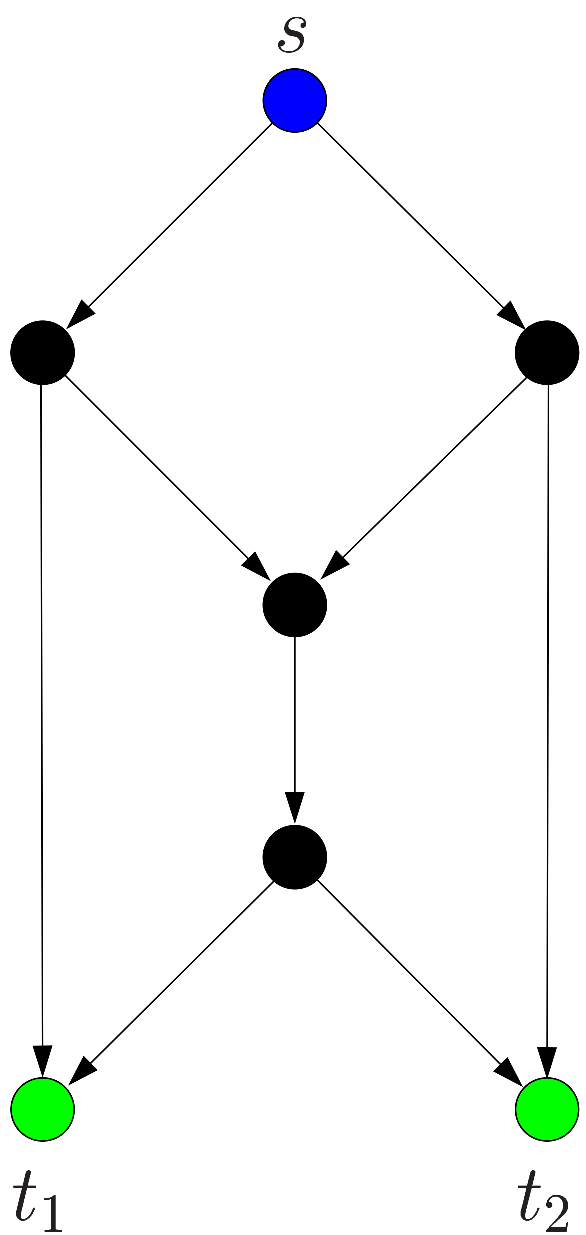
Secure Network Coding via Filtered Secret Sharing

Jon Feldman

Columbia University

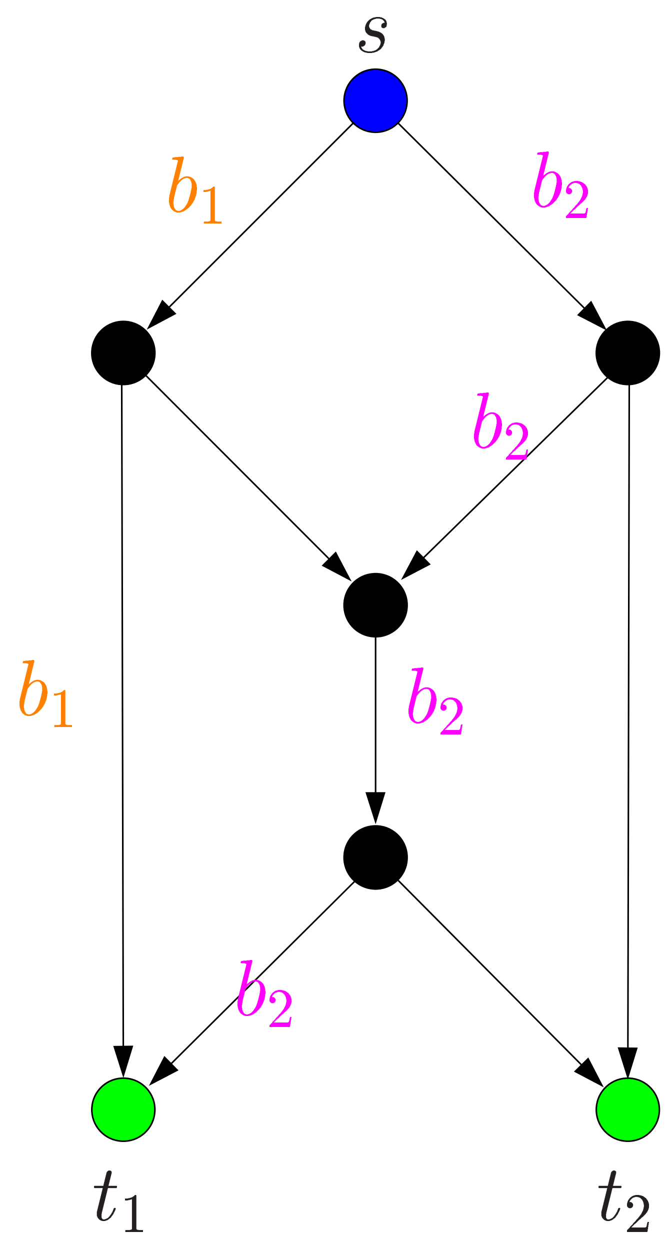
Joint Work with Cliff Stein, Tal Malkin and Rocco Servedio

Routing in a network



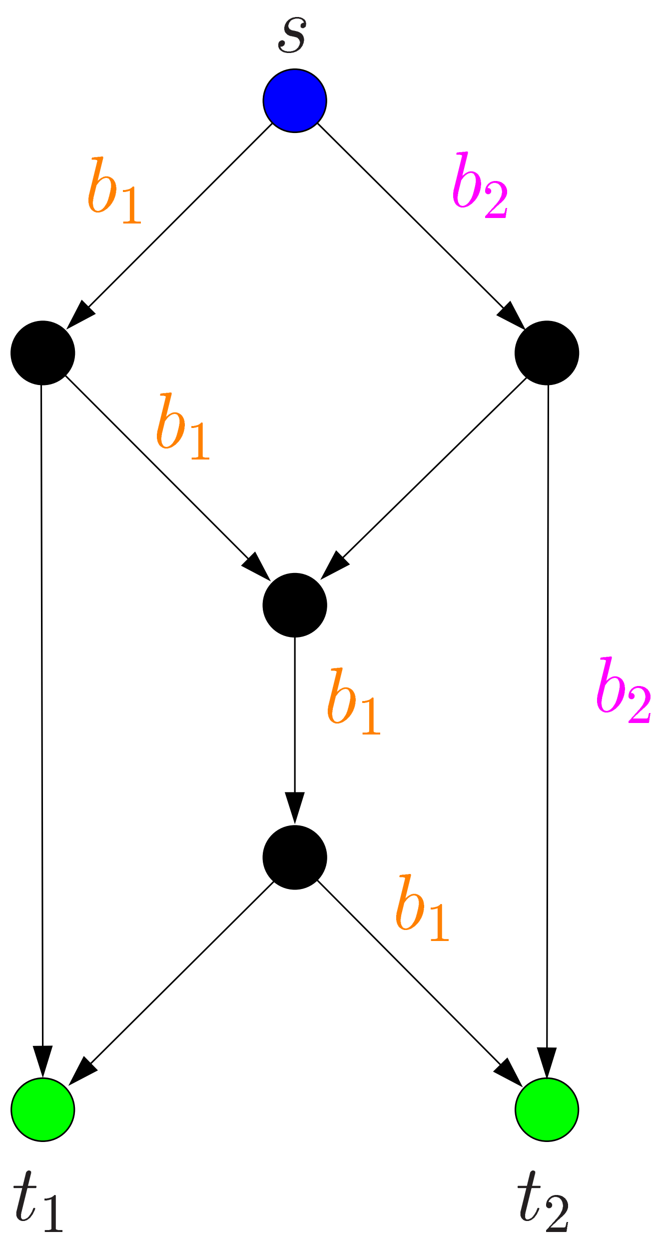
- Two bits b_1, b_2 at source s
- Send both bits to both t_1, t_2.

Routing in a network



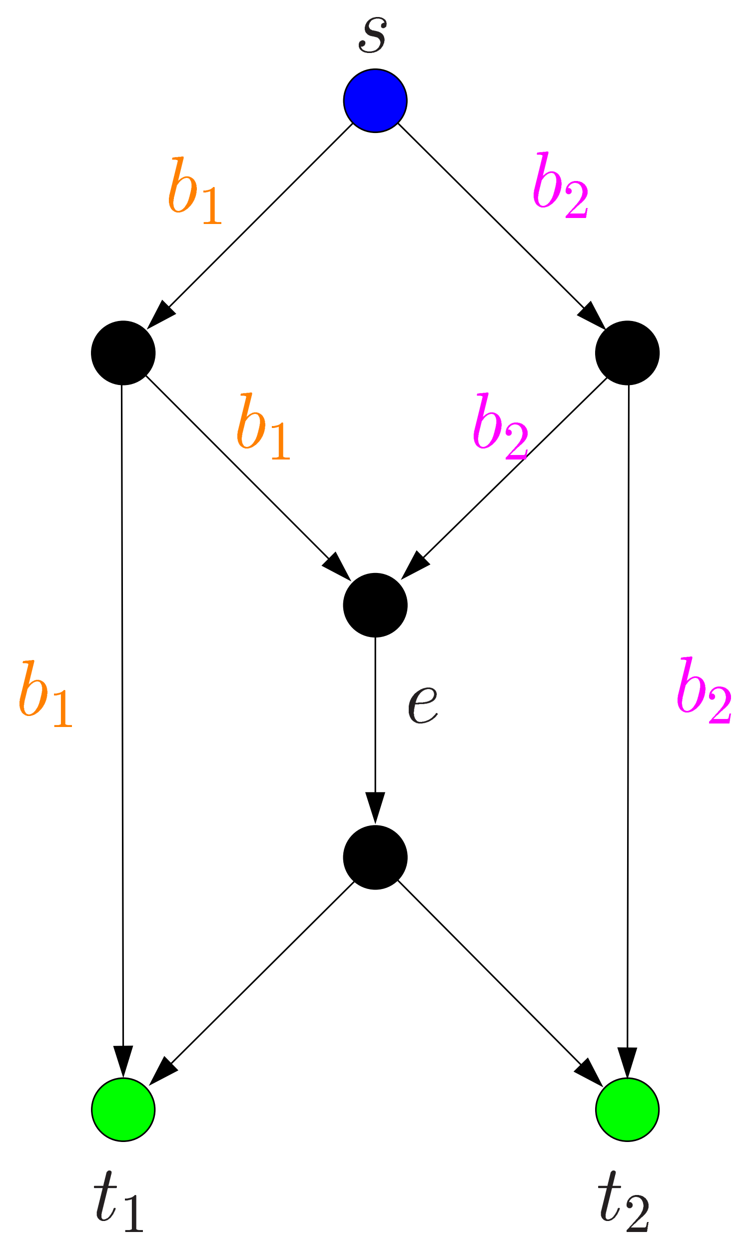
- Two bits b_1, b_2 at source s
- Send both bits to both t_1, t_2.
- Can send both bits to t_1

Routing in a network



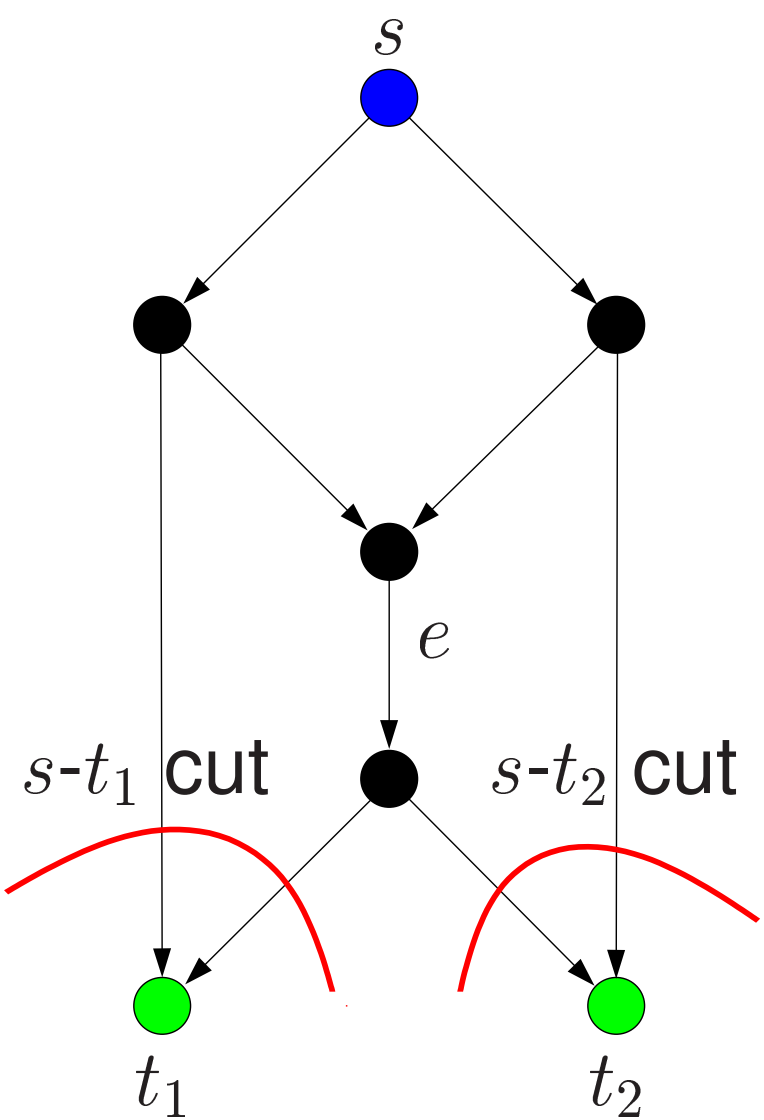
- Two bits b_1, b_2 at source s
- Send both bits to both t_1, t_2.
- Can send both bits to t_2

Network Coding



- Two bits b_1, b_2 at source s
- Send both bits to both t_1, t_2.
- Can't satisfy both simultaneously
  - wlog, send b_1 to the left, b_2 to the right.
  - t_1 has b_1 and t_2 has b_2.
  - What goes on edge e?

Minimum Cuts



- Two bits b_1, b_2 at source s
- Send both bits to both t_1, t_2.
- Can't satisfy both simultaneously
  - Can send from s to t_1 an amount equal to the minimum (s,t_1) cut.
  - Can send from s to t_2 an amount equal to the minimum (s,t_2)-cut.

Question

- Is it possible to multicast s.t. each sink receives the maximum number of bits (= min cut).
- What additional power do we need to add to store-and-forward routing?

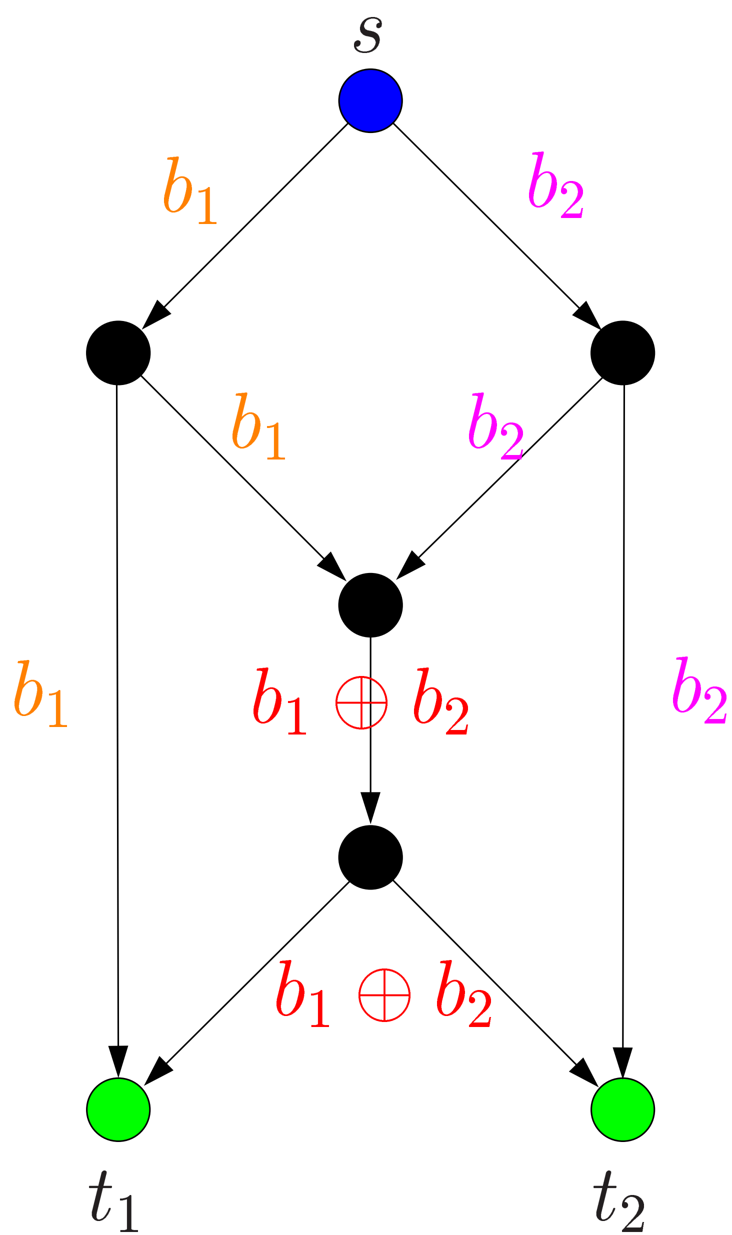
Question

- Is it possible to multicast s.t. each sink receives the maximum number of bits (= min cut).
- What additional power do we need to add to store-and-forward routing?

Answer

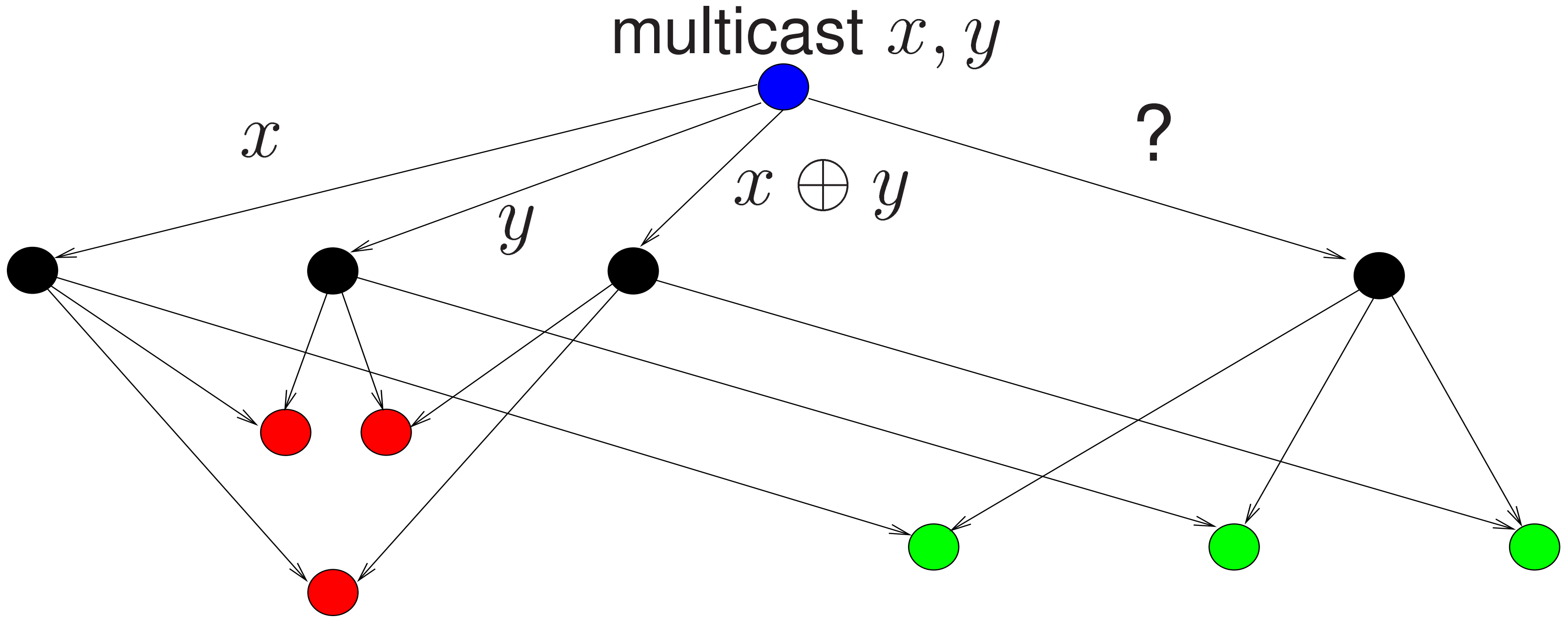
- Allow the nodes to do some computation.

Network Coding



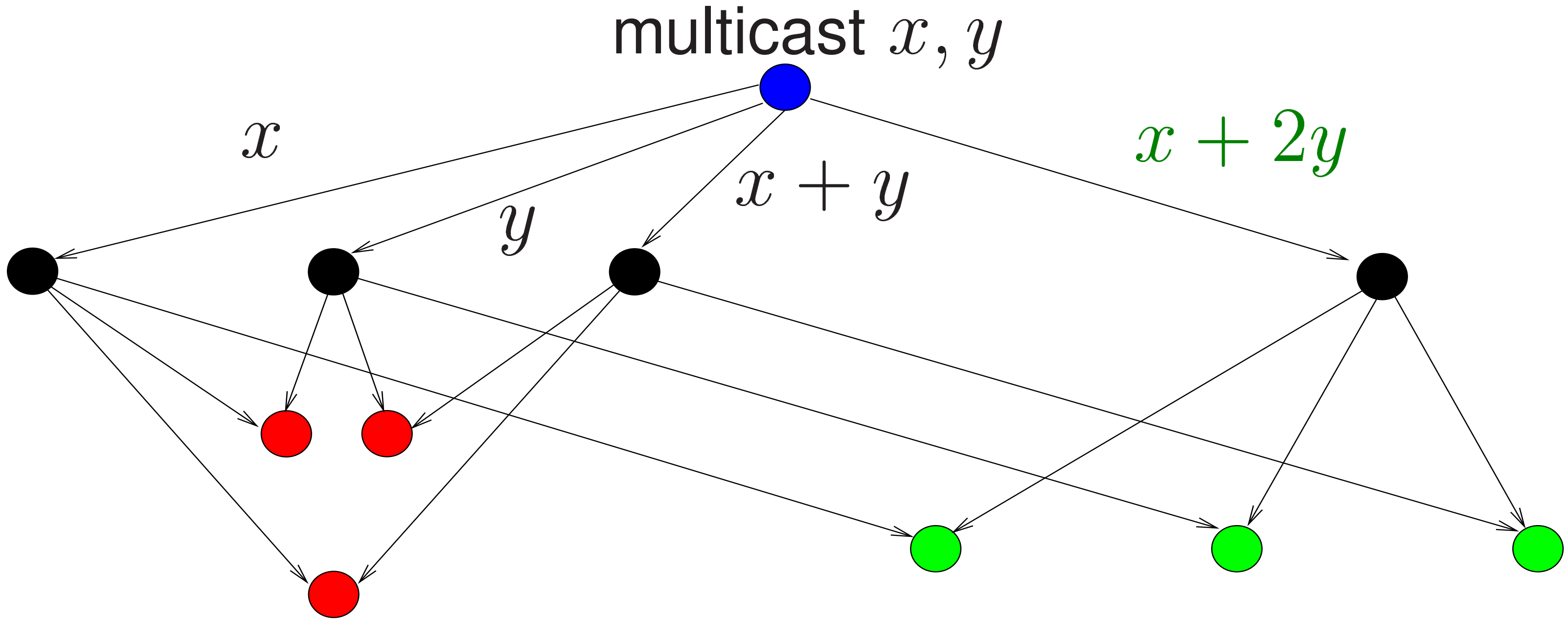
- Two bits b_1, b_2 at source s
- Send both bits to both t_1, t_2.
- Coding:
  - Send b_1, b_2 as before
  - Send $b_1 \oplus b_2$ on edge e.
  - Sinks can both recover b_1, b_2 by $\oplus$-ing symbols.

Another example



- Each sink (green) wants both bits.
- Only choice for ? edge is x, y or $x \oplus y$.
- None are acceptable.

Solution



- Use larger finite field (e.g., $\mathbb{F}_3$).
- Can set ? = $x + 2y$.
- Each sink can compute x and y.

Network Coding Questions

- Introduced by Ahlswede, Cai, Li and Yeung [2000]
- Network coding home page: 90 papers.
- Typical questions:
  - When is it possible to satisfy demand?
  - When do you get a “coding advantage” ? (...how big is the advantage?)
  - How do you construct a code?
  - How big an alphabet do you need?
  - Do linear codes suffice?
  - . . .

Practical applications

- Content distribution networks [GR 2004, LME 04]
- Ad-hoc, wireless networks [CWJ 03, WCK 04]
- Sensor networks, P2P, BitTorrent, etc.
- ...

Multicast case

- In any network code:

$$\text{\# symbols } s \to t_i \le \text{min } (s, t_i) \text{ cut.}$$

- Using network codes, one can send

$$n = \min_i \text{ [min } (s, t_i) \text{ cut]}$$

symbols to every sink simultaneously [LYC01]. (Need to use $\mathbb{F}_q$, where $q \ge$ # sinks.)

- Can find code (functions at routers) in poly-time [JSCEEJT, 2003].

Network Coding and Security

- Network coding: new model of transmission...
  - ...how do we make it secure?

- Security Models:
  - Cryptographic Security
  - Partial Security
  - Information Theoretic Security

- Wire-tap adversary: can “listen” to k edges of network.

The Trade-offs

- Rate: number of symbols of information sent

- Security: number of wires the adversary can tap.

- Bandwidth: edge capacity (in bits) required to communicate.

Want: high rate, high security, low bandwidth.

Past work on security of network coding

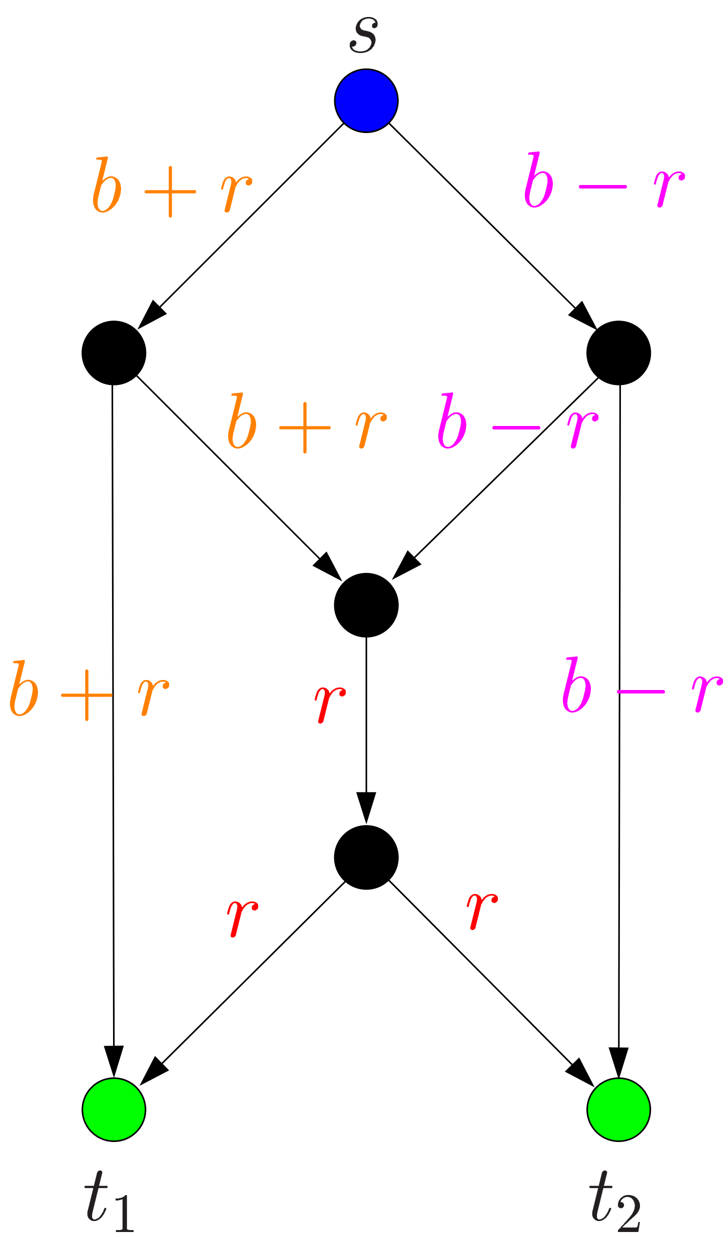
- Cai and Yeung[02]
  - wire-tap adversary: can look at any k edges.
  - Suff. conditions for $\exists$ secure multicast code.
- Jain[04]: More precise cond. (one terminal).
- Ho, Leong, Koetter, Médard, Effros, Karger [04]: Byzantine modification detection.

Past work on security of network coding

- Cai and Yeung[02]
  - wire-tap adversary: can look at any k edges.
  - Suff. conditions for $\exists$ secure multicast code.
- Jain[04]: More precise cond. (one terminal).
- Ho, Leong, Koetter, Médard, Effros, Karger [04]: Byzantine modification detection.

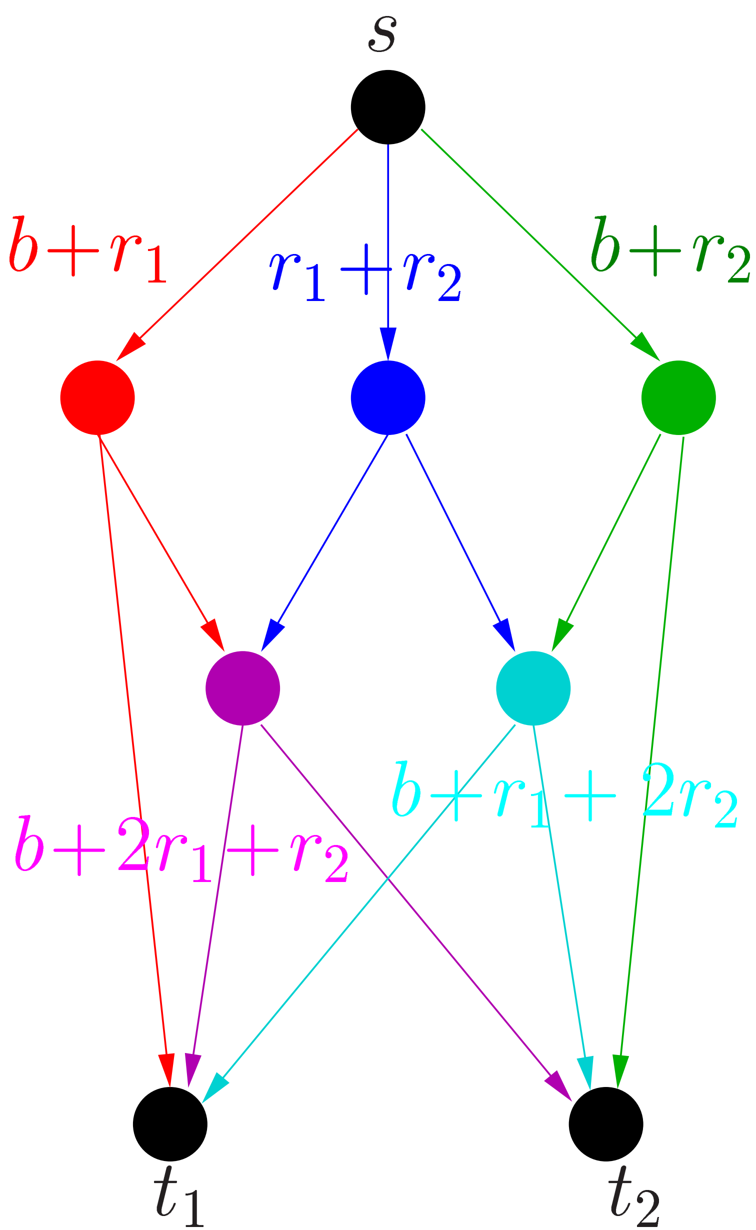
- Our work: precise analysis of wire-tap adversary, trade-offs between security, rate, edge bandwidth.

Making our Example Secure



- Use $\mathbb{F}_3 = \{0, 1, 2\}$.
- Less ambitious goal: Send *one* symbol $b \in \mathbb{F}_3$ to both sinks.
- Choose $r \in \mathbb{F}_3$ randomly.
- Can define symbols s.t. any single wire-tapper learns nothing about b, both sinks can compute b.

Another example



- Sink 1 knows $b + r_1$, $b + 2r_1 + r_2$, and $b + r_1 + 2r_2$.
- Sink 2 knows $b + r_2$, $b + 2r_1 + r_2$, and $b + r_1 + 2r_2$.

  $\Longrightarrow$ code multicasts b

- Adversary, given any two edges, cannot compute b.
- Adversary can, e.g., compute r_2, but this gives no information.

Linear Multicast Network Coding (No Security)

Given: Network $G = (V, E)$, source $s \in V$, sinks $T \subseteq V$, s.t. $n = \min_{t \in T} \{ \min (s,t)$ cut $\}$.

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.

Linear Multicast Network Coding (No Security)

$\boxed{\text{Given: Network } G = (V, E)\text{, source } s \in V\text{, sinks } T \subseteq V\text{, s.t. } n = \min_{t \in T} \{ \min (s,t) \text{ cut } \}.}$

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.
- Network code:
  - Define coding vectors $v[e] \in \mathbb{F}_q^n$ for each edge.
  - Edge carries symbol $m \cdot v[e]$.

Linear Multicast Network Coding (No Security)

Given: Network $G = (V, E)$, source $s \in V$, sinks $T \subseteq V$, s.t. $n = \min_{t \in T} \{ \min (s,t) \text{ cut } \}$.

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.
- Network code:
  - Define coding vectors $v[e] \in \mathbb{F}_q^n$ for each edge.
  - Edge carries symbol $m \cdot v[e]$.
- Feasibility of transmission:

  (i) Every $v[b, c]$ spanned by $\{v[a, b]\}_a$ (or $a = s$).

Linear Multicast Network Coding (No Security)

Given: Network $G = (V, E)$, source $s \in V$, sinks $T \subseteq V$, s.t. $n = \min_{t \in T} \{ \min (s,t) \text{ cut } \}$.

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.
- Network code:
  - Define coding vectors $v[e] \in \mathbb{F}_q^n$ for each edge.
  - Edge carries symbol $m \cdot v[e]$.
- Feasibility of transmission:

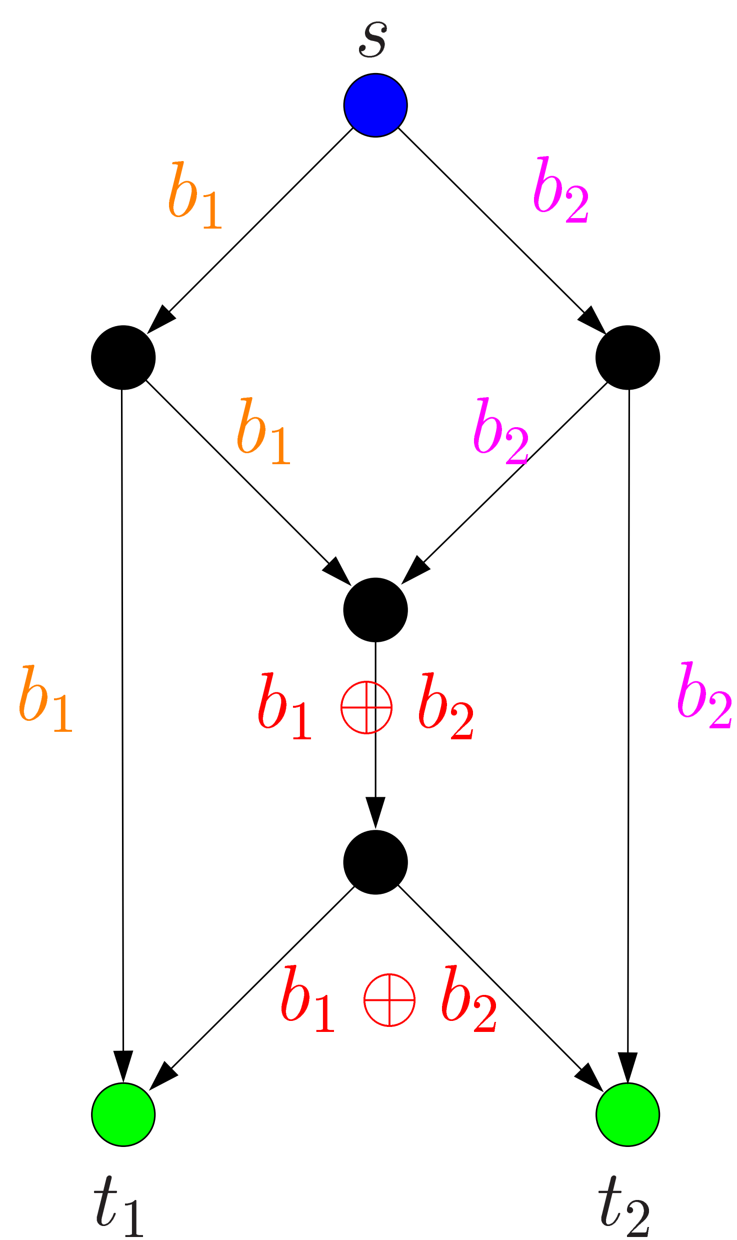
  (i) Every $v[b, c]$ spanned by $\{v[a, b]\}_a$ (or $a = s$).
- Recoverability at sinks:

  (ii) For all $t \in T$, the vectors $\{v[a, t]\}_a$ span $\mathbb{F}_q^n$.

Constructions of Linear Multicast Network Codes

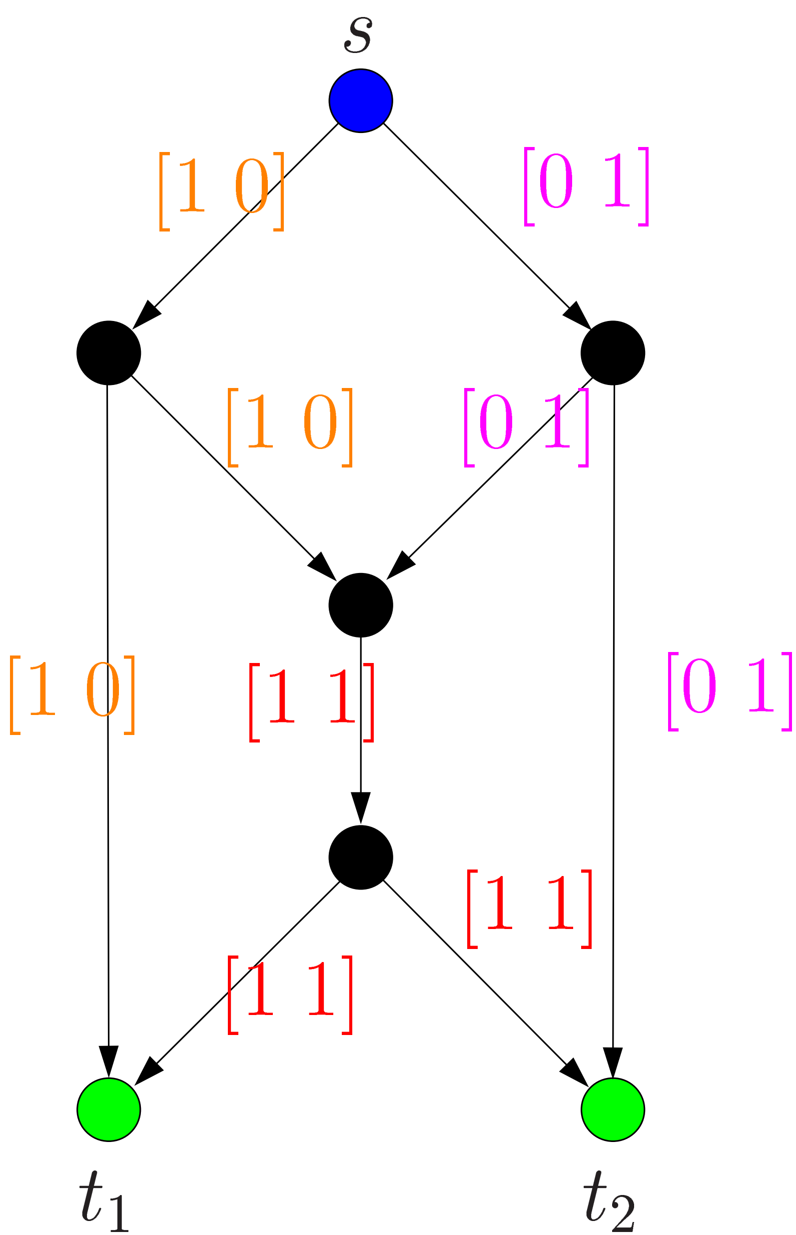
- [Li, Yeung, Cai, 01]: Solution (feasible, recoverable) always exists as long as $q \geq |T|$.
- [Rasala-Lehman, Lehman, 03]: There are instances requiring $q \geq \sqrt{|T|}$.
- [Koetter, Médard, 03]: Random feasible codes are recoverable w.h.p., with q suff. large.
- [Jaggi, Sers, Chou, Effros, Egner, Jain, Tolhuizen, 03]: Feasible, recoverable solution (coding vectors) can be found in poly($|E|$) time.

Existence of a network code



- Can express any solution using "coding vectors" on edges...

Existence of a network code



- Edge vector spanned by vectors into tail node ...linear combo determines routing function.
- To decode, each sink must have linearly independent set of n vectors... ...solve system of equations at sink to recover info.

Wire-Tap Model, Randomness at the Source

Task: design function $m = f(x, r)$ at source, coding vectors on edges s.t.:

- Coding vectors satisfy feasibility,
- Information x recoverable at each sink,
- Information x secure against tap(k) adversary.

Study achievable trade-offs:

Security = k = # edges tapped $< n$.

Rate = t = $|x|$ = # information symbols multicast.

Edge Bandwidth = $\log q$, where symbols in $\mathbb{F}_q$.

Security, Rate and Bandwidth

Security = k = # edges tapped $< n$.

Rate = t = $|x|$ = # information symbols multicast.

Edge Bandwidth = $\log q$, where symbols in $\mathbb{F}_q$.

- Easy to show: $t \leq n - k$.

Security, Rate and Bandwidth

Security = k = # edges tapped $< n$.

Rate = t = $|x|$ = # information symbols multicast.

Edge Bandwidth = $\log q$, where symbols in $\mathbb{F}_q$.

- Easy to show: $t \leq n - k$.
- Cai and Yeung [02]: If $q > \binom{|E|}{k}$, can send $t = n - k$ symbols securely.
  - Construction time $\approx \binom{|E|}{k}$.

Our Results

- If you give up a little capacity, bandwidth requirement reduced significantly:

Thm: For any $c > 1$, if $q \geq |E|^{\Omega(\frac{1}{c-1})}$, can send $t = n - ck$ symbols securely.

- Algorithm: poly-time, secure w.h.p.
- If $k = \Theta(|E|)$, only need $q \geq 2^{\Omega(\frac{1}{c-1})}$.

Our Results

- If you give up a little capacity, bandwidth requirement reduced significantly:

Thm: For any $c > 1$, if $q \geq |E|^{\Omega(\frac{1}{c-1})}$, can send $t = n - ck$ symbols securely.

  - Algorithm: poly-time, secure w.h.p.
  - If $k = \Theta(|E|)$, only need $q \geq 2^{\Omega(\frac{1}{c-1})}$.

- If you do not give up capacity, then bandwidth might have to be large:

Thm: If $t = n - k$, then there are examples where all solutions (using this method) must have $q \gtrsim |E|^{\sqrt{k}}$.

Relation w/ Cai & Yeung

- Core Lemma of Cai and Yeung: If one can construct a matrix with certain independence properties relative to the coding vectors, then the network code can be altered to achieve security.

Relation w/ Cai & Yeung

- Core Lemma of Cai and Yeung: If one can construct a matrix with certain independence properties relative to the coding vectors, then the network code can be altered to achieve security.
- Our extensions:
  - Independence properties are also necessary.
  - Using orthogonal space, re-cast as coding theory problem.
  - Give up some capacity to make coding problem solvable.
  - Use necessary direction, covering radius, to prove negative result.
  - Observation: don’t alter code, just input.

Making a Given Network Code Secure

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

Making a Given Network Code Secure

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.

Making a Given Network Code Secure

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.

Making a Given Network Code Secure

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.
- Security condition: If adversary looks at any k edges E', can learn nothing about x; i.e.,

$$|E'| \leq k \implies (f(x, r) \cdot v[e])_{e \in E'} \text{ indep. of } x.$$

Making a Given Network Code Secure

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.
- Security condition: If adversary looks at any k edges E', can learn nothing about x; i.e.,

$$|E'| \leq k \implies (f(x, r) \cdot v[e])_{e \in E'} \text{ indep. of } x.$$

- Recoverability, feasibility follow immediately (as long as x can be determined from $f(x, r)$).

Making a Given Network Code Secure

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.
- Security condition: If adversary looks at any k edges E', can learn nothing about x; i.e.,

  $|E'| \leq k \implies (f(x, r) \cdot v[e])_{e \in E'}$ indep. of x.

- Recoverability, feasibility follow immediately (as long as x can be determined from $f(x, r)$).
- Advantage: don't need to alter network code.

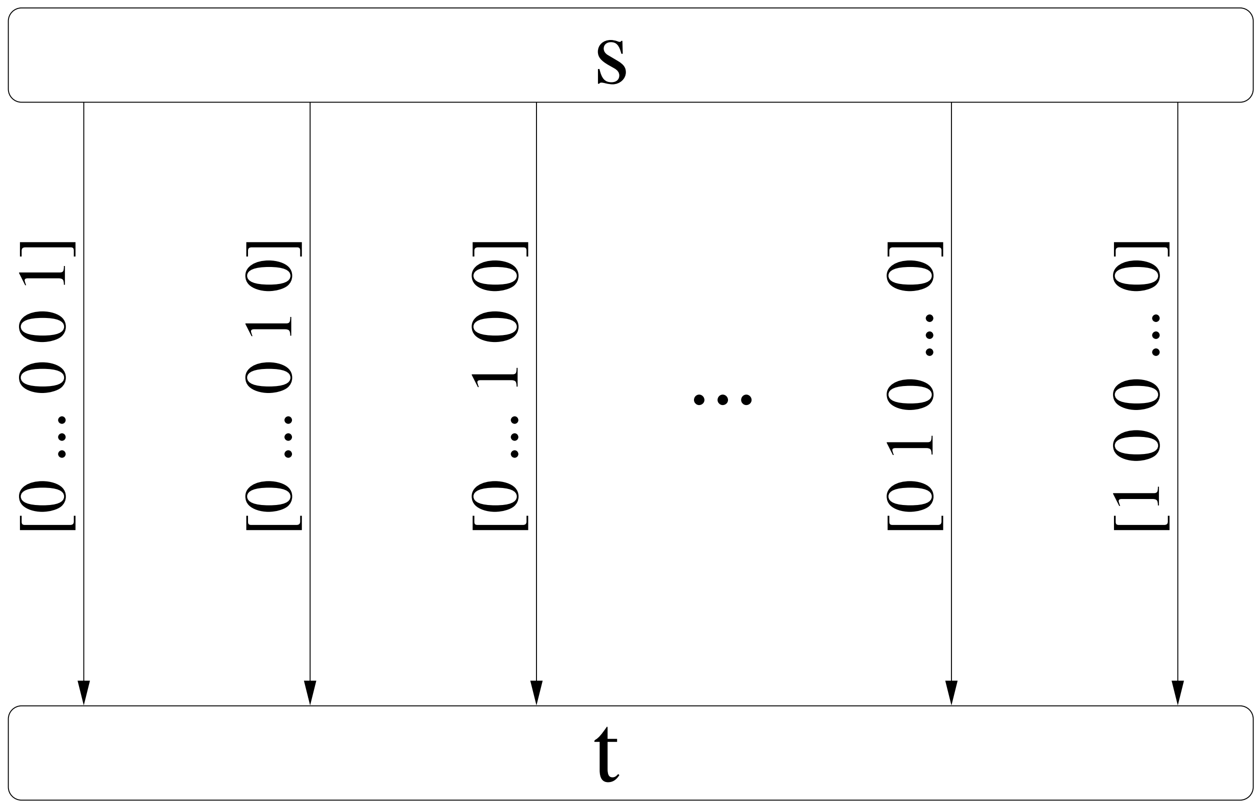
Secret Sharing

Task: divide up information into pieces so that the sinks can reconstruct information, but an adversary who sees some of the pieces cannot.

- This sounds like secret sharing

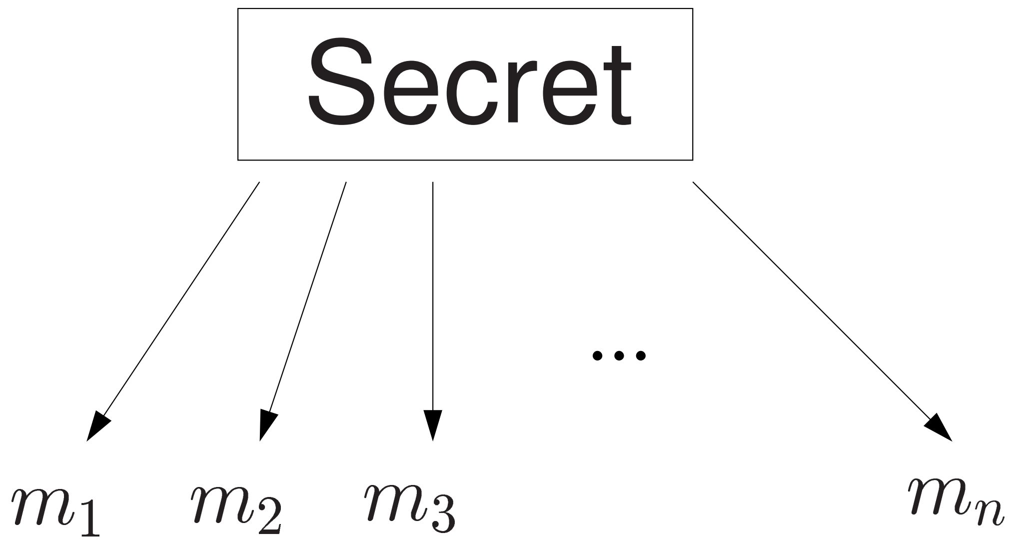
Connection to Secret Sharing

- Simple case: single source/sink, n parallel edges.



- Modest goal $t = 1$: send one symbol $x \in \mathbb{F}_q$.

Secret Sharing (Shamir)



- Dealer has "secret" $x \in \mathbb{F}_q$.
- Distribute shares $m_1, \ldots, m_n$ s.t.
  - Given any all shares, can recover secret.
  - Given any k shares, learn nothing.

Connection to Secret Sharing

- Encoding:
  - Choose $(r_1, \ldots r_k)$ at random.
  - Let $p(z) = x + r_1 z + r_2 z^2 + \cdots + r_k z^k$
  - Set message $m = (p(\alpha_1), \ldots, p(\alpha_n))$.
  - (Encode (x,r) using a Reed-Solomon code.)

Connection to Secret Sharing

- Encoding:
  - Choose $(r_1, \ldots r_k)$ at random.
  - Let $p(z) = x + r_1 z + r_2 z^2 + \cdots + r_k z^k$
  - Set message $m = (p(\alpha_1), \ldots, p(\alpha_n))$.
  - (Encode (x,r) using a Reed-Solomon code.)
- Decoding:
  - Knowing $(k+1)$ points $p(\alpha_i)$ reveals p (interp.).
  - Knowing k or fewer m_i's tells you nothing.
  - Works for any $k \leq n - 1$.

Filtered Secret Sharing

- “Classical” secret sharing: adversary has k shares.

Filtered Secret Sharing

- “Classical” secret sharing: adversary has k shares.
- Network coding: adversary has k linear combinations of all n shares.

Filtered Secret Sharing

- "Classical" secret sharing: adversary has k shares.
- Network coding: adversary has k linear combinations of all n shares.
- Shares $(m_1, \ldots, m_n)$ passed through coding "filter" before being revealed to adversary.

Filtered Secret Sharing

- "Classical" secret sharing: adversary has k shares.
- Network coding: adversary has k linear combinations of all n shares.
- Shares $(m_1, \ldots, m_n)$ passed through coding "filter" before being revealed to adversary.

- Given: n-by-m full-rank "filter" matrix V over $\mathbb{F}_q$.
- Find: Function $f : \mathbb{F}_q^t \times \mathbb{F}_q^\ell \to \mathbb{F}_q^n$ such that:

  For all n-by-k submatrices V' of V,

  over random $r \leftarrow \mathbb{F}_q^\ell$,

  we have $f(x, r) \cdot V'$ indep. of x $(\forall x)$.

Filtered Secret Sharing

- Given: n-by-m full-rank "filter" matrix V over $\mathbb{F}_q$.
- Find: Function $f : \mathbb{F}_q^t \times \mathbb{F}_q^\ell \to \mathbb{F}_q^n$ such that:

  For all n-by-k submatrices V' of V,

  over random $r \leftarrow \mathbb{F}_q^\ell$,

  we have $f(x, r) \cdot V'$ indep. of x.

- Classical: special case $t = 1$, $m = n$, $V = I$.
- For network coding:
  - $m = |E|$, V is n-by-$|E|$ matrix of coding vectors.
  - Ignores network topology.

Solving Filtered Secret Sharing Problem

It turns out that filtered secret sharing is a special case of a coding theory problem.

Design of Linear Error-Correcting Codes

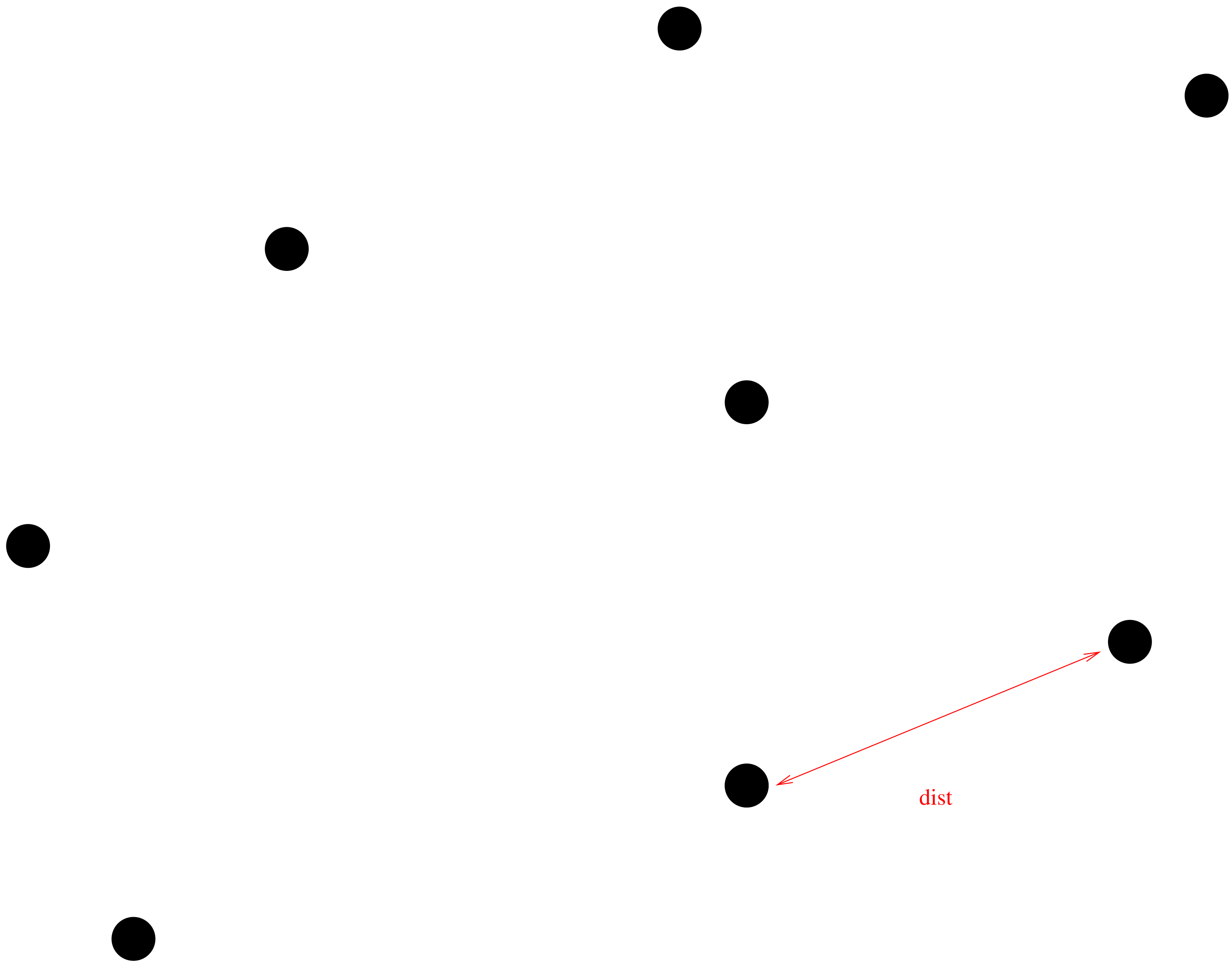
- Code $C \subseteq \mathbb{F}_q^m$, where $|C| = q^\kappa$.
- Distance(C) = $\min\limits_{y \in C, y' \in C} \Delta(y, y)$.
- Goal in coding theory:
  - For given rate κ/m, find code with large distance.

Design of Linear Error-Correcting Codes

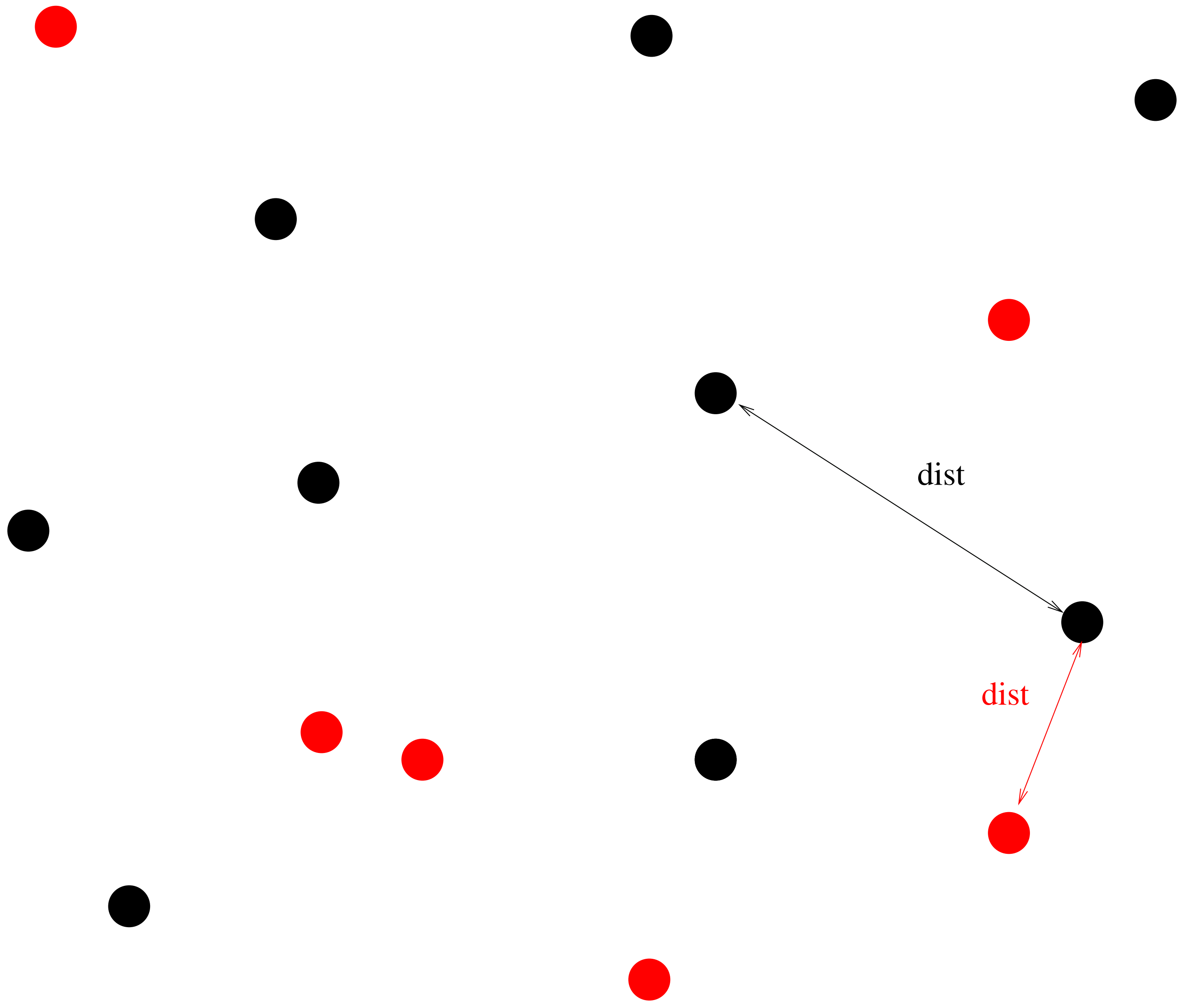
- Code $C \subseteq \mathbb{F}_q^m$, where $|C| = q^\kappa$.
- Distance(C) = $\min_{y \in C, y' \in C} \Delta(y, y)$.
- Goal in coding theory:
  - For given rate κ/m, find code with large distance.

- Generalization:
  - Codewords must be far from each other, and some other given points.

Design of Linear Error-Correcting Codes

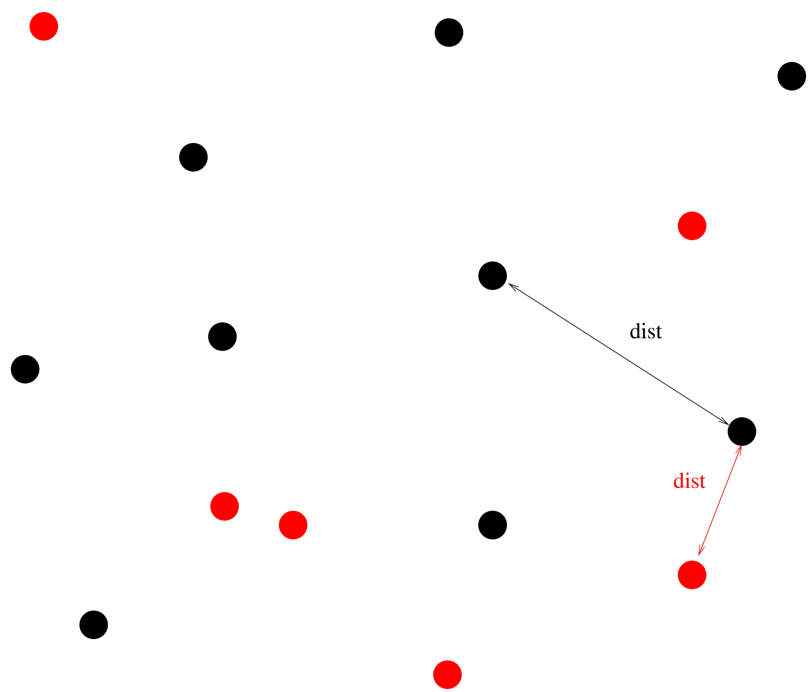


Design of Generalized Coding Problem



Generalized Coding Problem

"Span Distance Problem" (over field $\mathbb{F}_q$): Given a λ-by-m matrix A , find a κ-by-m matrix B such that all $y \in C_B \setminus 0^n$ have $\Delta(y, C_A) \geq k$.



Filtered Secret sharing $\iff$ Span Distance

- Filtered secret sharing (linear f) is a special case of the span distance problem:

$\exists$ (linear) f.s.s. solution f with $t = n - ck$

$\iff$ (for all $c \geq 1$)

$\exists$ solution to the **span distance problem** with

- $A = V^{\perp}$, (A generates null space of V)
- $\kappa = t$,
- Required distance = k.

- Parameter $c \geq 1$ for network coding application: amount of capacity given up.

Positive result

- Given any A, choose B randomly.

Positive result

- Given any A, choose B randomly.
- Analysis like random lin. codes on G-V bound:
  - Each vector in C_B has $\leq |C_A|\mathrm{Vol}_q(k,m)/q^m$ prob. of having dist. $\leq k$ from C_A.
  - Union bound over q^κ codewords C_B:

Random B has $\Delta_s(A,B) > k$ w/ prob.

$$\begin{aligned} &\geq 1 - q^\kappa q^\lambda q^{-m}\mathrm{Vol}_q(k,m) \\ &= 1 - q^{-ck}\mathrm{Vol}_q(k,m) \end{aligned}$$

Positive result

- Given any A, choose B randomly.
- Analysis like random lin. codes on G-V bound:
  - Each vector in C_B has $\leq |C_A|\text{Vol}_q(k,m)/q^m$ prob. of having dist. $\leq k$ from C_A.
  - Union bound over q^κ codewords C_B:

Random B has $\Delta_s(A,B) > k$ w/ prob.

$$\begin{aligned} &\geq 1 - q^\kappa q^\lambda q^{-m}\text{Vol}_q(k,m) \\ &= 1 - q^{-ck}\text{Vol}_q(k,m) \end{aligned}$$

- In general, Pr > 0 if $q > m^{\Omega(\frac{1}{c-1})}$.
- For $k = \Theta(m)$, need only $q > 2^{\Omega(1/(c-1))}$.

Negative Result: Using the Covering Radius

- *Covering radius*$(C_A) = \max_{z \in \mathbb{F}_q^m} \Delta(z, C_A)$.

Negative Result: Using the Covering Radius

- *Covering radius*$(C_A) = \max_{z \in \mathbb{F}_q^m} \Delta(z, C_A)$.
- If C_A has covering radius $k \implies$ no $b \in \mathbb{F}_q^m$ (let alone subspace C_B) has dist $> k$ from C_A.

Negative Result: Using the Covering Radius

- *Covering radius*$(C_A) = \max_{z \in \mathbb{F}_q^m} \Delta(z, C_A)$.
- If C_A has covering radius $k \implies$ no $b \in \mathbb{F}_q^m$ (let alone subspace C_B) has dist $> k$ from C_A.
- Setting $c = 1$, using result of [Cohen, Frankl 85]:

Thm: $\forall\ \lambda, \kappa$ s.t. $\left(\lambda = m - \frac{\log m}{\log q} - \frac{\log \mathsf{Vol}_q(k,m)}{\log q} + 2\log m + \log q + \log\ln q\right)$ and $\left(k + \kappa < m - \lambda = \frac{\log m}{\log q} + \frac{\log \mathsf{Vol}_q(k,m)}{\log q} - 2\log m - \log q - \log\ln q\right)$, $\exists\ A$ s.t. $\nexists\ B$ where $\Delta_s(A, B) > k = m - \lambda - \kappa$.

Negative Result: Using the Covering Radius

- *Covering radius*(C_A) = $\max_{z \in \mathbb{F}_q^m} \Delta(z, C_A)$.
- If C_A has covering radius $k \implies$ no $b \in \mathbb{F}_q^m$ (let alone subspace C_B) has dist $> k$ from C_A.
- Setting $c = 1$, using result of [Cohen, Frankl 85]:

Thm: $\forall\ \lambda, \kappa$ s.t. $\left(\lambda = m - \frac{\log m}{\log q} - \frac{\log \mathrm{Vol}_q(k,m)}{\log q} + 2\log m + \log q + \log\ln q\right)$ and $\left(k + \kappa < m - \lambda = \frac{\log m}{\log q} + \frac{\log \mathrm{Vol}_q(k,m)}{\log q} - 2\log m - \log q - \log\ln q\right)$, $\exists\ A$ s.t. $\nexists\ B$ where $\Delta_s(A,B) > k = m - \lambda - \kappa$.

- $\exists$ reasonable settings of λ, κ, k, where $\exists A$ s.t. $q \geq m^{\Omega(\sqrt{k}/\log k)}$ if B exists. (Contrast to C/Y upper bound $\binom{m}{k}$.)

Conclusions

- Secure network coding under wiretap
  $\iff$ filtered secret sharing
  $\iff$ span distance problem
- To achieve security: trade-off between rate $(t = n - ck)$ and required link bandwidth $(\log q)$.
  - Sacrificing small amount of capacity allows large savings in required bandwidth.

Future Work

- Better upper/lower bounds ($c = 1$, or in general).
- Consider (network topology, code design, security, robustness) simultaneously.
- Allow more power at nodes [Jain: random bits].
- Relax notion of security [Jain: computationally bounded adversary].
- Different adversaries, non-linear network codes, non-multicast?