Random Linear Intersession Network Coding With Selective Cancelling

Chih-Chun Wang — Purdue University

2009 IEEE Information Theory Workshop,
Taormina, Italy, 10/15/2009

joint work with Ness B. Shroff with The Ohio State University

Intrasession Network Coding

Network coding within a single multicast session $(s, \{d_j\})$.

- The capacity of intrasession network coding is governed by min-cut/max-flow theorem. [Ahlswede *et al*. 00], [Li *et al*. 03], and [Koetter *et al*. 03].

Intrasession Network Coding

Network coding within a single multicast session $(s, \{d_j\})$.

- The capacity of intrasession network coding is governed by min-cut/max-flow theorem. [Ahlswede *et al.* 00], [Li *et al.* 03], and [Koetter *et al.* 03].
- Capacity-achieving random linear network code [Ho *et al.* 06] and the associated polynomial time construction [Jaggi *et al.* 05].

Intrasession Network Coding

Network coding within a single multicast session $(s, \{d_j\})$.

- The capacity of intrasession network coding is governed by min-cut/max-flow theorem. [Ahlswede *et al.* 00], [Li *et al.* 03], and [Koetter *et al.* 03].
- Capacity-achieving random linear network code [Ho *et al.* 06] and the associated polynomial time construction [Jaggi *et al.* 05].
- Both the capacity characterization and code construction have been generalized from acyclic to cyclic networks [Erez *et al.* 05].

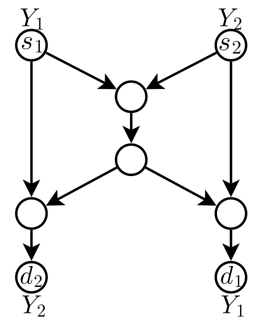
Intrasession Network Coding

Network coding within a single multicast session $(s, \{d_j\})$.

- The capacity of intrasession network coding is governed by min-cut/max-flow theorem. [Ahlswede *et al*. 00], [Li *et al*. 03], and [Koetter *et al*. 03].
- Capacity-achieving random linear network code [Ho *et al*. 06] and the associated polynomial time construction [Jaggi *et al*. 05].
- Both the capacity characterization and code construction have been generalized from acyclic to cyclic networks [Erez *et al*. 05].
- Linear block and convolutional codes [Lu 07].
- Many more ...

Intersession Network Coding

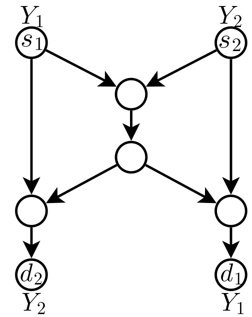
Coding across multiple multicast sessions $(s_i, \{d_{i,j}\})$.



- Intersession network codes are hard to construct and analyzed:

Intersession Network Coding

Coding across multiple multicast sessions $(s_i, \{d_{i,j}\})$.



- Intersession network codes are hard to construct and analyzed:
  - Linear codes are insufficient to achieve the capacity.
  - Feasibility of NC is alphabet-dependent $\equiv$ solving a multivariable polynomial equation. Determining the linear NC feasibility generally requires exhaustive search of the roots of the polynomial.
  - Constructing a good NC is NP-complete w.r.t. # of coexisting sessions.
  - The capacity of network coding is highly non-approximable.
  - Info-theoretic char. involves the fundamental regions of the entropy function, and sometimes the non-Shannon inequalities.
  - References: [Koetter *et al.* 03], [Dougherty *et al.* 05, 07, 08], [Lehman 05], [Yao *et al.* 09], and [Yan *et al.* 07], [Subramanian *et al.* 08], etc.

Intersession Network Coding

An open problem: The graph-theoretic characterization of the existence of intersession NCs for general traffic demand and general network topology.

Intersession Network Coding

An open problem: The graph-theoretic characterization of the existence of intersession NCs for general traffic demand and general network topology.

- Capacity is known only for some special topologies: directed cycles [Harvey 06], degree-2 three-layer directed acyclic networks [Yan *et al.* 06].
- Capacity remains unknown for general topologies:

Intersession Network Coding

An open problem: The graph-theoretic characterization of the existence of intersession NCs for general traffic demand and general network topology.

- Capacity is known only for some special topologies: directed cycles [Harvey 06], degree-2 three-layer directed acyclic networks [Yan *et al.* 06].
- Capacity remains unknown for general topologies:
  - Cut-based outer bounds: Network sharing bound [Yan *et al.* 06], Information dominance condition [Harvey *et al.* 06] and [Kramer *et al.* 06].

Intersession Network Coding

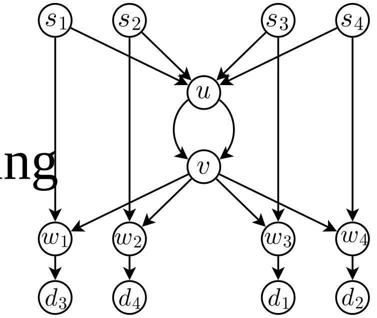
An open problem: The graph-theoretic characterization of the existence of intersession NCs for general traffic demand and general network topology.

- Capacity is known only for some special topologies: directed cycles [Harvey 06], degree-2 three-layer directed acyclic networks [Yan *et al.* 06].
- Capacity remains unknown for general topologies:
  - Cut-based outer bounds: Network sharing bound [Yan *et al.* 06], Information dominance condition [Harvey *et al.* 06] and [Kramer *et al.* 06].
  - In practice, it is critical to study the achievable rates, the capacity inner bound.

Capacity Inner Bounds

Achievable rates for general topology:

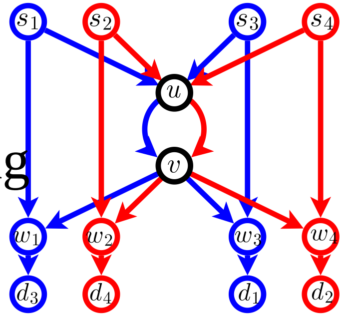
- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.



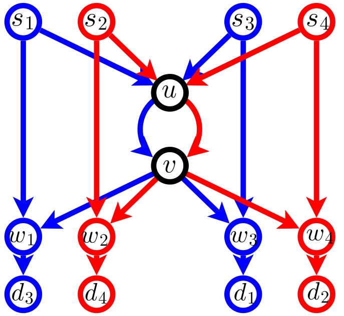
Capacity Inner Bounds

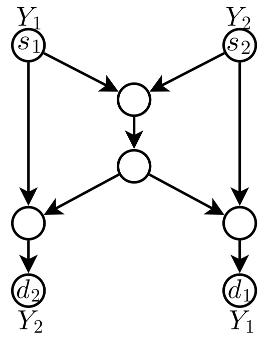
Achievable rates for general topology:

- Superposition of butterflies by linear programming (LP) [Traskov *et al*. 06], etc.



Capacity Inner Bounds





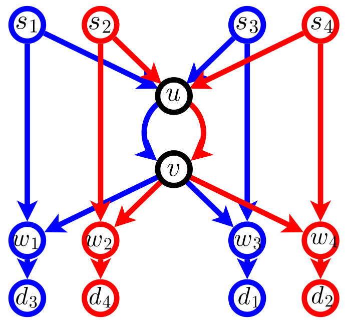
Achievable rates for general topology:

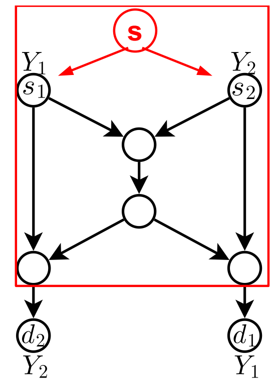
- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.
- Decoding the broadcast traffic (the min-cut/max-flow thm) & then re-encoding [Cui *et al.* 08] and [Wu 06].

Capacity Inner Bounds

Achievable rates for general topology:

- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.
- Decoding the broadcast traffic (the min-cut/max-flow thm) & then re-encoding [Cui *et al.* 08] and [Wu 06].

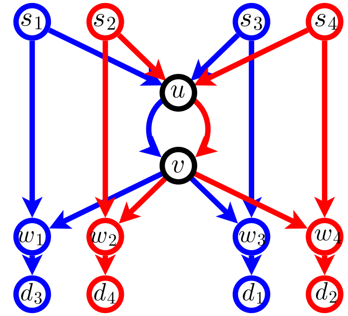


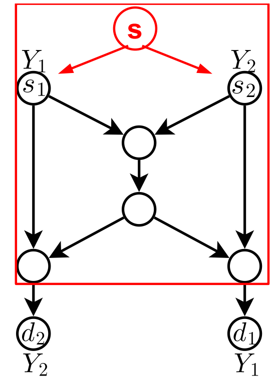


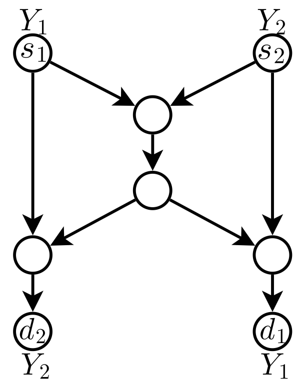
Capacity Inner Bounds

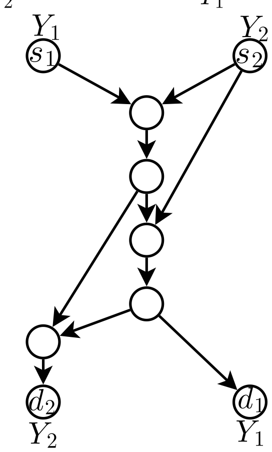
Achievable rates for general topology:

- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.
- Decoding the broadcast traffic (the min-cut/max-flow thm) & then re-encoding [Cui *et al.* 08] and [Wu 06].
- When limiting to two multicast symbols, the graph-theoretic capacity characterization is known [Wang *et al.* 07]. + LP superposition.





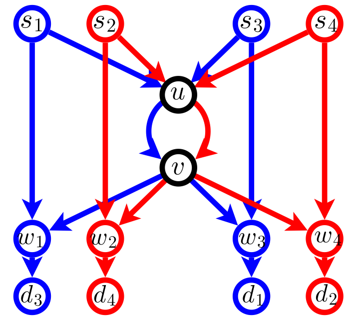


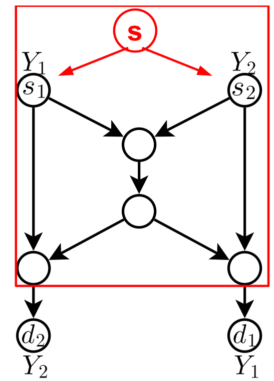


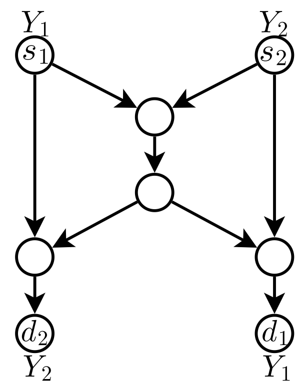
Capacity Inner Bounds

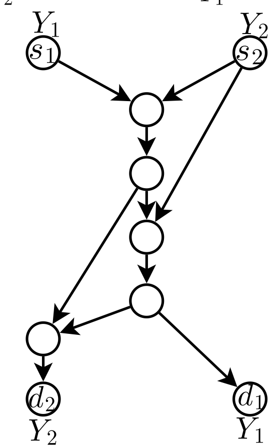
Achievable rates for general topology:

- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.
- Decoding the broadcast traffic (the min-cut/max-flow thm) & then re-encoding [Cui *et al.* 08] and [Wu 06].
- When limiting to two multicast symbols, the graph-theoretic capacity characterization is known [Wang *et al.* 07]. + LP superposition.









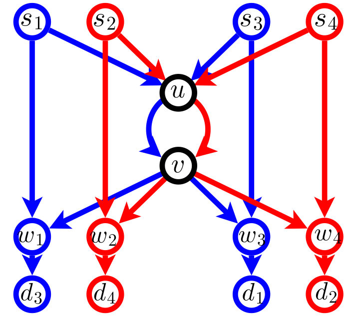
The above inner bounds share many similarities:

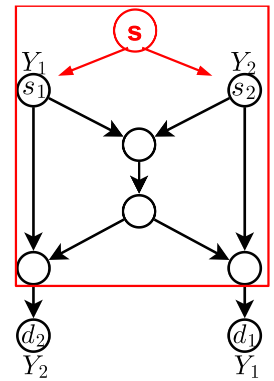
- Linear network coding (LNC).

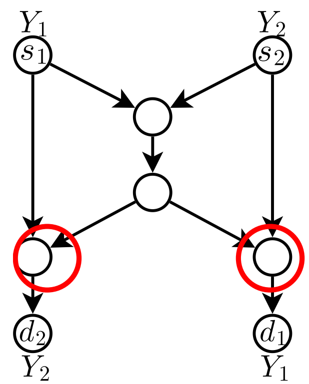
Capacity Inner Bounds

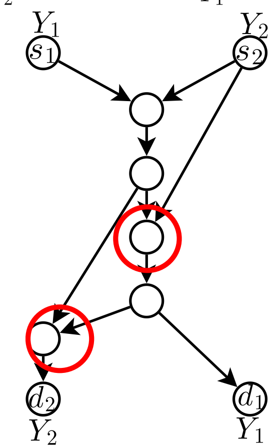
Achievable rates for general topology:

- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.
- Decoding the broadcast traffic (the min-cut/max-flow thm) & then re-encoding [Cui *et al.* 08] and [Wu 06].
- When limiting to two multicast symbols, the graph-theoretic capacity characterization is known [Wang *et al.* 07]. + LP superposition.









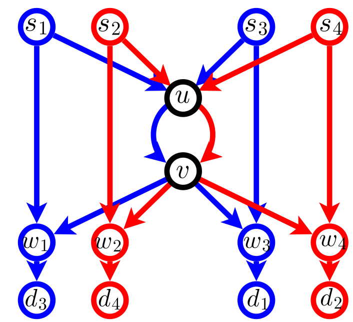
The above inner bounds share many similarities:

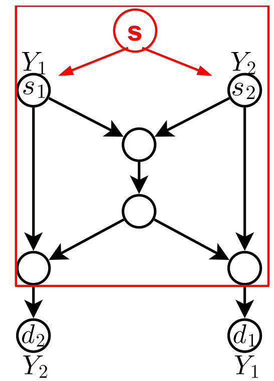
- Random linear network coding (RLNC) for most part of the network. Special operation for a limited portion of the network.

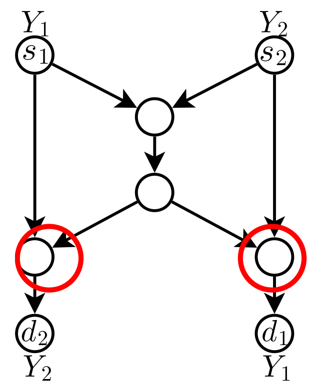
Capacity Inner Bounds

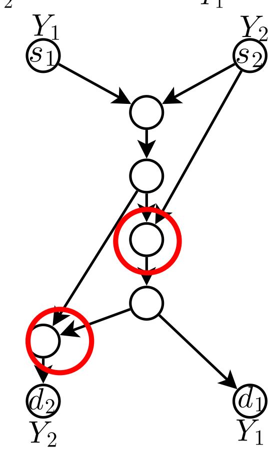
Achievable rates for general topology:

- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.
- Decoding the broadcast traffic (the min-cut/max-flow thm) & then re-encoding [Cui *et al.* 08] and [Wu 06].
- When limiting to two multicast symbols, the graph-theoretic capacity characterization is known [Wang *et al.* 07]. + LP superposition.







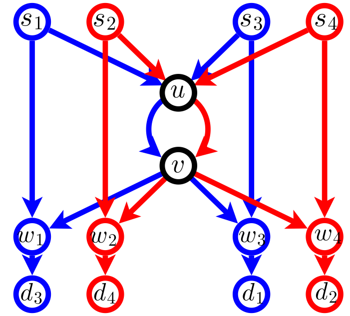


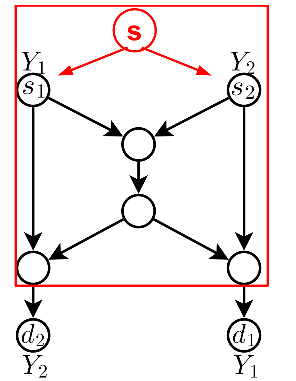
The above inner bounds share many similarities:

- Random linear network coding (RLNC) for most part of the network. Special operation for a limited portion of the network.
- This thus promises better practicality. [Eryilmaz *et al.* 07], [Cui *et al.* 08], and [Khreishah *et al.* 09].

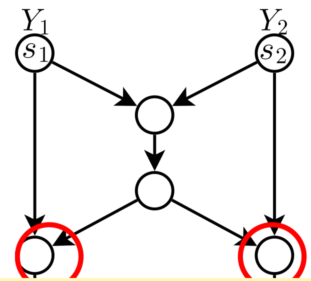
Capacity Inner Bounds

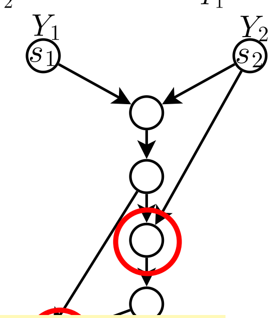
Achievable rates for general topology:





- Superposition of butterflies by linear programming (LP) [Traskov *et al.* 06], etc.
- Decoding the broadcast traffic (the min-cut/max-flow thm) & then re-encoding [Cui *et al.* 08] and [Wu 06].
- When limiting to two multicast symbols, the graph-theoretic capacity characterization is known [Wang *et al.* 07]. + LP superposition.





Can we generalize them for a larger achievable rate region?

- Random linear network coding (RLNC) for most part of the network. Special operation for a limited portion of the network.
- This thus promises better practicality. [Eryilmaz *et al.* 07], [Cui *et al.* 08], and [Khreishah *et al.* 09].

Settings and notation

- Directed acyclic graphs $G = (V, E)$ with unit edge capacity.
- Traffic demands: Multiple multicast sessions $(s_i, \mathbf{d}_i)$, $\mathbf{d}_i = \{d_{i,j} : \forall j\}$.
- Each destination of session i requests a single symbol Y_i.
- $\mathrm{In}(v)$: All incoming edges (u, v).
- M_e: The global coding vector on edge e.
- The s_i-coordinate of M_e means the coefficient of Y_i.
- For each $e = (u, v)$, we define the local kernel $x_{w,e}$:

$$M_e = \sum_{(w,u) \in \mathrm{In}(u)} x_{w,e} M_{w,u}. \tag{1}$$

Encoding of RLNC w. SC

- Each RLNC w. SC scheme is associated with an edge subset $E_K = \{e_1, \cdots, e_K\} \subsetneq E$, termed the Cancelling Edge Set.
  - Each edge $e_k \in E_K$ is associated with a source s_{e_k}.

Encoding of RLNC w. SC

- Each RLNC w. SC scheme is associated with an edge subset $E_K = \{e_1, \cdots, e_K\} \subsetneq E$, termed the Cancelling Edge Set.
  - Each edge $e_k \in E_K$ is associated with a source s_{e_k}.
- Random linear network coding on most of the network.
  - $\forall e = (u, v) \in E \backslash E_K$, perform RLNC:

$$M_e = \sum_{(w,u) \in \mathrm{In}(u)} x_{w,e} M_{w,u}, \tag{2}$$

with $x_{w,e}$ randomly chosen from $\mathsf{GF}(q)$.

Encoding of RLNC w. SC

- Each RLNC w. SC scheme is associated with an edge subset $E_K = \{e_1, \cdots, e_K\} \subsetneq E$, termed the Cancelling Edge Set.
  - Each edge $e_k \in E_K$ is associated with a source s_{e_k}.
- Random linear network coding on most of the network.
  - $\forall e = (u, v) \in E \backslash E_K$, perform RLNC:

$$M_e = \sum_{(w,u) \in \mathrm{In}(u)} x_{w,e} M_{w,u}, \tag{3}$$

    with $x_{w,e}$ randomly chosen from $\mathsf{GF}(q)$.
- Selective cancelling only on E_K.
  - $\forall e_k \in E_K$, the local kernels x_{w,e_k} are chosen such that M_{e_k} has zero s_{e_k}-coordinate.

    Namely, the s_{e_k}-coordinate of M_{e_k} is cancelled.

Remarks on Cancelling

- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.

Remarks on Cancelling

- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.
- If more than one set of $\{x_{w,e_k} : \forall w\}$ can cancel s_{e_k}, choose $\{x_{w,e_k}\}$ uniformly randomly.

Remarks on Cancelling

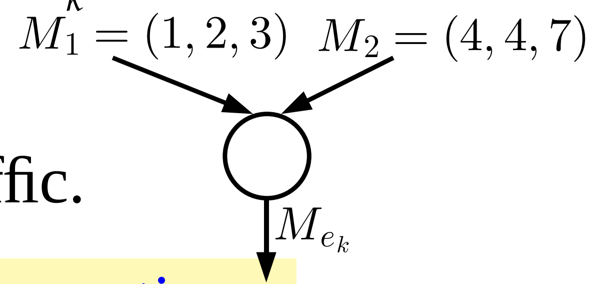
- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.
- If more than one set of $\{x_{w,e_k} : \forall w\}$ can cancel s_{e_k}, choose $\{x_{w,e_k}\}$ uniformly randomly.
- After cancelling, M_{e_k} is remixed with other traffic.

Remarks on Cancelling

- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.
- If more than one set of $\{x_{w,e_k} : \forall w\}$ can cancel s_{e_k}, choose $\{x_{w,e_k}\}$ uniformly randomly.
- After cancelling, M_{e_k} is remixed with other traffic.
- Both RLNC and selective cancelling are local operations.

Remarks on Cancelling

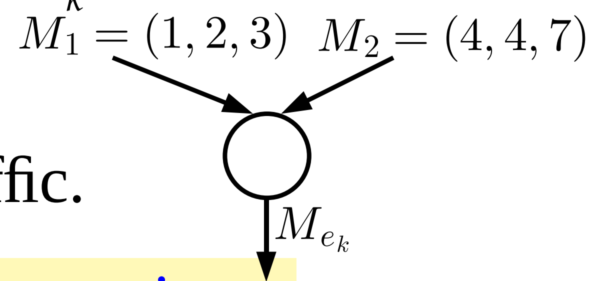
- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.
- If more than one set of $\{x_{w,e_k} : \forall w\}$ can cancel s_{e_k}, choose $\{x_{w,e_k}\}$ uniformly randomly.
- After cancelling, M_{e_k} is remixed with other traffic.



- Both RLNC and selective cancelling are local operations.
- Cancelling $\neq$ blocking: Ex: $M_1 = (1,2,3)$ and $M_2 = (4,4,7)$. Blocking does not cancel the s_1-coordinate, but $4M_1 - M_2$ does.

Remarks on Cancelling

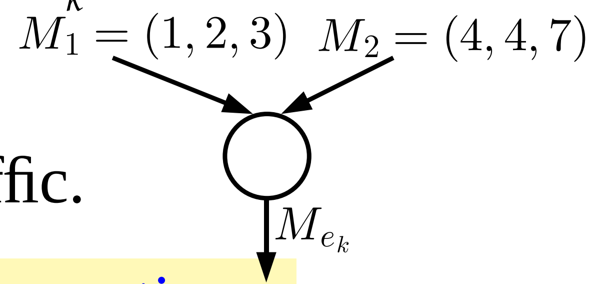
- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.
- If more than one set of $\{x_{w,e_k} : \forall w\}$ can cancel s_{e_k}, choose $\{x_{w,e_k}\}$ uniformly randomly.



- After cancelling, M_{e_k} is remixed with other traffic.
- Both RLNC and selective cancelling are local operations.
- Cancelling $\neq$ blocking: Ex: $M_1 = (1,2,3)$ and $M_2 = (4,4,7)$. Blocking does not cancel the s_1-coordinate, but $4M_1 - M_2$ does.
- Cancelling $\neq$ decoding: Ex: $M_1 = (1,2,3)$ and $M_2 = (4,4,7)$. No decoding is possible.

Remarks on Cancelling

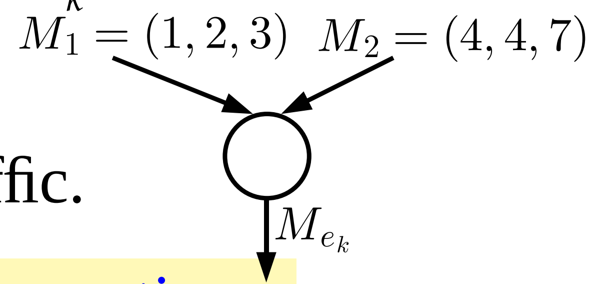
- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.
- If more than one set of $\{x_{w,e_k} : \forall w\}$ can cancel s_{e_k}, choose $\{x_{w,e_k}\}$ uniformly randomly.
- After cancelling, M_{e_k} is remixed with other traffic.



- Both RLNC and selective cancelling are local operations.
- Cancelling includes blocking and decoding as special cases, as long as we choose E_K properly and allow successive cancelling by adding auxiliary edges in the middle of a node.

Remarks on Cancelling

- Cancelling is always feasible as we can choose $x_{w,e_k} = 0$ for all w, $\Rightarrow M_{e_k} = 0$.
- If more than one set of $\{x_{w,e_k} : \forall w\}$ can cancel s_{e_k}, choose $\{x_{w,e_k}\}$ uniformly randomly.
- After cancelling, M_{e_k} is remixed with other traffic.
- Both RLNC and selective cancelling are local operations.
- Cancelling includes blocking and decoding as special cases, as long as we choose E_K properly and allow successive cancelling by adding auxiliary edges in the middle of a node.
  - Blocking is equivalent to cancelling all coordinates.
  - Decoding s_i is equivalent to cancelling all other s_j-coordinates, $j \neq i$.



- RLNC w. SC is a natural extension of RLNC (with $E_K = \emptyset$).
- Very simple operations once the cancelling edge set E_K is chosen.

- RLNC w. SC is a natural extension of RLNC (with $E_K = \emptyset$).
- Very simple operations once the cancelling edge set E_K is chosen.
- How to design a good E_K is a non-trivial problem.

- RLNC w. SC is a natural extension of RLNC (with $E_K = \emptyset$).
- Very simple operations once the cancelling edge set E_K is chosen.
- How to design a good E_K is a non-trivial problem.
- This paper does not fully address this problem.

- RLNC w. SC is a natural extension of RLNC (with $E_K = \emptyset$).
- Very simple operations once the cancelling edge set E_K is chosen.
- How to design a good E_K is a non-trivial problem.
- This paper does not fully address this problem.
- Instead, we focus on the question

  What is a good choice of E_K?

  and provide a sufficient condition of good E_K, which includes all existing results of superposition, decoding-remixing, and simple pairwise intersession network coding as special cases.
- Future work: Use the sufficient condition to actively search for good E_K.

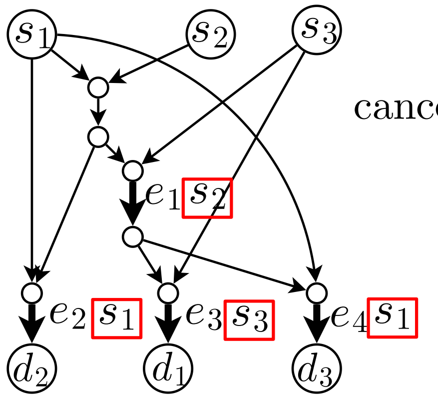
Our goal is to identify a graph-theoretic sufficient condition for the general multiple multicast problem such that for all i, all $d_{i,j} \in \mathbf{d}_i$ are able to recover Υ_i with close-to-one probability using the RLNC w. SC scheme when a sufficiently large GF(q) is used. In this case, we say that the E_K of RLNC w. SC is good.

The First Try

In a multiple-multicast setting, a good E_K should shield destinations $\mathbf{d}_i = \{d_{i,j}\}$ from all interfering sources $s'_i \neq s_i$.

Condition 1 (Interference-shielding condition) *$\forall i, d_{i,j} \in \mathbf{d}_i$, and $\forall s'_i \neq s_i$, $\{e \in E_K : s_e = s'_i\}$ is an* **edge cut** *separating s'_i and $d_{i,j}$.*

The First Try

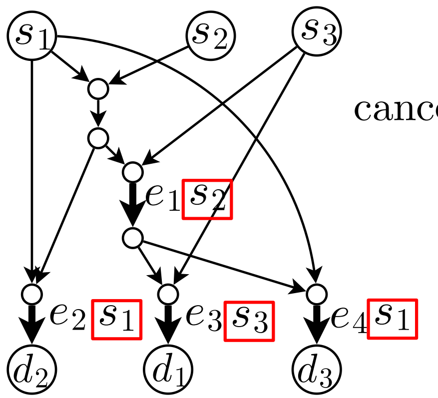


canceling edges :

$$s_{e_1} = s_2$$
$$s_{e_2} = s_1$$
$$s_{e_3} = s_3$$
$$s_{e_4} = s_1$$

Condition 1 (Interference-shielding condition) $\forall i, d_{i,j} \in \mathbf{d}_i$, *and* $\forall s'_i \neq s_i$, $\{e \in E_K : s_e = s'_i\}$ *is an* **edge cut** *separating* s'_i *and* $d_{i,j}$.

The First Try



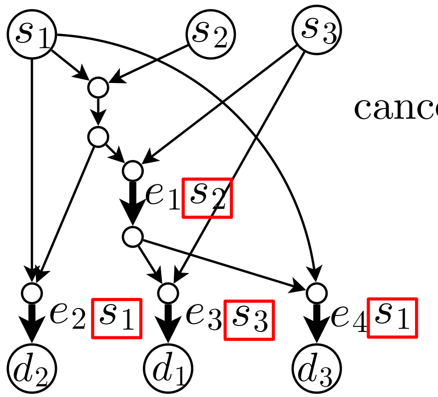
canceling edges :
$s_{e_1} = s_2$
$s_{e_2} = s_1$
$s_{e_3} = s_3$
$s_{e_4} = s_1$

Condition 1 (Interference-shielding condition) *$\forall i, d_{i,j} \in \mathbf{d}_i$, and $\forall s'_i \neq s_i$, $\{e \in E_K : s_e = s'_i\}$ is an* **edge cut** *separating s'_i and $d_{i,j}$.*

Following the wisdom of RLNC, one possible sufficient condition is

Condition 2 (Per-destination rank condition) *For each $d_{i,j}$, there exists a local kernel $\{x_{w,e}\}$ that guarantees the desired rank for $d_{i,j}$ while the resulting M_e comply with all cancelling edges.*

The First Try



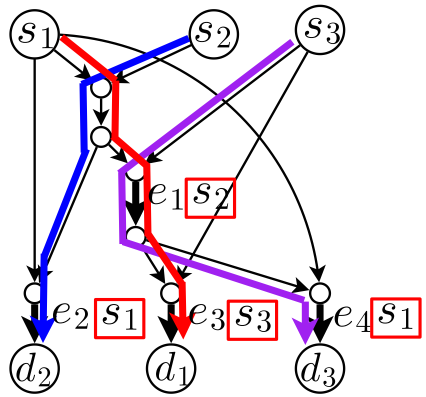
canceling edges :

$s_{e_1} = s_2$

$s_{e_2} = s_1$

$s_{e_3} = s_3$

$s_{e_4} = s_1$

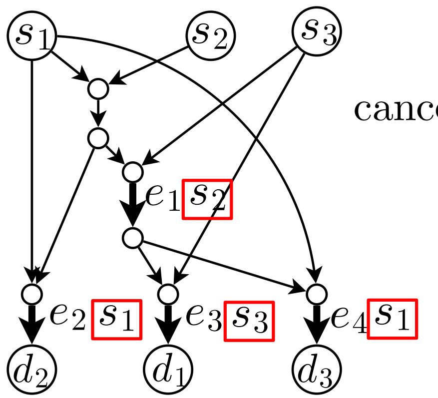


Condition 1 (Interference-shielding condition) *$\forall i, d_{i,j} \in \mathbf{d}_i$, and $\forall s'_i \neq s_i$, $\{e \in E_K : s_e = s'_i\}$ is an* **edge cut** *separating s'_i and $d_{i,j}$.*

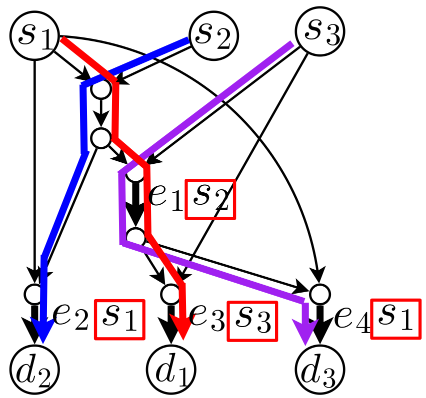
Following the wisdom of RLNC, one possible sufficient condition is

Condition 2 (Per-destination rank condition) *For each $d_{i,j}$, there exists a local kernel $\{x_{w,e}\}$ that guarantees the desired rank for $d_{i,j}$ while the resulting M_e comply with all cancelling edges.*

The First Try



canceling edges :
$s_{e_1} = s_2$
$s_{e_2} = s_1$
$s_{e_3} = s_3$
$s_{e_4} = s_1$



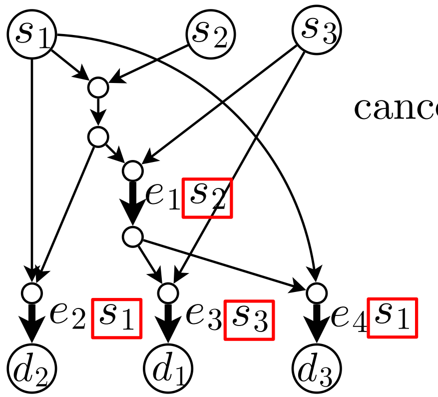
Condition 1 (Interference-shielding condition) $\forall i, d_{i,j} \in \mathbf{d}_i$, *and* $\forall s_i' \neq s_i$, $\{e \in E_K : s_e = s_i'\}$ *is an* **edge cut** *separating* s_i' *and* $d_{i,j}$.

Following the wisdom of RLNC, one possible sufficient condition is

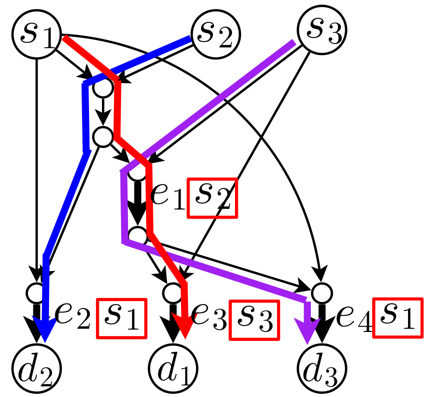
Condition 2 (Per-destination rank condition) *For each* $d_{i,j}$, *there exists a local kernel* $\{x_{w,e}\}$ *that guarantees the desired rank for* $d_{i,j}$ *while the resulting* M_e *comply with all cancelling edges.*

Unfortunately, these two conditions alone do not guarantee the existence of a good RLNC w. SC scheme.

The First Try



canceling edges :
$s_{e_1} = s_2$
$s_{e_2} = s_1$
$s_{e_3} = s_3$
$s_{e_4} = s_1$



The cancelling edges collectively may kill the desired symbols as well.

Condition 1 (Interference-shielding condition) *$\forall i, d_{i,j} \in \mathbf{d}_i$, and $\forall s'_i \neq s_i$, $\{e \in E_K : s_e = s'_i\}$ is an* **edge cut** *separating s'_i and $d_{i,j}$.*

Following the wisdom of RLNC, one possible sufficient condition is

Condition 2 (Per-destination rank condition) *For each $d_{i,j}$, there exists a local kernel $\{x_{w,e}\}$ that guarantees the desired rank for $d_{i,j}$ while the resulting M_e comply with all cancelling edges.*

Unfortunately, these two conditions alone do not guarantee the existence of a good RLNC w. SC scheme.

Revisit Random LNC

Why random LNC works for a single multicast session? [Koetter *et al.* 03] and [Ho *et al.* 06].

- The M_e on each edge is a polynomial of the local kernels $\{x_{w,e}\}$.

$$M_e = \sum_{(w,u)\in \mathrm{In}(u)} x_{w,e} M_{w,u}. \quad (4)$$

- The objective function (determinant) is a polynomial of $\{x_{w,e}\}$.
- The degree of the polynomial is bounded above by a function of the network structure.
  $\Rightarrow$ If the polynomial is not zero, the number of roots is upper bounded.
  $\Rightarrow$ With a sufficiently large $\mathsf{GF}(q)$, a random choice of $\{x_{w,e}\}$ achieves a non-zero objective value with close-to-1 probability.

Hurdles for RLNC w. SC

- M_e is no longer a polynomial, as $\mathbf{x} = \{x_{w,e_k}\}$ is chosen based on the input coding vectors $M_{w,\mathsf{tail}(e_k)}$. RLNC w. SC is reactive.

Hurdles for RLNC w. SC

- M_e is no longer a polynomial, as $\mathbf{x} = \{x_{w,e_k}\}$ is chosen based on the input coding vectors $M_{w,\mathsf{tail}(e_k)}$. RLNC w. SC is reactive.
- Two modes of encoding on $e_k = (u, v)$:
  - Trivial cancelling: All incoming $M_{w,u}$ have zero s_{e_k}-coordinate. $|\mathrm{In}(u)|$ degrees of freedom.
  - Non-trivial cancelling: One incoming $M_{w,u}$ has non-zero s_{e_k}-coordinate. $|\mathrm{In}(u)| - 1$ degrees of freedom.

Hurdles for RLNC w. SC

- M_e is no longer a polynomial, as $\mathbf{x} = \{x_{w,e_k}\}$ is chosen based on the input coding vectors $M_{w,\mathsf{tail}(e_k)}$. RLNC w. SC is reactive.
- Two modes of encoding on $e_k = (u, v)$:
  - Trivial cancelling: All incoming $M_{w,u}$ have zero s_{e_k}-coordinate. $|\mathrm{In}(u)|$ degrees of freedom.
  - Non-trivial cancelling: One incoming $M_{w,u}$ has non-zero s_{e_k}-coordinate. $|\mathrm{In}(u)| - 1$ degrees of freedom.

Lemma 1 *There exist $2|E|$ polynomials $F_e(\mathbf{x})$ and $Q_e(\mathbf{x})$ for all $e \in E$, such that for any $\mathbf{x}$ such that all $e_k \in E_K$ are non-trivial cancelling, we can express the resulting M_e as a fractional $\frac{F_e(\mathbf{x})}{Q_e(\mathbf{x})}$. Moreover, if such globally non-trivial cancelling $\mathbf{x}$ exists, all $Q_e(\mathbf{x})$ are non-zero functions.*

Hurdles for RLNC w. SC (Cont'd)

- If $\exists$ a globally non-trivially cancelling $\mathbf{x}$ that conveys Y_i to $d_{i,j}$, $\forall i, j$.

Hurdles for RLNC w. SC (Cont'd)

- If $\exists$ a globally non-trivially cancelling $\mathbf{x}$ that conveys Y_i to $d_{i,j}$, $\forall i, j$.
  $\Rightarrow M_e$ are fractional $\frac{F_e(\mathbf{x})}{Q_e(\mathbf{x})}$ + non-zero $Q_d(\mathbf{x})$ and $F_d(\mathbf{x})$

Hurdles for RLNC w. SC (Cont'd)

- If $\exists$ a globally non-trivially cancelling $\mathbf{x}$ that conveys Y_i to $d_{i,j}$, $\forall i, j$.
  $\Rightarrow M_e$ are fractional $\frac{F_e(\mathbf{x})}{Q_e(\mathbf{x})}$ + non-zero $Q_d(\mathbf{x})$ and $F_d(\mathbf{x})$
  $\Rightarrow$ Random LNC w. SC is feasible with ≈ 1 probability.

Hurdles for RLNC w. SC (Cont'd)

- If $\exists$ a globally non-trivially cancelling $\mathbf{x}$ that conveys Y_i to $d_{i,j}$, $\forall i, j$.
  $\Rightarrow M_e$ are fractional $\frac{F_e(\mathbf{x})}{Q_e(\mathbf{x})}$ + non-zero $Q_d(\mathbf{x})$ and $F_d(\mathbf{x})$
  $\Rightarrow$ Random LNC w. SC is feasible with ≈ 1 probability.
- Nonetheless, finding a global solution is hard.

Hurdles for RLNC w. SC (Cont'd)

- If $\exists$ a globally non-trivially cancelling $\mathbf{x}$ that conveys Y_i to $d_{i,j}$, $\forall i, j$.
  $\Rightarrow M_e$ are fractional $\frac{F_e(\mathbf{x})}{Q_e(\mathbf{x})}$ + non-zero $Q_d(\mathbf{x})$ and $F_d(\mathbf{x})$
  $\Rightarrow$ Random LNC w. SC is feasible with ≈ 1 probability.
- Nonetheless, finding a global solution is hard.

Definition 1 $\mathbf{x} = \{x_{w,e}\}$ *is partially non-trivial cancelling if the resulting M_e satisfy: $\forall e_k \in E_K$, $M_{e_k} \neq 0 \Rightarrow e_K$ is non-trivial cancelling.*

Hurdles for RLNC w. SC (Cont'd)

- If $\exists$ a globally non-trivially cancelling $\mathbf{x}$ that conveys Y_i to $d_{i,j}$, $\forall i, j$.
  $\Rightarrow M_e$ are fractional $\frac{F_e(\mathbf{x})}{Q_e(\mathbf{x})}$ + non-zero $Q_d(\mathbf{x})$ and $F_d(\mathbf{x})$
  $\Rightarrow$ Random LNC w. SC is feasible with ≈ 1 probability.
- Nonetheless, finding a global solution is hard.

Definition 1 $\mathbf{x} = \{x_{w,e}\}$ *is partially non-trivial cancelling if the resulting* M_e *satisfy:* $\forall e_k \in E_K, M_{e_k} \neq 0 \Rightarrow e_K$ *is non-trivial cancelling.*

Theorem 1 *If for each destination* $d_{i,j}$*, there exists a partially non-trivial cancelling* $\mathbf{x}_{i,j}$ *that conveys* Y_i *to* $d_{i,j}$*, then there exists a (globally*† *non-trivial) cancelling* $\mathbf{x}^*$ *that conveys* Y_i *to* $d_{i,j}$ *for all* i *and* j*. The converse is also true.*

Per-destination characterization

A sufficient condition of a good E_K becomes:

Condition 1 (Interference-shielding condition) $\forall i, d_{i,j} \in \mathbf{d}_i$, *and* $\forall s_i' \neq s_i$, $\{e \in E_K : s_e = s_i'\}$ *is an* **edge cut** *separating* s_i' *and* $d_{i,j}$.

$+$

Condition 2 (Per-destination partially NTC condition) *For each* $d_{i,j}$, *there exists a local kernel* $\{x_{w,e}\}$ *that is partially non-trivial cancelling while conveying* Y_i *to* $d_{i,j}$.

$\Rightarrow$ (1) The existence of a globally non-trivial cancelling solution;
(2) RLNC w. SC meets the traffic demand with ≈ 1 probability.

Per-destination characterization

A sufficient condition of a good E_K becomes:

Condition 1 (Interference-shielding condition) $\forall i, d_{i,j} \in \mathbf{d}_i$, *and* $\forall s_i' \neq s_i$, $\{e \in E_K : s_e = s_i'\}$ *is an* **edge cut** *separating* s_i' *and* $d_{i,j}$.

$$+$$

Condition 2 (Per-destination partially NTC condition) *For each* $d_{i,j}$, *there exists a local kernel* $\{x_{w,e}\}$ *that is partially non-trivial cancelling while conveying* Y_i *to* $d_{i,j}$.

$\Rightarrow$ (1) The existence of a globally non-trivial cancelling solution;
(2) RLNC w. SC meets the traffic demand with ≈ 1 probability.

- We then proceed to convert Condition 2 to a graph-theoretic condition.

A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.

A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.

A Simple Graph Condition

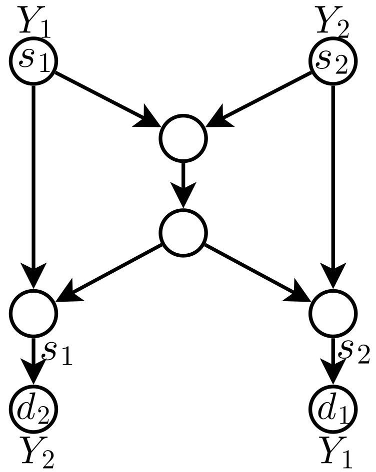
In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.

A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

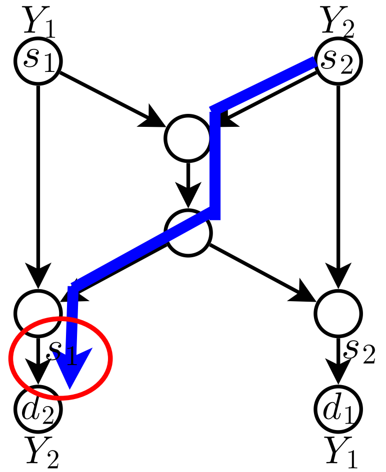
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.



A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

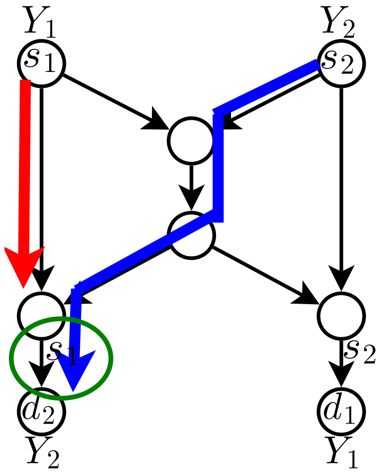
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.



A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

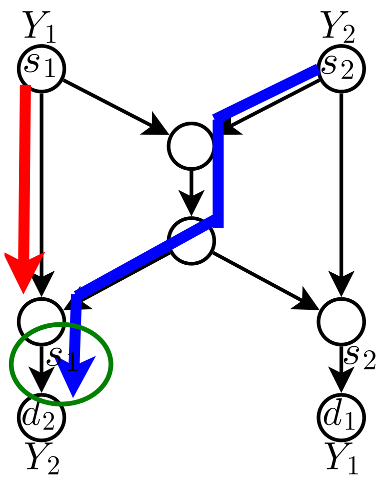
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.

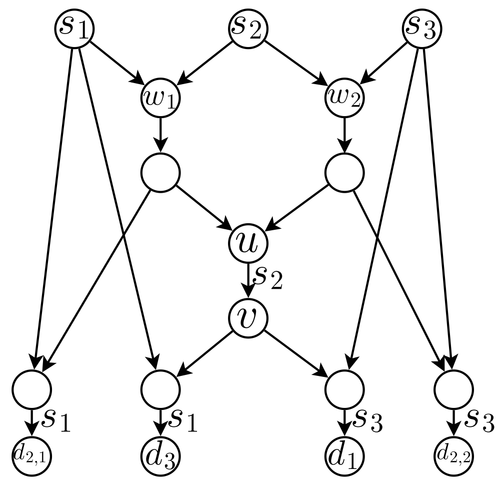


A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.

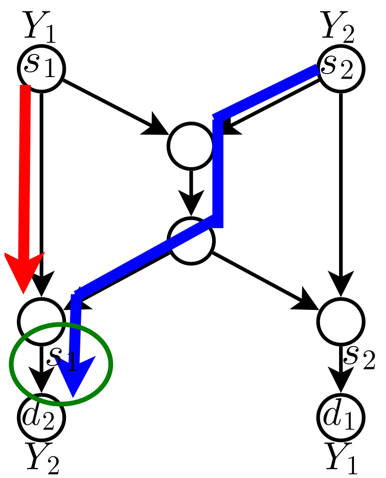


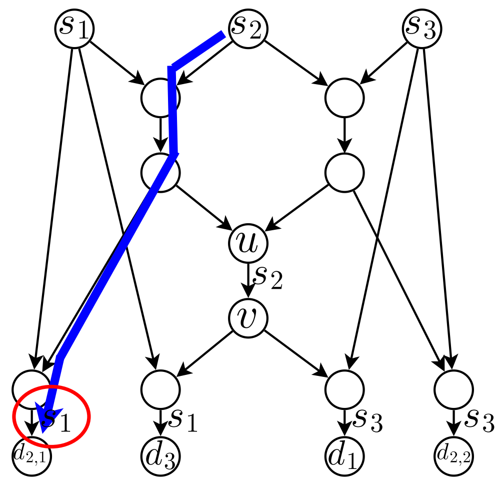


A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.

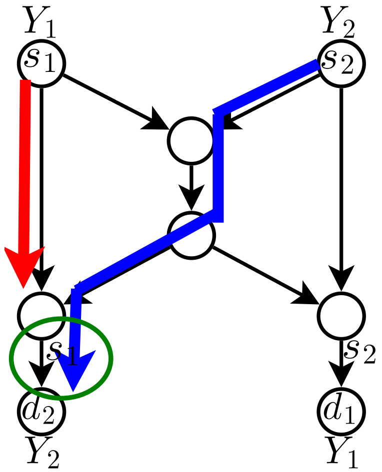


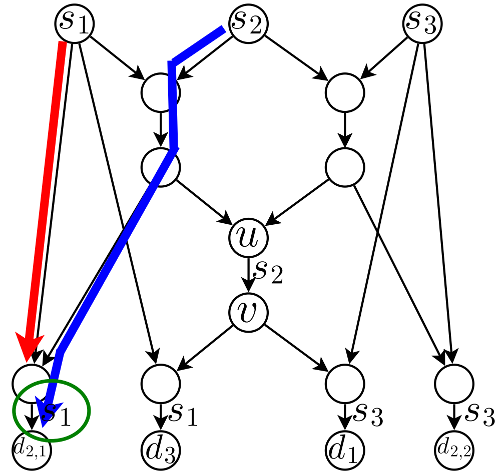


A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.

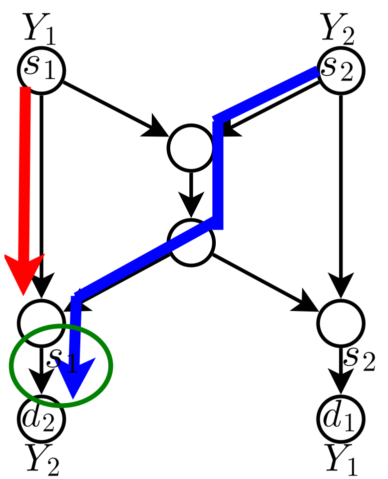


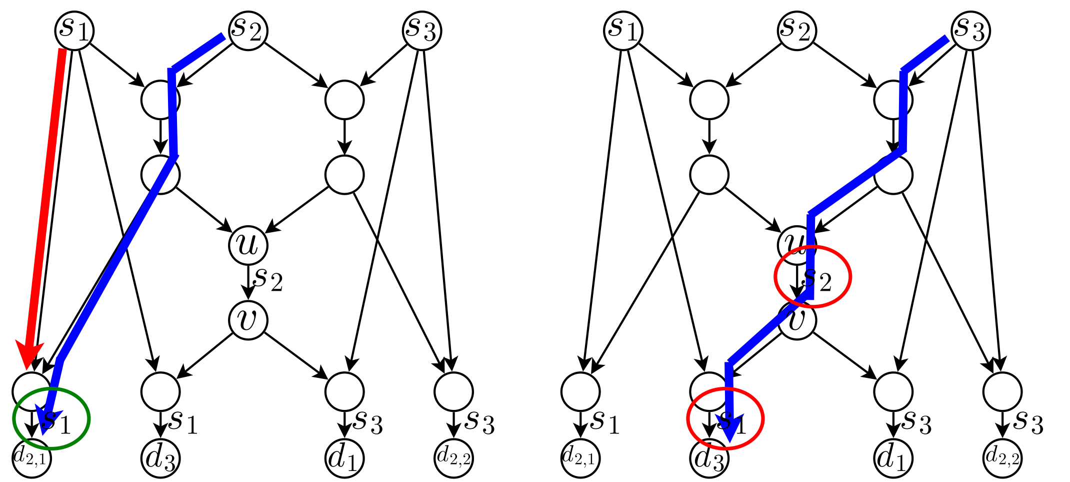


A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.

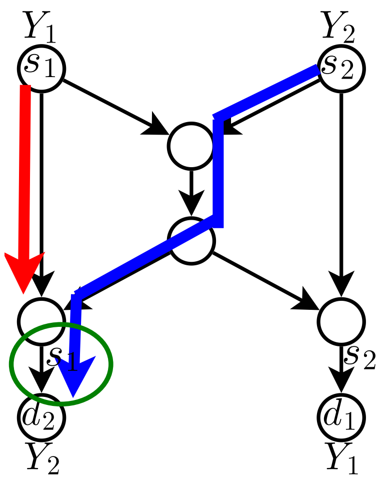


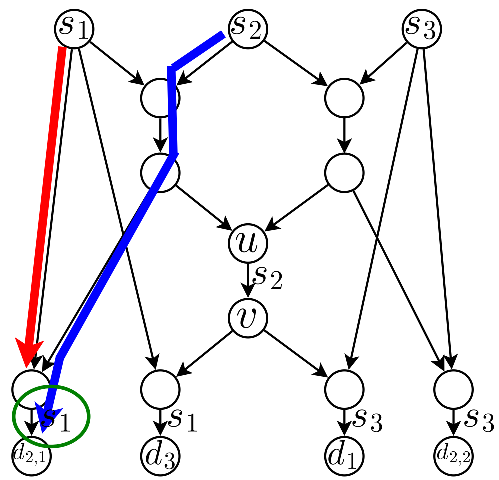


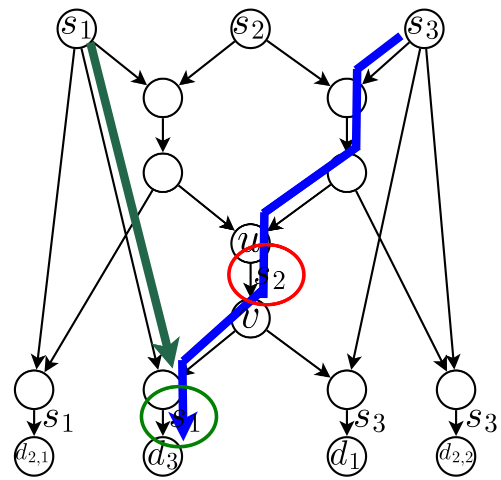
A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.



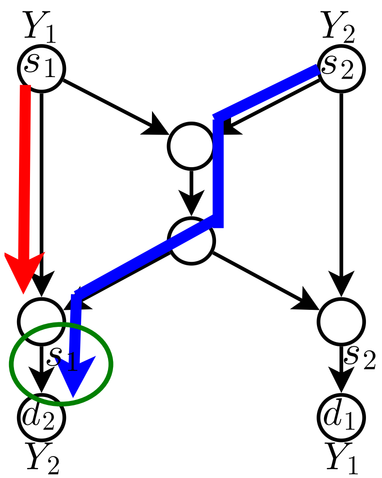


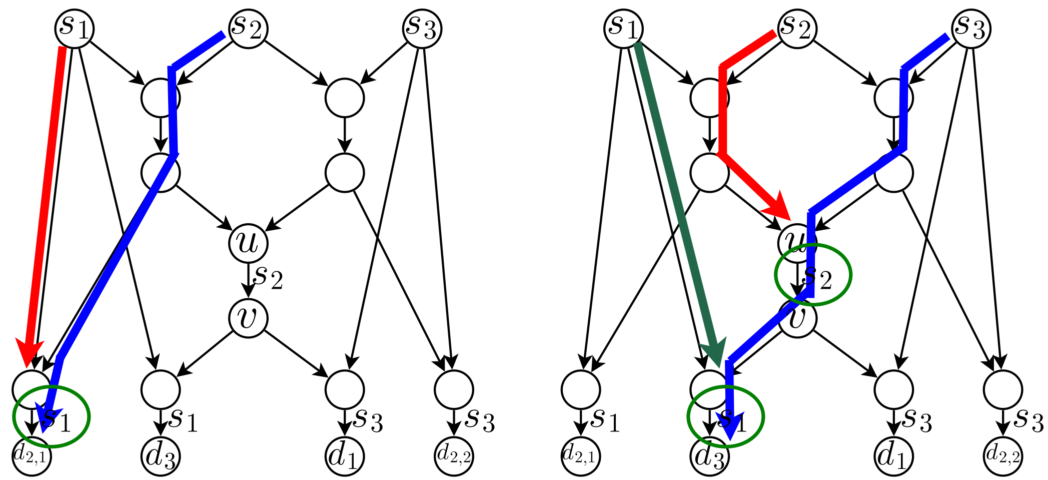


A Simple Graph Condition

In addition to the interference shielding condition, for each $d_{i,j}$, we focus on how to construct a partially non trivial cancelling solution:

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$ $\Rightarrow$ decodability.
- (3.b) The disjoint partially non-trivial cancelling trees.
- (3.c) The iteratively incurred cost of using a cancelling edge.





A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

A Stronger Graph Condition

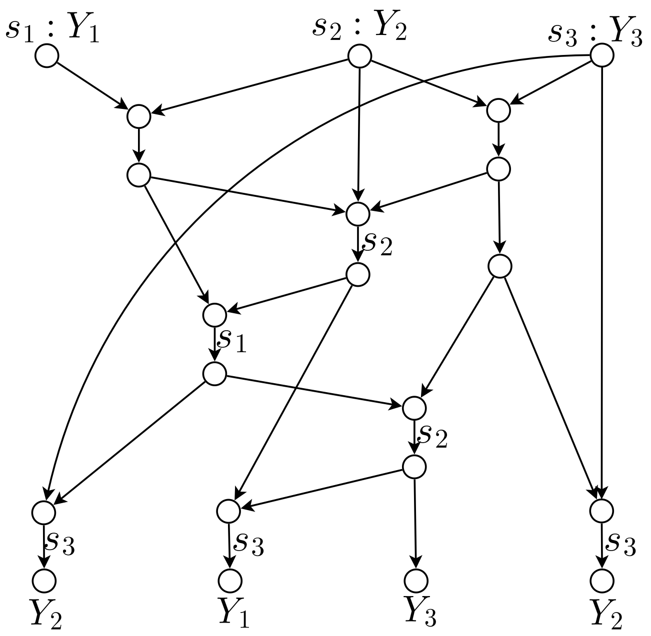
We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.

A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

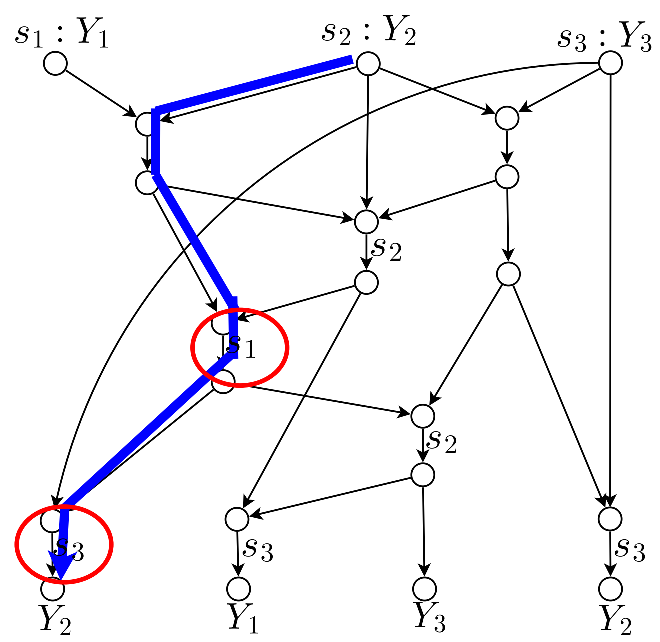
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

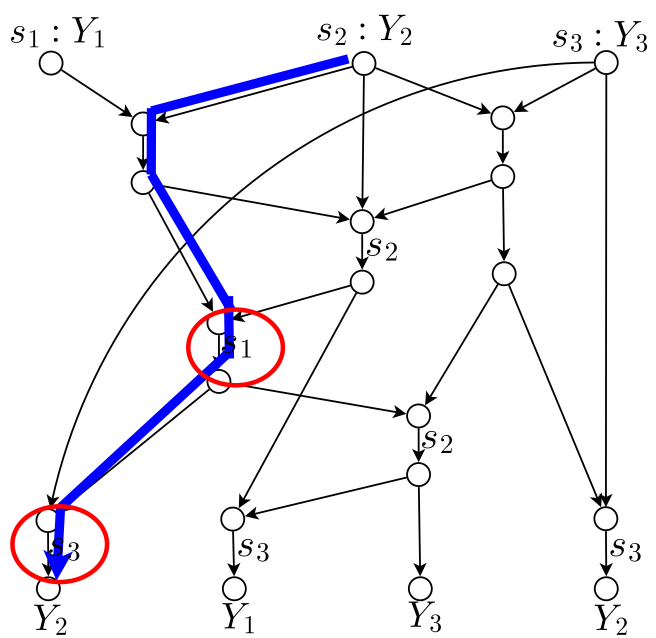
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

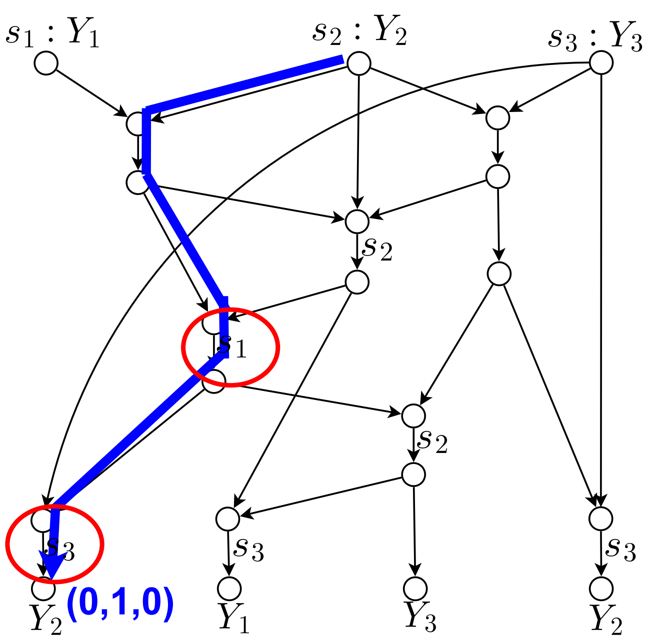
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

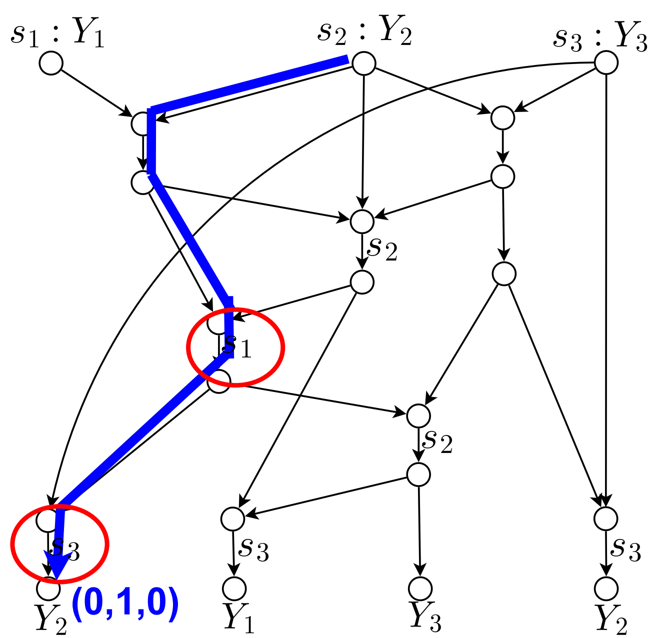
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

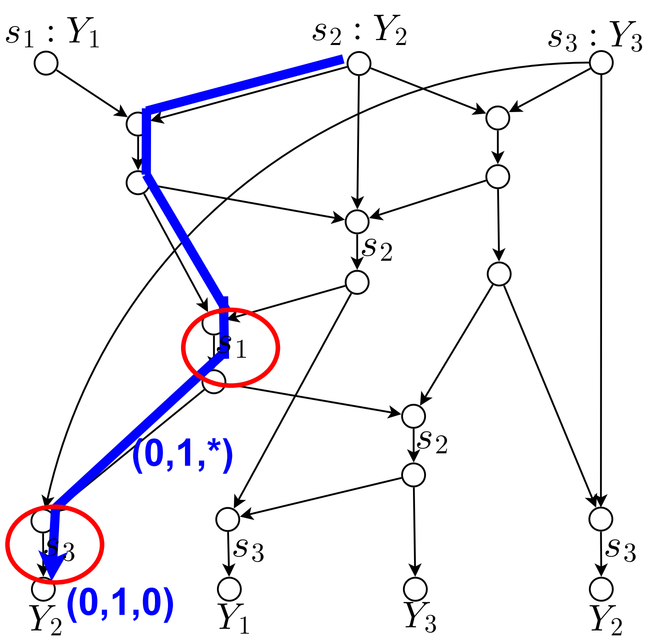
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
    - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
    - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
    - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

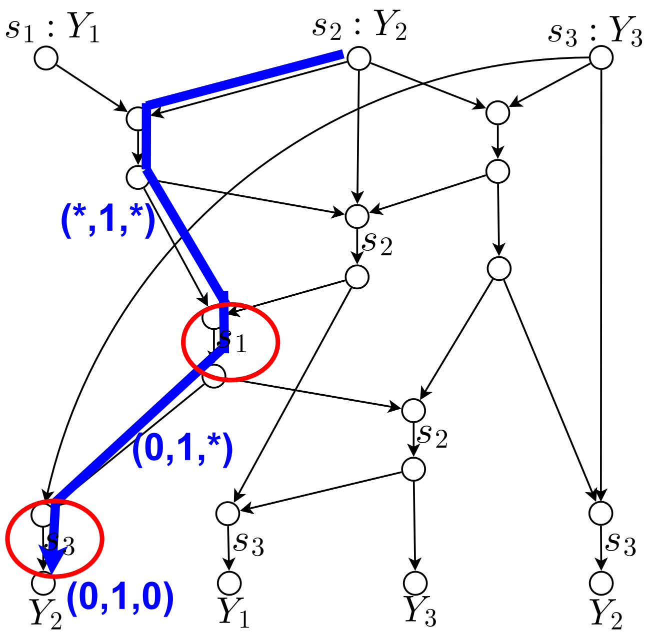
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

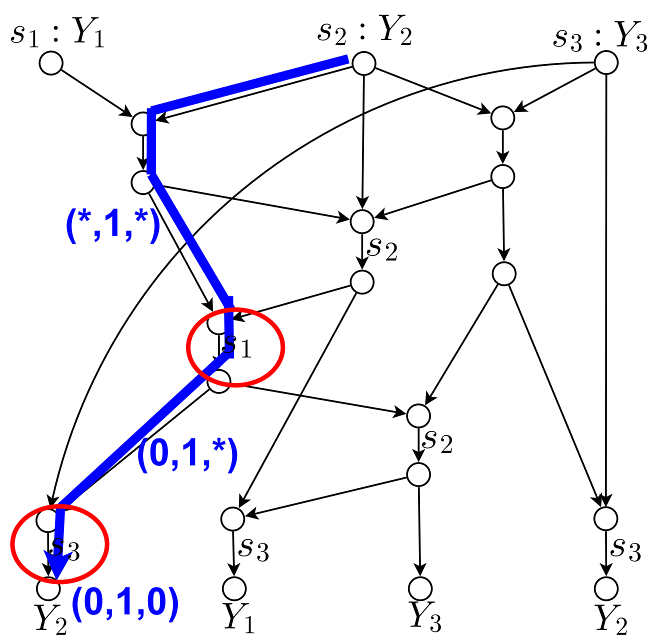
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

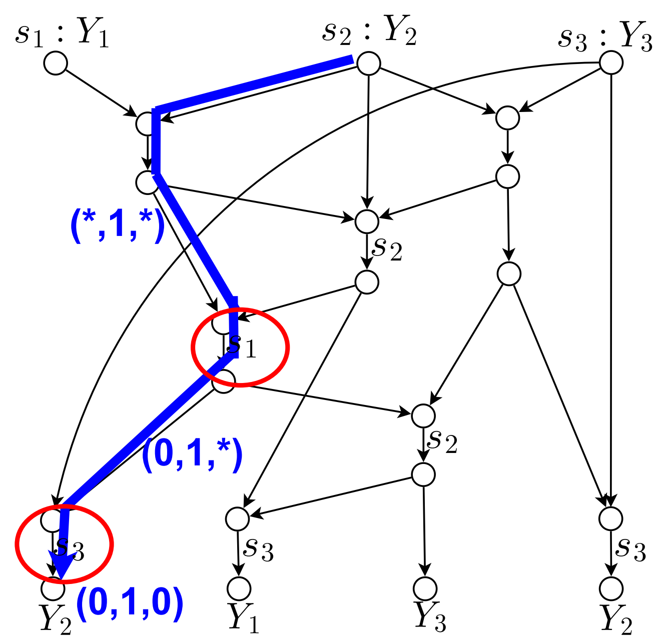
- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."
  - When two branches merge, keep the union of the 0 positions; take the intersection of the * locations.



A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."
  - When two branches merge, keep the union of the 0 positions; take the intersection of the * locations.
  - Two different trees share an edge e only if the constraint vectors on e are compatible with each other.

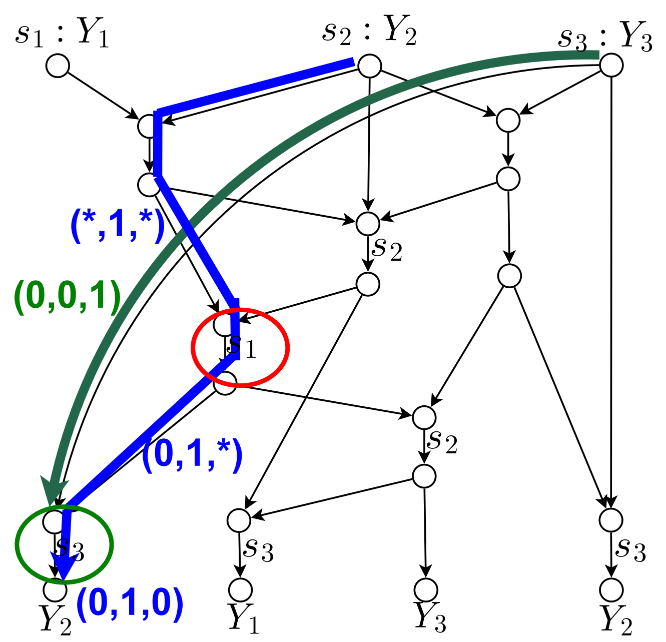


- (3.c) The iteratively incurred cost of using a cancelling edge.

A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."
  - When two branches merge, keep the union of the 0 positions; take the intersection of the * locations.
  - Two different trees share an edge e only if the constraint vectors on e are compatible with each other.

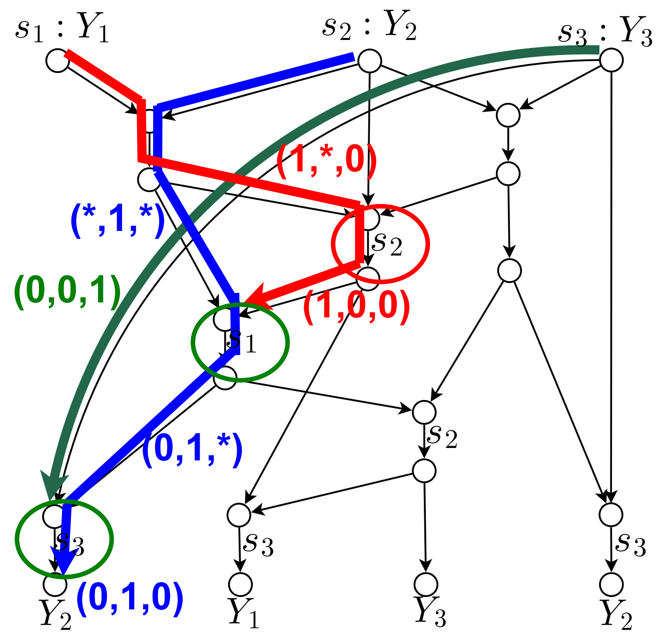


- (3.c) The iteratively incurred cost of using a cancelling edge.

A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."
  - When two branches merge, keep the union of the 0 positions; take the intersection of the * locations.
  - Two different trees share an edge e only if the constraint vectors on e are compatible with each other.

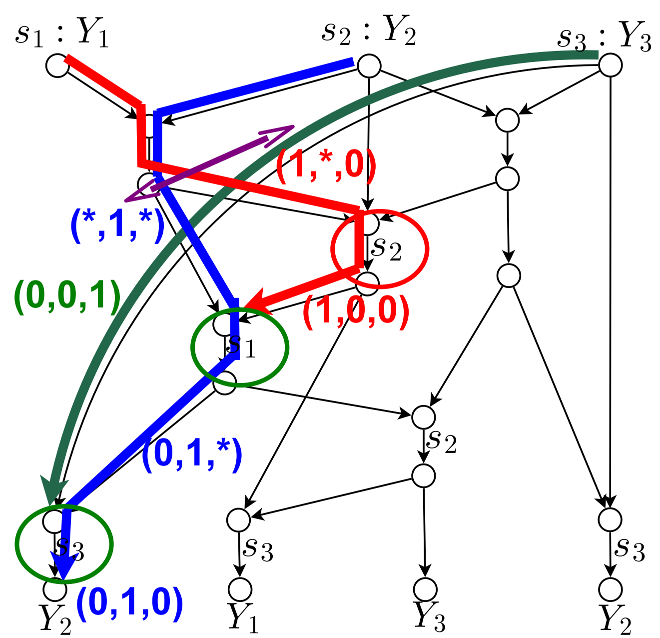


- (3.c) The iteratively incurred cost of using a cancelling edge.

A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."
  - When two branches merge, keep the union of the 0 positions; take the intersection of the * locations.
  - Two different trees share an edge e only if the constraint vectors on e are compatible with each other.

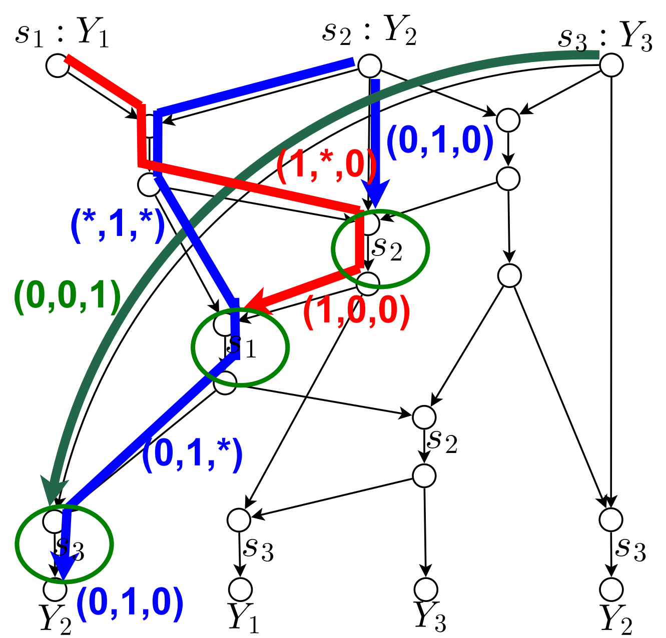


- (3.c) The iteratively incurred cost of using a cancelling edge.

A Stronger Graph Condition

We can further relax the disjointness of the partially-non-trivial cancelling trees. For each individual destination $d_{i,j}$, we have

- (3.a) The existence of a principal information path from s_i to $d_{i,j}$.
- (3.b) The non-disjoint partially-non-trivial cancelling trees:
  - From the leaves e_k, $s_{e_k} = s_i$ back to s_i, each edge of the i-th cancelling tree is assigned a constraint vector (CV).
  - For each leaf e_k w. $s_{e_k} = s_i$, the CV is $(0, 0, \underbrace{1}_{s_i}, 0)$.
  - If a branch has used another cancelling edge $s_{e'} = s'_i$, then the i'-th position of the CV becomes "$*$: don't care."
  - When two branches merge, keep the union of the 0 positions; take the intersection of the * locations.
  - Two different trees share an edge e only if the constraint vectors on e are compatible with each other.



- (3.c) The iteratively incurred cost of using a cancelling edge.

Conclusion

- We propose random linear network coding with selective cancelling.
- Extending RLNC from single multicast session to multiple multicast sessions.
- RLCN w. SC includes blocking, superposition, decoding+remixing, and many existing schemes as special cases.
- Simple coding operation once the cancelling edge set E_K is given.

Conclusion

- We propose random linear network coding with selective cancelling.
- Extending RLNC from single multicast session to multiple multicast sessions.
- RLCN w. SC includes blocking, superposition, decoding+remixing, and many existing schemes as special cases.
- Simple coding operation once the cancelling edge set E_K is given.
- A graph-theoretic sufficient condition of good E_K for general networks, which relies only on per-destination characterization.

Conclusion

- We propose random linear network coding with selective cancelling.
- Extending RLNC from single multicast session to multiple multicast sessions.
- RLCN w. SC includes blocking, superposition, decoding+remixing, and many existing schemes as special cases.
- Simple coding operation once the cancelling edge set E_K is given.
- A graph-theoretic sufficient condition of good E_K for general networks, which relies only on per-destination characterization.
- The sufficient condition is tight for many instances, ex: min-cut/max-flow theorem for single multicast, the two-symbol pairwise intersession NC.

Conclusion

- We propose random linear network coding with selective cancelling.
- Extending RLNC from single multicast session to multiple multicast sessions.
- RLCN w. SC includes blocking, superposition, decoding+remixing, and many existing schemes as special cases.
- Simple coding operation once the cancelling edge set E_K is given.
- A graph-theoretic sufficient condition of good E_K for general networks, which relies only on per-destination characterization.
- The sufficient condition is tight for many instances, ex: min-cut/max-flow theorem for single multicast, the two-symbol pairwise intersession NC.
- Open question: How to use the suff. condition to search for good E_K?

Conclusion

- We propose random linear network coding with selective cancelling.
- Extending RLNC from single multicast session to multiple multicast sessions.
- RLCN w. SC includes blocking, superposition, decoding+remixing, and many existing schemes as special cases.
- Simple coding operation once the cancelling edge set E_K is given.
- A graph-theoretic sufficient condition of good E_K for general networks, which relies only on per-destination characterization.
- The sufficient condition is tight for many instances, ex: min-cut/max-flow theorem for single multicast, the two-symbol pairwise intersession NC.
- Open question: How to use the suff. condition to search for good E_K?
- As a sufficient condition, any E_K satisfying our characterization will guarantee decodability. We can thus strike some complexity and performance tradeoff.