Characterisation of network coding regions via entropy functions

Terence Chan[†], Alex Grant[†]

[†] Institute for Telecommunications Research,
University of South Australia

October 2011

Outline

Problem formulation

Networks

- A network G – a directed hypergraph $(\mathcal{V}, \mathcal{E})$
- $\mathcal{V} = \{V_1, \ldots, V_{|\mathcal{V}|}\}$ – communication nodes
- $\mathcal{E} = \{E_1, \ldots, E_{|\mathcal{E}|}\}$ – error-free “broadcast” links
- Each link $e \in \mathcal{E}$ is a tuple $(\mathit{tail}(e), \mathit{head}(e))$
- $\mathit{tail}(e) \in \mathcal{V}$ is the transmitter node
- $\mathit{head}(e) \subseteq \mathcal{V}$ are nodes which hear what $\mathit{tail}(e)$ transmits
- if $\mathit{head}(e)$ is a singleton, then the link e is ordinary point-to-point link

Connection constraint

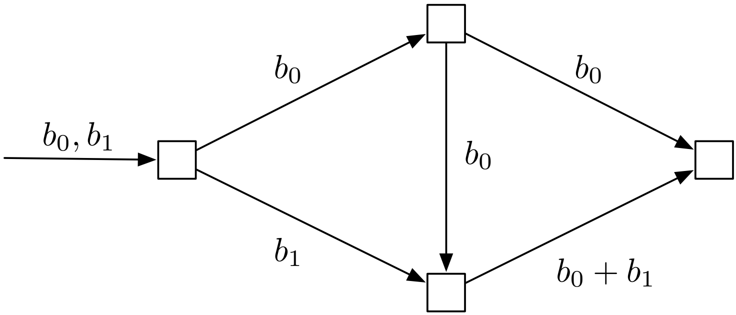
Definition

Connection constraint M *is a tuple* $(\mathcal{S}, O, D)$ *where*

1. $\mathcal{S}$ *– source indices*
2. $O(s)$ *– nodes that access* s^{th} *source*
3. $D(s)$ *– sink nodes ask for* s^{th} *source*

- Network coding problem P defined by (G, M).
- *colocated* sources – all sources are generated at the same nodes (i.e., $O(s)$ is the same for all s).

Example



Network codes

A network code is a set of random variables
$\mathbf{Y} = (Y_s, Y_e, s \in \mathcal{S}, e \in \mathcal{E})$ such that

- Y_s – s^{th} source and is uniformly distributed over its supports
- Y_e – network coded symbol transmitted along link e

These random variables satisfy the following constraint:

Scr. Indep: $H(Y_s, s \in \mathcal{S}) = \sum_{s\in\mathcal{S}} H(Y_s)$

Encode: $H(Y_e | Y_f : f \to e) = 0, \quad \forall e \in \mathcal{E}$

Decode: for all $s \in \mathcal{S}$ and $u \in D(s)$,

$$H(Y_s | Y_f : f \to u, f \in \mathcal{S} \cup \mathcal{E}) \leq H(P_e) + P_e H(Y_s)$$

System parameters

For a given network code $\mathbf{Y} = (Y_s, Y_e, s \in \mathcal{S}, e \in \mathcal{E})$

- Rate capacity tuple

$$(\log|\mathrm{SP}(Y_s)|, \log|\mathrm{SP}(Y_e)|, \quad s \in \mathcal{S}, e \in \mathcal{E})$$

and $(c \log|\mathrm{SP}(Y_s)|, c \log|\mathrm{SP}(Y_e)|, \quad s \in \mathcal{S}, e \in \mathcal{E})$

- Error probability - probability that at least one of the decoder fails to reconstructed its requested source message

System parameters

For a given network code $\mathbf{Y} = (Y_s, Y_e, s \in \mathcal{S}, e \in \mathcal{E})$

- Rate capacity tuple

$$(\log |\mathrm{SP}(Y_s)|, \log |\mathrm{SP}(Y_e)|, \quad s \in \mathcal{S}, e \in \mathcal{E})$$

$$\text{and } (c \log |\mathrm{SP}(Y_s)|, c \log |\mathrm{SP}(Y_e)|, \quad s \in \mathcal{S}, e \in \mathcal{E})$$

- Error probability - probability that at least one of the decoder fails to reconstructed its requested source message

System parameters

For a given network code $\mathbf{Y} = (Y_s, Y_e, s \in \mathcal{S}, e \in \mathcal{E})$

- Rate capacity tuple

$$(\log|\mathrm{SP}(Y_s)|, \log|\mathrm{SP}(Y_e)|, \quad s \in \mathcal{S}, e \in \mathcal{E})$$

and $(c \log|\mathrm{SP}(Y_s)|, c \log|\mathrm{SP}(Y_e)|, \quad s \in \mathcal{S}, e \in \mathcal{E})$

- Error probability - probability that at least one of the decoder fails to reconstructed its requested source message

0-Achievability

Definition

A rate capacity tuple $(\lambda, \omega) = (\lambda(s) : s \in \mathcal{S}, \omega(e) : e \in \mathcal{E})$ *is* *0-achievable* *if there exists zero-error network codes*

$$\{Y_f^n : f \in \mathcal{E} \cup \mathcal{S}\}$$

and $c_n > 0$ *such that for all* $e \in \mathcal{E}$ *and* $s \in \mathcal{S}$*,*

$$\lim_{n\to\infty} c_n \log |\mathrm{SP}(Y_s^n)| \geq \lambda(s),$$
$$\lim_{n\to\infty} c_n \log |\mathrm{SP}(Y_e^n)| \leq \omega(e).$$

v-Achievability

Definition

A rate capacity tuple $(\lambda, \omega) = (\lambda(s): \ s \in \mathcal{S}, \ \omega(e): \ e \in \mathcal{E})$ *is 0-achievable if there exists network codes (with vanishing errors)*

$$\{Y_f^n : f \ \in \mathcal{E} \cup \mathcal{S}\}$$

and $c_n > 0$ *such that for all* $e \in \mathcal{E}$ *and* $s \in \mathcal{S}$*,*

$$\lim_{n\to\infty} c_n \log |\mathrm{SP}(Y_s^n)| \geq \lambda(s),$$

$$\lim_{n\to\infty} c_n \log |\mathrm{SP}(Y_e^n)| \leq \omega(e).$$

Property

- For any subset $\mathcal{R}$, CL($\mathcal{R}$) contains all tuples (λ, ω) such that there exists a sequence of $(\lambda^n, \omega^n) \in \mathcal{R}$ and positive numbers c_n satisfying

$$\lim_{n\to\infty} c_n\omega^n(e) \le \omega(e),$$
$$\lim_{n\to\infty} c_n\lambda^n(s) \ge \lambda(s).$$

- if every tuple in $\mathcal{R}$ is 0-achievable (or v-achievable), then CL($\mathcal{R}$) is also 0-achievable (or v-achievable)

The Question

What is the set of 0-achievable and v-achievable rate tuples?

Tool: Entropy functions

- Let $\mathcal{N} = \mathcal{S} \cup \mathcal{E}$ and $|\mathcal{N}| = n$
- $\mathcal{H}[\mathcal{N}]$ – 2^n-dimensional Euclidean space
- $h \in \mathcal{H}[\mathcal{N}] \triangleq (h(\alpha), \alpha \subseteq \mathcal{N})$.
- h is called a rank function.
- h is *entropic* if there exists a set of random variables $\{Y_i, i \in \mathcal{N}\}$ such that $h(\alpha) = H(Y_\alpha)$ for all $\alpha \subseteq \mathcal{N}$.
- $h(\alpha|\beta) \triangleq h(\alpha \cup \beta) - h(\beta)$
- Let Γ^* be the set of all entropic rank functions.

Idea

- For any zero-error network code $(Y_s, s \in \mathcal{S}, Y_e, e \in \mathcal{E})$, it induces an entropic function $h \in \Gamma^*$ such that

$$h(\mathcal{S}) = \sum_{s \in \mathcal{S}} h(s)$$
$$h(e \mid f : f \to e, f \in \mathcal{S} \cup \mathcal{E}) = 0$$
$$h(s \mid f : f \to u, f \in \mathcal{S} \cup \mathcal{E}) = 0.$$

- Let

$$\lambda(s) = \log |\mathrm{SP}(Y_s)|$$
$$\omega(e) = \log |\mathrm{SP}(Y_e)|$$

- Then (λ, ω) is 0-achievable.

Idea

- Define

$$\mathcal{C}_I \triangleq \left\{ g : g(\mathcal{S}) = \sum_{s \in \mathcal{S}} g(s) \right\}.$$

$$\mathcal{C}_E \triangleq \{ g : g\,(e \mid f : f \to e, f \in \mathcal{S} \cup \mathcal{E}) = 0,\ \forall e \in \mathcal{E} \}$$

$$\mathcal{C}_D \triangleq \left\{ \begin{aligned} & g : g\,(s \mid f : f \to u, f \in \mathcal{S} \cup \mathcal{E}) = 0, \\ & \qquad\qquad\qquad \forall s \in \mathcal{S}, u \in D(s) \end{aligned} \right\}$$

- $h \in \mathcal{C}_I \cap \mathcal{C}_E \cap \mathcal{C}_D$
- Furthermore, $h(s) = \lambda^*(s)$ and $h(e) \leq \omega^*(e)$.
- Let $\mathrm{proj}(h) \triangleq (h(f), f \in \mathcal{S} \cup \mathcal{E})$ .. *coordinate-wise projection*
- Hence, $(\lambda, \omega) \in \mathrm{CL}(\mathrm{proj}(h))$

Outer bound

Theorem (Outer bound)

If a rate-capacity tuple (λ, ω) is 0-achievable, then

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}(\bar{\Gamma}^* \cap \mathcal{C}_I \cap \mathcal{C}_E \cap \mathcal{C}_D)).$$

Existing results

Theorem (Outer bound (Yeung))

A rate-capacity tuple (λ, ω) *is v-achievable, then*

$$(\lambda, \omega) \in \mathrm{CL}(\mathrm{proj}(\bar{\Gamma}^* \cap \mathcal{C}_I \cap \mathcal{C}_E \cap \mathcal{C}_D)).$$

Theorem (Achievable region (Yan et al.))

A rate-capacity tuple (λ, ω) *is v-achievable if and only if*

$$(\lambda, \omega) \in \mathrm{CL}(\mathrm{proj}(\overline{\mathrm{con}}(\Gamma^* \cap \mathcal{C}_I \cap \mathcal{C}_E) \cap \mathcal{C}_D)).$$

Our contributions

Theorem (Colocated sources)

If all sources are colocated, then

1. *A rate-capacity tuple* (λ, ω) *is 0-achievable if and only if it is v-achievable.*
2. *The outer bound is tight.*

Conjecture

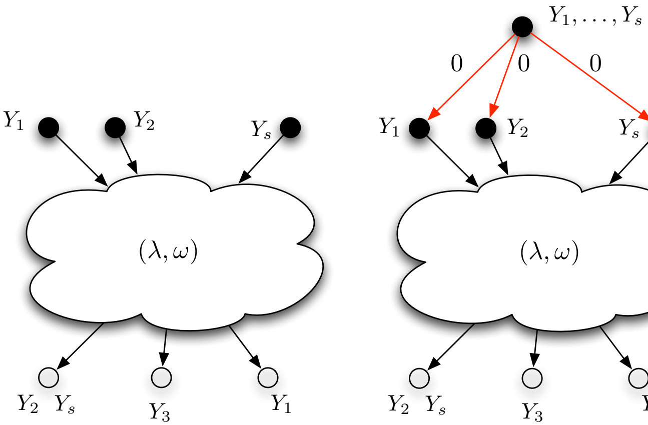
Theorem

The outer bound (for v-achievability)

$$\mathrm{CL}(\mathrm{proj}(\bar{\Gamma}^* \cap \mathcal{C}_I \cap \mathcal{C}_E \cap \mathcal{C}_D))$$

is tight even when sources are not colocated.

Evidence



- Our conjecture is true, if

  adding a super source node with vanishing rate to the original source nodes does not enlarge the set of v-achievable tuples.

Linearity constraint

Linear codes

Definition

A network code $\{Y_f : f \in \mathcal{E} \cup \mathcal{S}\}$ *with local encoding functions*

$$\Phi \triangleq \{\phi_e : e \in \mathcal{E}\}$$

is called q-linear if

1. Y_s *is a random row vector over* $GF(q)$.
2. *all the local encoding functions* ϕ_e *are linear.*

Linear codes

- Let the length of Y_s be λ_s.
- there exists matrices G_s and G_e such that

$$Y_s = [Y_i, i \in \mathcal{S}] \times G_s$$
$$Y_e = [Y_i, i \in \mathcal{S}] \times G_e.$$

- The matrices

$$\{G_f, f \in \mathcal{S} \cup \mathcal{E}\}$$

  will be called the *global encoding kernels*
- Define the linear relation between Y_e (the message sent along edge e) and $\{Y_s, s \in \mathcal{S}\}$ (the symbols generated at the sources).

Linear codes

- For using linear codes, decoding error is either 0 or at least $1 - 1/q$.
- 0-achievability and v-achievability are the same
- A network coding problem is subject to a *q-linearity constraint* if all allowable network codes are q-linear.
- Question - characterisation of 0-achievable rate capacity tuples subject to linearity constraint
- By using *representable functions*.

Representable functions

Definition

A rank function h is called q-representable if there exists vector subspaces

$$\{\mathbb{U}_i, i \in \mathcal{S} \cup \mathcal{E}\}$$

over $GF(q)$ such that for all $\alpha \subseteq \mathcal{S} \cup \mathcal{E}$,

$$h(\alpha) = \dim\langle \mathbb{U}_i, i \in \alpha \rangle.$$

Achievability

Theorem

For any networks (even when sources are not collocated), a rate-capacity tuple (λ, ω) is achievable if and only if

$$(\lambda, \omega) \in \mathsf{CL}\left(\mathrm{proj}\left[\bar{\Upsilon}_q^* \cap \mathcal{C}_I \cap \mathcal{C}_E \cap \mathcal{C}_D\right]\right).$$

where $\bar{\Upsilon}_q^*$ is the minimal closed and convex cone containing all representable functions.

Routing constraint

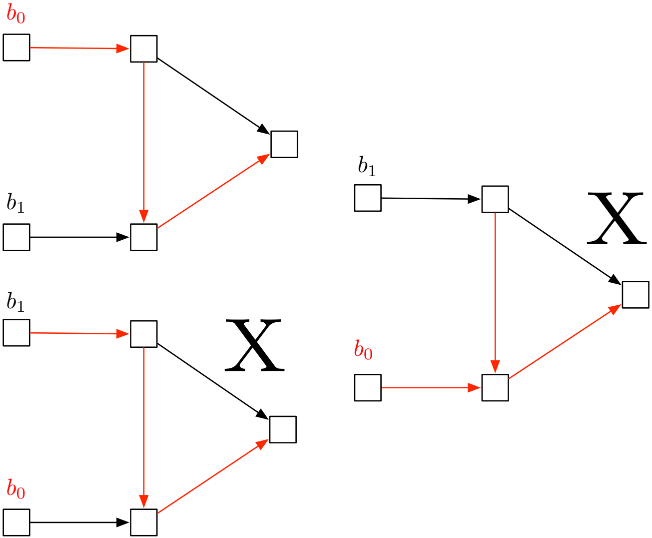
Routing subnetworks

Definition

A routing subnetwork is a subset $\mathcal{T} \subseteq \mathcal{S} \cup \mathcal{E}$ *such that*

1. $|\mathcal{T} \cap \mathcal{S}| = 1$ *(denoted it by* $\nu(\mathcal{T})$*)*
2. *For any link e* $\in \mathcal{T}$*, either there exists another link f* $\in \mathcal{T}$ *such that*
$$f \in in(e),$$
or the originating node of link e has access to the source $\nu(\mathcal{T})$*.*
3. *Hence, the subnetwork formed by the set of links in* $\mathcal{T}$ *is in fact "connected" and is "rooted" at* $\nu(\mathcal{T})$*.*

Example



Routing based scheme 1

- Each source Y_s is a q-ary row vector of length $\lambda(s)$.
- Routing subnetwork $\mathcal{T}_i$ will transmit c_i's q-ary symbols of Y_s to all sink nodes $u \in D(s)$.
- For error free decoding,

$$\lambda(s) = \sum_{i:\nu(\mathcal{T}_i)=s} c_i.$$

- Total number of q-ary symbols transmitted on e is

$$\lambda(s) = \sum_{i:e\in\mathcal{T}_i} c_i$$

Routing based scheme 1

Definition (Achievability)

A rate-capacity tuple (λ, ω) is achievable subject to a routing constraint if there exists a collection of routing subnetworks $\mathcal{T}_i$ and subnetwork capacities $c_i \geq 0$ such that

(R1) *For any edge $e \in \mathcal{E}$,*

$$\omega(e) \geq \sum_{i:e\in\mathcal{T}_i} c_i.$$

(R2) *For any i and $u \in D(\nu(\mathcal{T}_i))$, u is on the routing subnetwork. In other words, there exists $e \in \mathcal{T}_i$ such that $u \in head(e)$.*

(R3) *For any source $s \in \mathcal{S}$,*

$$\lambda(s) = \sum_{i:\nu(\mathcal{T}_i)=s} c_i.$$

Routing based scheme 1

- Source nodes perform no coding, except “partitioning” a source message into several independent pieces
- Each piece sent via a routing subnetwork
- each sink node must receive ALL piece from the requested source.
- A more general solution: source node encodes the source messages into “correlated pieces” instead.

Routing based scheme 2

- Let Y_s be a q-ary row vector of length $\lambda(s)$.
- Encode Y_s into $\sum_{i:\nu(\mathcal{T}_i)=s} c_i$'s q-ary symbols
- Any $\lambda(s)$ encoded symbols can reconstruct Y_s
- Sent these $\sum_{i:\nu(\mathcal{T}_i)=s} c_i$'s encoded symbols via the routing subnetworks
- Intermediate network nodes only store-and-forward
- A decoder can decode if it receives at least $\lambda(s)$'s encoded symbols of Y_s.

Routing based scheme 2

Definition (Generalised routing constraint)

A tuple (λ, ω) is called admissible subject to a generalised routing constraint if there exists a collection of routing subnetworks $\mathcal{T}_i$ and subnetwork capacities $c_i \geq 0$ such that

(R1) *For any edge $e \in \mathcal{E}$,*

$$\omega(e) \geq \sum_{i:e\in\mathcal{T}_i} c_i.$$

(R2') *for any source $s \in \mathcal{S}$ and any sink node $u \in D(s)$,*

$$\lambda(s) \leq \sum_{i:in(u)\cap\mathcal{T}_i\neq\emptyset \text{ and } \nu(\mathcal{T}_i)=s} c_i.$$

Routing capacity

- Characterisation of the set of achievable tuples, subject to routing constraint, is not new
- If
$$|head(e)| = 1, \quad \forall e \in \mathcal{E},$$
then the characterisation of admissible rate-capacity tuples subject to (generalised) routing constraint can be obtained by solving variations of the fractional Steiner tree packing problem.
- Our characterisation however highlight the differences (and similarities) between different characterisations with or without a (generalised) routing constraint.

Atomic functions

Definition (Atomic rank function)

A rank function h is called atomic in $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$ if there exists $\mathcal{T} \subseteq \mathcal{S} \cup \mathcal{E}$ such that

$$h(\beta) = \begin{cases} 1 & \text{if } \beta \cap \mathcal{T} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

It is called almost atomic if it can be written as a non-negative linear combination of atomic functions. In other words, h can be written as the following sum

$$h = \sum_i c_i h^i$$

where for all i, $c_i \geq 0$ and h^i is atomic.

Almost atomic functions

- Let $\Gamma_{\mathrm{AA}}(\mathrm{P})$, or simply Γ_{AA}, be the set of all almost atomic rank functions in $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$.
- Γ_{AA} is a closed and convex cone contained in Γ^*.
- Thus, all almost atomic rank functions are entropic.

Routing capacity

Theorem

A rate-capacity tuple (λ, ω) is admissible subject to a routing constraint if and only if

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[\Gamma_{\mathsf{AA}} \cap \mathcal{C}_E \cap \mathcal{C}_D \cap \mathcal{C}_I]).$$

Generalised routing capacity

Theorem

A rate-capacity tuple (λ, ω) is admissible subject to the generalised routing constraint if and only if

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}^*[\Gamma_{\mathsf{AA}} \cap \mathcal{C}_E \cap \mathcal{C}_I]).$$

where

$$\mathrm{proj}^*[h](s) \triangleq \min_{u \in D(s)} [h(in(u)) - h(s, in(u)) + h(s)]$$
$$\mathrm{proj}^*[h](e) \triangleq h(e).$$

Secrecy constraint

Secrecy constraint

- $|\mathcal{R}|$ adversaries in network
- Adversary r eavesdrop links in the set $\mathcal{B}_r$
- Aims to decode the set of sources $\mathcal{A}_r$.
- $W \triangleq \{(\mathcal{A}_r, \mathcal{B}_r), r \in \mathcal{R}\}$ is *wiretapping pattern*

The goal of "secure communications" is to transmit information over a network such that an eavesdropper can gain no information about its interested sources.

Stochastic network codes

Definition

A stochastic network code is a set of random variables

$$\{Y_f, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$$

such that Y_s is uniformly distributed and

$$h \in \mathcal{C}_I \cap \mathcal{C}_E$$

where h is its induced entropy function and

$$\mathcal{C}_I \triangleq \left\{ g : g(\mathcal{S}, \mathcal{V}) = \sum_{s \in \mathcal{S}} g(s) + \sum_{u \in \mathcal{V}} g(u) \right\}$$

$$\mathcal{C}_E \triangleq \{ g : g(s, in(e), tail(e)) = g(in(e), tail(e)), \forall e \in \mathcal{E} \}.$$

Stochastic network codes

Furthermore, the code is error free and strongly secure if

$$\mathcal{C}_D \triangleq \{g : g\,(in(u)) = g\,(s, in(u))\,, \forall s \in \mathcal{S}, u \in D(s)\}\,,$$
$$\mathcal{C}_S \triangleq \{g : g(\mathcal{A}_r) + g(\mathcal{B}_r) - g(\mathcal{A}_r, \mathcal{B}_r) = 0, \forall r \in \mathcal{R}\}\,.$$

Stochastic network codes

- $\{Y_u, u \in \mathcal{V}\}$ are "random seeds" available at nodes $u \in \mathcal{V}$ for stochastic encoding.

$$Y_e = \phi_e(Y_i, i \in in(e), Y_{tail(e)}).$$

- Hence, $H(Y_e \mid Y_i, i \in in(e), Y_{tail(e)})$ and $\mathcal{C}_E$
- *No* correlated or common keys shared among nodes in advance. $\{Y_u, u \in \mathcal{V}\}$ are NOT common keys. Locally and independently generated at each node.
- Hence, $\{Y_f, f \in \mathcal{S} \cup \mathcal{V}\}$ are mutually independent and $\mathcal{C}_I$

Strong secrecy constraint

Definition

A tuple (λ, ω) is 0-achievable subject to a strong secrecy constraint if there exists stochastic network codes

$$\{Y_f^n : f \in \mathcal{E} \cup \mathcal{S} \cup \mathcal{V}\}$$

and $c_n > 0$ such that

$$\begin{aligned} \lim_{n\to\infty} c_n \log |\mathrm{SP}(Y_e^n)| &\leq \omega(e), \\ \lim_{n\to\infty} c_n \log |\mathrm{SP}(Y_s^n)| &\geq \lambda(s), \\ H(Y_s^n | Y_f^n, f \in in(u)) &= 0, \\ I(Y_{\mathcal{A}_r}^n ; Y_{\mathcal{B}_r}^n) &= 0. \end{aligned}$$

Secure linear network codes

Theorem (Admissible region via linear codes)

Suppose $O(s)$ is a singleton for all $s \in \mathcal{S}$. A rate-capacity tuple (λ, ω) is achievable subject to q-linearity and strong secrecy constraint if and only if

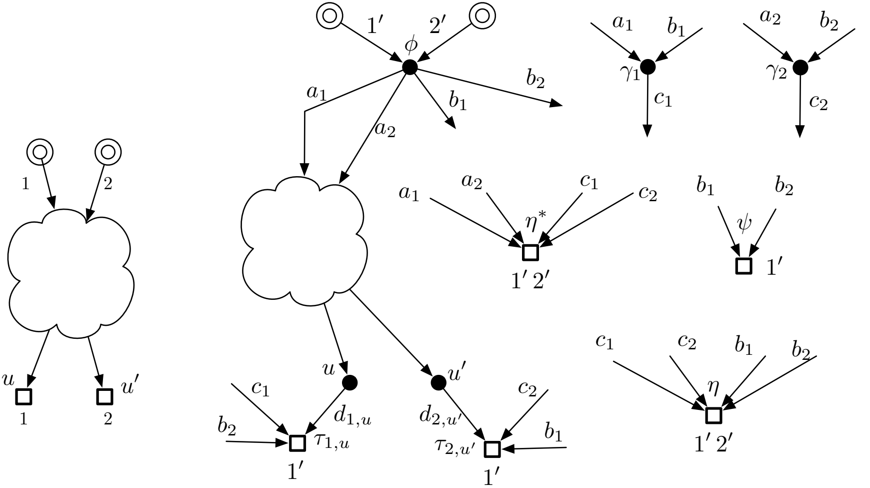
$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}[\bar{\Upsilon}_q^* \cap \mathcal{C}_I \cap \mathcal{C}_E \cap \mathcal{C}_D \cap \mathcal{C}_\mathsf{S}]).$$

Challenges

Incremental Multicast

- Totally ordered sources – receiver reconstruct source s also reconstruct all sources i for $i < s$.
- Common in transmission of multimedia – encoded into multiple layers.

Transformation

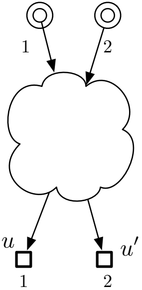


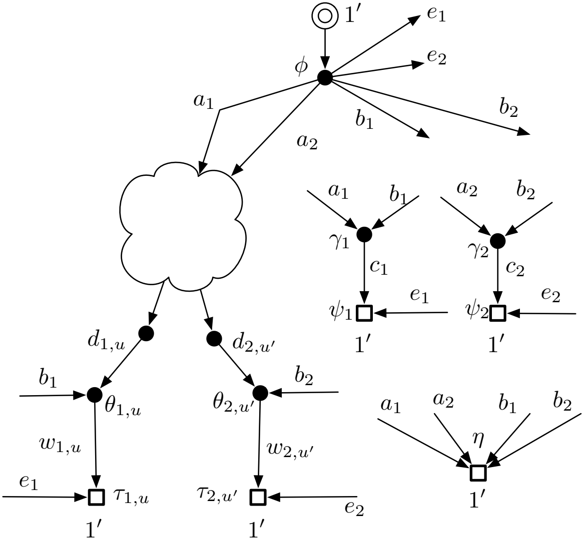
Incremental multicast is HARD

Theorem

Determining the set of v-achievable tuples in the incremental problem is NOT EASIER than determining the set of v-achievable tuples in the original multicast problem.

Secure multicast





Secure multicast is HARD

Theorem

Determining the set of v-achievable tuples in the secure multicast problem is NOT EASIER than determining the set of v-achievable tuples in the original multicast problem.

Conclusion

- We proved that when sources are colocated
  - Outer bound is tight
  - Imposing zero-error constraint will not reduce capacity.
- Conjecture that the outer bound is also tight when sources are not colocated
- Linearity, Routing and Secrecy constraint are considered
- Incremental multicast and secure multicast are as difficult as general multicast problems

Thank You !!