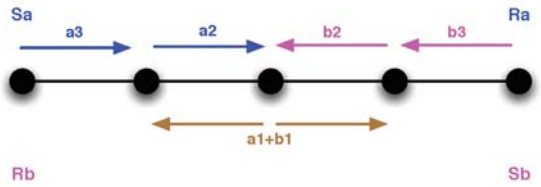
- information exchange between two nodes



Y. Wu, P. A. Chou, S.-Y. Kung, ``Information exchange in wireless networks with network coding and physical-layer broadcast,'' Microsoft Technical Report, MSR-TR-2004-78, Aug. 2004

Coding advantages

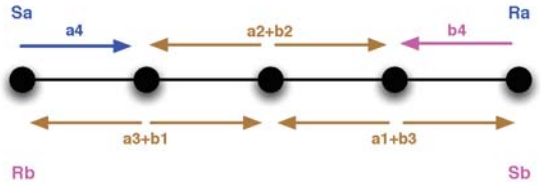
- information exchange between two nodes



Y. Wu, P. A. Chou, S.-Y. Kung, ``Information exchange in wireless networks with network coding and physical-layer broadcast,'' Microsoft Technical Report, MSR-TR-2004-78, Aug. 2004

Coding advantages

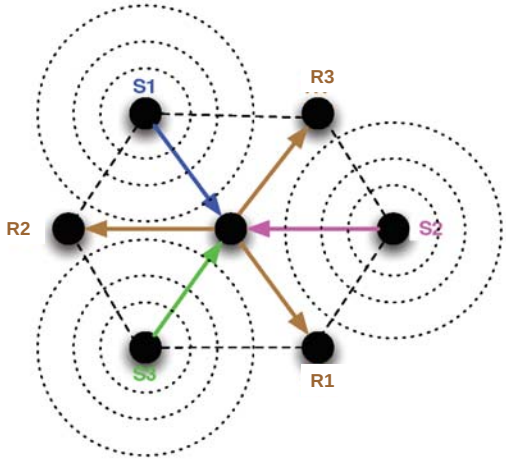
- information exchange between two nodes



Y. Wu, P. A. Chou, S.-Y. Kung, ``Information exchange in wireless networks with network coding and physical-layer broadcast,'' Microsoft Technical Report, MSR-TR-2004-78, Aug. 2004

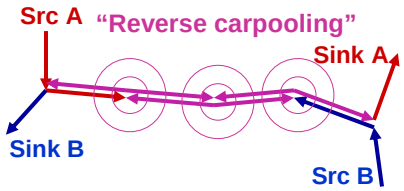
Coding advantages

- Intersection of paths



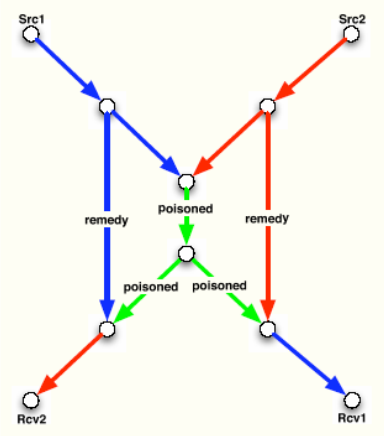
Construction approaches

- Use the information exchange, path intersection and butterfly scenarios as building blocks for constructing network codes
- Opportunistic (COPE): choose routes independently, code where possible [KatKatHu+05]
- Optimization-based: allows for optimality guarantees
  - Dynamic/linear/integer programming [TraRatLun+06,EffHoKim06]
  - Back pressure based [HoChaHan06,EryLun06]



XOR coding across pairs of unicasts

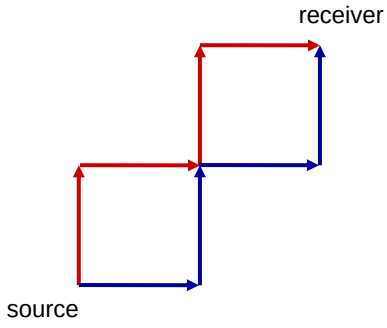
- Canonical butterfly module [RKH05]:



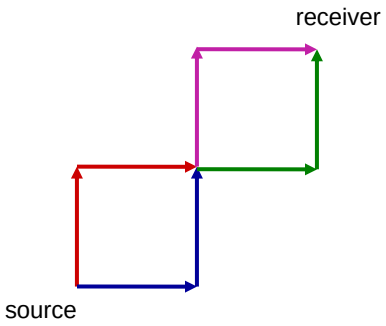
XOR coding across pairs of unicasts

- If we limit coding to pairs of uncoded or decoded flows, the problem becomes one of optimally fitting together canonical modules
- Can form a linear optimization problem whose constraints are:
  - Conservation of uncoded, poison and remedy flows
  - Conversion rules

Robustness to link failures

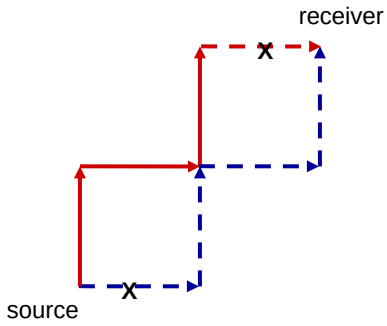


Multiple path routing

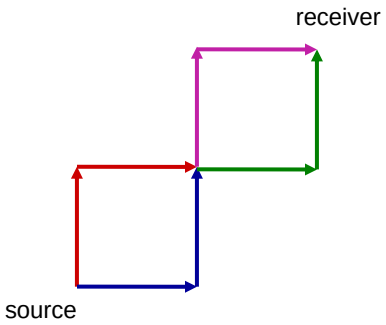


Network coding

Robustness to link failures

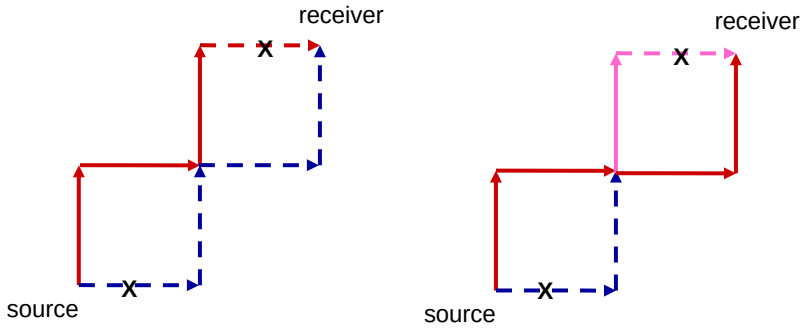


Multiple path routing



Network coding

Robustness to link failures

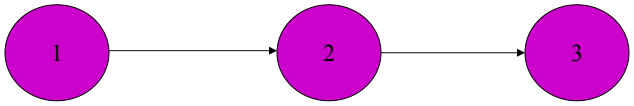


Multiple path routing

Network coding

•Network coding fully utilizes remaining network capacity

Reliability to probabilistic erasures



ε_{12} : Erasure probability on link (1, 2).
ε_{23}: Erasure probability on link (2, 3).

End-to-end erasure coding:
- Capacity is $(1-\varepsilon_{12})(1-\varepsilon_{23})$ packets per unit time.

As two separate channels:
- Capacity is $\min(1-\varepsilon_{12}, 1-\varepsilon_{23})$ packets per unit time.
- Can use block erasure coding on each channel. But delay is a problem.

Erasure reliability

- For erasures, correlated or not, we can in the multicast case deal with *average* flows uniquely [LME04], [LMK05], [DGPHE04]
- We consider a scheme [LME04] where
  - nodes store received packets in memory;
  - random linear combinations of memory contents sent out at every transmission opportunity (without waiting for full block).
- Scheme gets to capacity under arbitrary coding at every node for
  - unicast and multicast connections
  - networks with point-to-point and broadcast links.

Comments for erasure reliability

- Particularly suitable for
  - overlay networks using UDP, and
  - wireless packet networks (have erasures and can perform coding at all nodes).
- Code construction is completely decentralized.
- Scheme can be operated *ratelessly* - can be run indefinitely until successful reception.

Adversarial errors

- Network coding needed for optimal rate in multicast and in networks with packet losses and failures
  - Promising near-term applications in peer-to-peer and ad hoc networks; possibility of compromised participating nodes
- Information theoretic techniques for detecting and correcting errors introduced by an adversary who observes and controls unknown subsets of links/transmissions
  - Network coding facilitates use of a subgraph containing multiple paths to each sink, which can help security
  - However, coding at intermediate nodes causes error propagation → traditional approaches not suitable

Model

- Adversary knows the entire message and the coding scheme, but possibly not some of its random choices, e.g. random coding coefficients/random hash functions
- We consider a batch of exogenous source packets transmitted by distributed random network coding to a sink node which may be part of a unicast or multicast
- Adversary injects packets that may contain arbitrary errors; sink receives packets that are random linear combinations of the source and adversarial packets
- We will consider a few variants of this model which differ in terms of the adversary's knowledge and transmission capacity

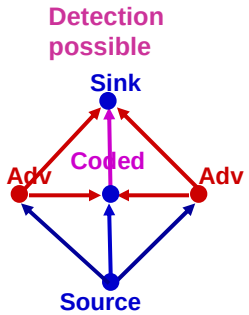
Detection and correction of adversarial errors

- Error detection/error correction capability added to random network coding scheme by adding appropriately designed redundancy; the only changes are at source and sink
- For error correction, overhead is lower bounded in terms of the number of adversarial transmissions as a proportion of the source-sink minimum cut
- For error detection, overhead can be traded off flexibly against detection probability and coding field size
- Error detection scheme can be used for low overhead monitoring when an adversary is not known to be present, in conjunction with a higher overhead error correction scheme activated upon detection of an adversary

Detection of adversarial errors

- Augment each source packet with a flexible number of hash symbols
- As long as not all adversarial packets have been designed with knowledge of the random coding combinations present in all packets received at the sink, adversarial errors result in decoded packets having non-matching data and hash values w.h.p.
- No limit on adversary's transmission capacity, require only that adversary has imperfect knowledge of random code

1	0	A	h_A
0	1	B	h_B



T. Ho, B. Leong, R. Koetter, M. Médard, M. Effros and D. R. Karger, "Byzantine Modification Detection in Multicast Networks with Random Network Coding", submitted to IEEE Transactions on Information Theory, 2006.

Error detection scheme

- Let each source packet contain n header/payload symbols $x_1,\ldots, x_n$ and k<n hash symbols $h_1, \ldots, h_k$, where n and k are design parameters which determine overhead
- $h_1 = \varphi(x_1,\ldots, x_t) = x_1^2 + \ldots + x_t^{t+1}$

  ...

  $h_k = \varphi(x_{t(k-1)+1},\ldots, x_n)$

  where $t = \lceil n/k \rceil$
- Sink observation Y=TX+UZ is the sum of a random linear transform T of source data X and a random linear transform U of adversarial errors Z
- Decoded packets given by $X+T^{-1}UZ$

Error detection performance

- For symbol length log q bits, if the sink receives s linearly independent combinations of source packets (which may be coded together with any number of adversarial packets), and at least one packet is erroneous, then

  a) for at least s decoded packets, the adversary cannot determine which of a set of at least q-1 possible values will be obtained

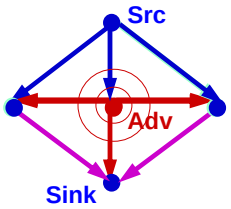
  (values can be partitioned into sets of the form $\{\mathbf{v}+\lambda\mathbf{w} : \lambda \in F_q\}$)

  b) the detection probability is at least $1-((t+1)/q)^s$
- Example:
  - With 2% overhead (t=50), symbol length=7 bits, s=5, the detection probability is 98.9%
  - With 1% overhead (t=100), symbol length=8 bits, s=5, the detection probability is 99.0%

Correction of adversarial errors

- C= capacity from source to sink
- z= capacity from adversary to sink
- n= length of each packet
- Sink receives Y=TX+UZ= T'X+E where
  - coefficient matrix T'= T+UL is $C \times b$
  - source matrix X is $b \times n$
  - error matrix E= U(Z-LX) is $C \times n$, rank $\leqslant$ z
- Note that if $b \leqslant C-z$, the column spaces of T' and E are linearly independent w.h.p.

Correction possible



S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, "Resilient Network Coding in the Presence of Byzantine Adversaries," Infocom 2007.

Case 1: Shared secret algorithm

- Source and sink share a low-rate secure secret channel, adversarial capacity z<C
- Source uses secret channel to send C random symbols $r_1,\ldots, r_C$ and corresponding hash vectors $h(r_i,X) = X\,[r_i\ r_i^2 \ldots r_i^n]^T$
- Sink calculates syndrome matrix S whose i^{th} column is the difference between $T'h(r_i,X)$ and the corresponding hash of the received data $h(r_i,Y)$, which is in the column space of E
- Since the adversary does not know $r_1,\ldots, r_C$ w.h.p. S has the same column space as E
- Since column spaces of T' and E are linearly independent, can solve Y=T'X+E for X
- For b= C-z, asymptotically achieves optimal rate

X:

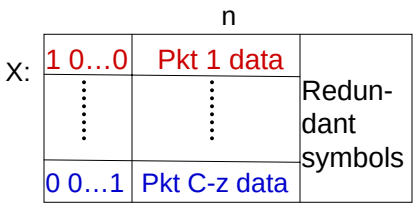
1 0…0	Pkt 1 data
⋮	⋮
0 0…1	Pkt b data

secret:

$r_1 \quad \ldots \quad r_C$
$h(r1,X) \ldots h(r_C,X)$

Case 2: Omniscient adversary algorithm

- Adversary knows everything, has transmission capacity $z<C/2$
- Source adds $(z+\varepsilon)n$ redundant symbols to header/data symbols s.t. resulting value X satisfies $(z+\varepsilon)n$ randomly chosen linear constraints and forms C-z packets of n symbols each
- W.h.p. over random constraints and random code, for all q^{zn} possible values of the set of adversarial packets, the sink can construct and solve a system of linear equations to obtain source data
- Optimal rate of C-2z is achieved asymptotically with n



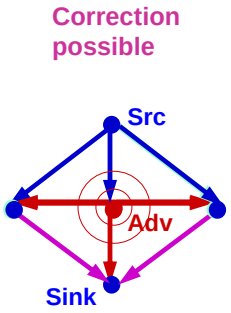
Case 3: Limited adversary algorithm

- Adversary observes y transmissions and controls z, where $2z+y<C$
- A small fraction of each packet consists of redundant information generated as follows:
  - Use shared secret algorithm to generate secret hash information
  - Use omniscient adversary algorithm to generate additional redundancy protecting a mix of secret hash information with extra random symbols (for secrecy)
- Sink first decodes secret hash information, then decodes message using shared secret algorithm
- Optimal rate of C-z is asymptotically achieved

Correction of adversarial errors

Common intuition behind algorithms:

- A sink observes the sum of a random linear transform T of data X transmitted by source and a random linear transform U of Z transmitted by adversary
- Design redundancy in source transmissions to satisfy constraints that adversarial data cannot (or is unlikely to) satisfy
- Algebraic decoding algorithms using the observations that:
  - U has rank $\leqslant$ z (#adversarial transmissions)
  - If b $\leqslant$ C-z, the column spaces of T and U are linearly independent w.h.p.



Distributed compression

- Distributed compression of correlated data by linear network coding
  - Nonlinear decoding required
  - Generalization of Slepian Wolf capacity and error exponents to arbitrary networks
  - Back pressure technique dynamically trades off data rates across sources according to network state

T. Ho, M. Médard, M. Effros and R. Koetter, "Network Coding for Correlated Sources", Invited Paper, Conference on Information Sciences and Systems, 2004.

T. Ho, H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding", Allerton Annual Conference on Communication, Control, and Computing, 2005.