Analyzing Large Communication Networks

Shirin Jalali

joint work with Michelle Effros and Tracey Ho

Dec. 2015

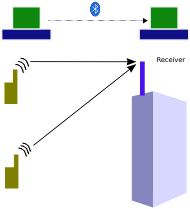
The gap

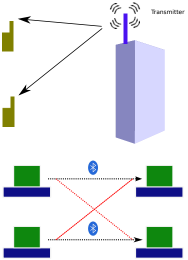
Fundamental questions:

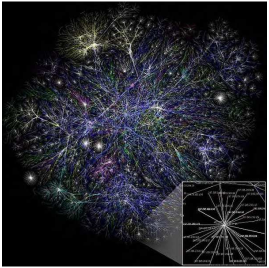
i. What is the best achievable performance?

ii. How to communicate over such networks?

Huge gap between theoretically analyzable and practical networks







visualization of the various routes through

a portion of the Internet from "The Opte Project".

This talk

Bridge the gap

- develop generic network analysis tools and techniques

Contributions:

- Noisy wireline networks:
  - Separation of source-network coding and channel coding is optimal
- Wireless networks:
  - Find outer and inner bounding noiseless networks.
- Noiseless wireline networks:
  - HNS algorithm

Noisy wired networks

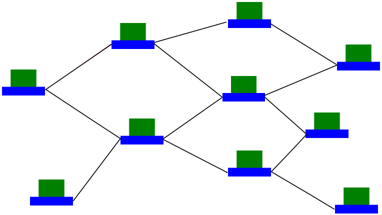
General wireline network

Example: Internet

Each user:

- sends data
- receives data from other users

Users observe dependent information



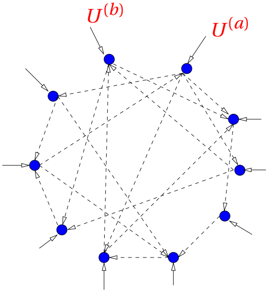
Wireline network

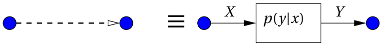
Represented by a directed graph:

- nodes = users and relays
- directed edges = point-to-point noisy channels

Node a:

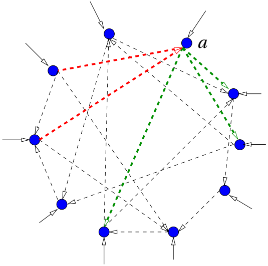
- observes random process $U^{(a)}$
- sources are dependent
- reconstructs a subset of processes observed by other nodes
- lossy or lossless reconstructions





Node operations

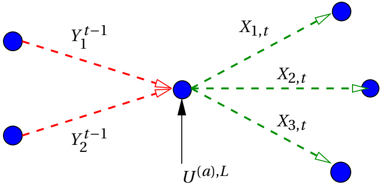
Node a observes $U^{(a),L}$.



Encoding at Node a:

- $t = 1, 2, \ldots, n$
- Map $U^{(a),L}$ and received signals up to time $t-1$ to the inputs of its outgoing channels

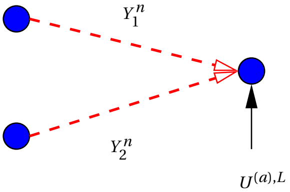
$$X_{j,t} = f_{j,t}(U^{(a),L}, Y_1^{t-1}, Y_2^{t-1})$$



Node operations

Decoding at Node a:

- At time $t = n$, maps $U^{(a),L}$ and its received signals to the reconstruction blocks.



- $\hat{U}^{(c \to a),L}$: reconstruction of node a from the data at node c

Performance measure

1. Rate:

   Joint source-channel-network: $\kappa \triangleq \frac{L}{n} = \frac{\text{source blocklength}}{\text{channel blocklength}}$

2. Reconstruction quality:

   - $U^{(a),L}$: observed block by node a
   - $\hat{U}^{(a\to c),L}$: reconstruction of node c from the data at node a

     i. Block-error probability (*Lossless reconstruction*):

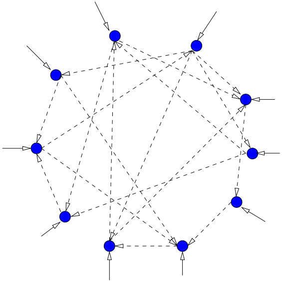
     $$\mathrm{P}(U^{(a),L} \neq \hat{U}^{(a\to c),L}) \to 0$$

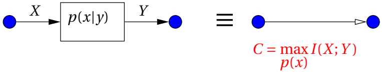
     ii. Expected average distortion (*Lossy reconstruction*):

     $$\mathrm{E}[d(U^{(a),L}, \hat{U}^{(a\to c),L})] \to D(a,c)$$

Separation of source-network coding and channel-network coding

Does separation hurt the performance?





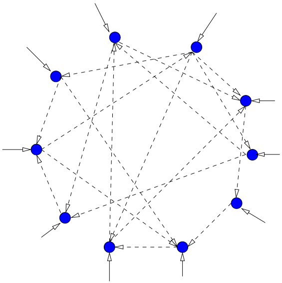
bit-pipe of capacity C carries $\lfloor nC \rfloor$ bits error-free over n communications.

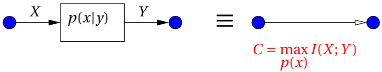
Theorem (SJ, Effros 2015)

Separation of source-network coding and channel coding is optimal in a wireline network with dependent sources.

Separation of source-network coding and channel-network coding

Does separation hurt the performance?





bit-pipe of capacity C carries $\lfloor nC \rfloor$ bits error-free over n communications.

Theorem (SJ, Effros 2015)

Separation of source-network coding and channel coding is optimal in a wireline network with dependent sources.

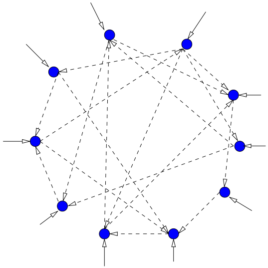
Separation: wireline networks

Single source multicast:

[Borade 2002], [Song, Yeung, Cai 2006]

Independent sources with lossless reconstructions:

[Hassibi, Shadbakht 2007] [Koetter, Effros, Medard 2009]



	multi-source	demands	dependent sources	lossless	lossy	continuous channels
[Borade 2002][Song et al. 2006]	no	multicast	no	yes	no	no
[Hassibi et al. 2007] [Koetter et al. 2009]	yes	arbitrary	no	yes	no	yes

Results

1. Separation of source-network coding and channel coding in wireline network with lossy and lossless reconstructions

2. Equivalence of zero-distortion and lossless reconstruction in general memoryless networks

	multi-source	demands	dependent sources	lossless	lossy	continuous channels
[Borade 2002] [Song et al. 2006]	no	multicast	no	yes	no	no
[Koetter et al. 2009]	yes	arbitrary	no	yes	no	yes
[SJ et al. 2015]	yes	arbitrary	yes	yes	yes	yes

Lossy reconstructions: Proof idea

Challenge: optimal region in not known!

Approach: any performance achievable on original network is achievable on the network of bit-pipes and vice versa.

Main ingredients:

- stacked networks
- channel simulation

Stacked network

Notation:

- Rate $\kappa = \frac{L}{n} = \frac{\text{source blocklength}}{\text{channel blocklength}}$
- $\mathcal{N}$: original network

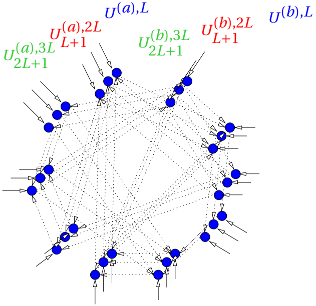
Defintions:

- $\mathscr{D}(\kappa, \mathcal{N})$: set achievable distortions on $\mathcal{N}$
- $\underline{\mathcal{N}}$: m-fold stacked version consisting of m copies of the original network

  [Koetter et al. 2009]

Theorem (SJ, Effros 2015)

$$\mathscr{D}(\kappa, \mathcal{N}) = \mathscr{D}(\kappa, \underline{\mathcal{N}})$$

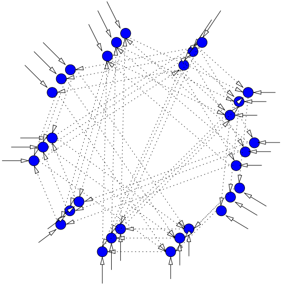


$\mathscr{D}(\kappa, \underline{\mathcal{N}}_b) = \mathscr{D}(\kappa, \underline{\mathcal{N}})$

$\mathcal{N}$ = original network

$\mathcal{N}_b$ = corresponding network of bit-pipes

$$\mathscr{D}(\kappa, \mathcal{N}) \stackrel{?}{=} \mathscr{D}(\kappa, \mathcal{N}_b)$$



It is enough to show that

$$\mathscr{D}(\kappa, \underline{\mathcal{N}}) = \mathscr{D}(\kappa, \underline{\mathcal{N}}_b).$$

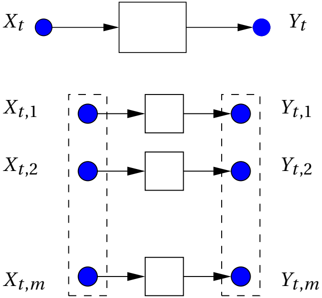
i. $\mathscr{D}(\kappa, \underline{\mathcal{N}}_b) \subset \mathscr{D}(\kappa, \underline{\mathcal{N}})$: easy (channel coding across the layers)

ii. $\mathscr{D}(\kappa, \underline{\mathcal{N}}) \subset \mathscr{D}(\kappa, \underline{\mathcal{N}}_b)$

Proof of $\mathscr{D}(\kappa, \underline{\mathscr{N}}) \subset \mathscr{D}(\kappa, \underline{\mathscr{N}}_b)$

Consider a noisy channel in $\mathscr{N}$ and its copies in $\underline{\mathscr{N}}$.

For $t = 1, \ldots, n$:



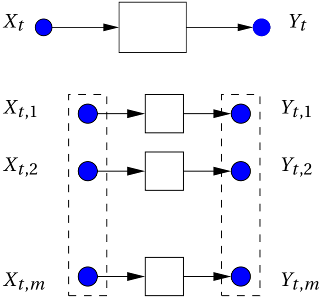
Define:

$$\hat{p}_{[X_t^m, Y_t^m]}(x, y) = \frac{|\{i : (X_{t,i}, Y_{t,i}) = (x, y)\}|}{m}$$

Proof of $\mathscr{D}(\kappa, \underline{\mathscr{N}}) \subset \mathscr{D}(\kappa, \underline{\mathscr{N}}_b)$

Consider a noisy channel in $\mathscr{N}$ and its copies in $\underline{\mathscr{N}}$.

For $t = 1, \ldots, n$:



Define:

$$\hat{p}_{[X_t^m, Y_t^m]}(x, y) = \frac{|\{i : (X_{t,i}, Y_{t,i}) = (x, y)\}|}{m}$$

Proof of $\mathscr{D}(\kappa, \underline{\mathcal{N}}) \subset \mathscr{D}(\kappa, \underline{\mathcal{N}}_b)$

In the original network:

$$\mathrm{E}[d(U^L, \hat{U}^L)] = \sum_{x,y} \mathrm{E}\left[d(U^L, \hat{U}^L) \,\middle|\, (X_t, Y_t) = (x, y)\right] \mathrm{P}\big((X_t, Y_t) = (x, y)\big).$$

Applying the same code across the layers in the m-fold stacked network:

$$\mathrm{E}\left[d(U^{mL}, \hat{U}^{mL})\right] = \sum_{x,y} \mathrm{E}\left[d(U^L, \hat{U}^L) \,\middle|\, (X_t, Y_t) = (x, y)\right] \mathrm{E}[\hat{p}_{[X_t^m, Y_t^m]}(x, y)].$$

Goal:

$$p_t(x) p(y|x) \approx \mathrm{E}[\hat{p}_{[X_t^m, Y_t^m]}(x, y)]$$

Proof of $\mathscr{D}(\kappa, \underline{\mathcal{N}}) \subset \mathscr{D}(\kappa, \underline{\mathcal{N}}_b)$

In the original network:

$$\mathrm{E}[d(U^L, \hat{U}^L)] = \sum_{x,y} \mathrm{E}\left[d(U^L, \hat{U}^L) \,\middle|\, (X_t, Y_t) = (x, y)\right] \mathrm{P}\big((X_t, Y_t) = (x, y)\big).$$

Applying the same code across the layers in the m-fold stacked network:

$$\mathrm{E}\left[d(U^{mL}, \hat{U}^{mL})\right] = \sum_{x,y} \mathrm{E}\left[d(U^L, \hat{U}^L) \,\middle|\, (X_t, Y_t) = (x, y)\right] \mathrm{E}[\hat{p}_{[X_t^m, Y_t^m]}(x, y)].$$

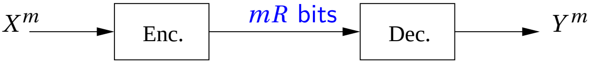
Goal:

$$p_t(x) p(y|x) \approx \mathrm{E}[\hat{p}_{[X_t^m, Y_t^m]}(x, y)]$$

Channel simulation

Channel $p_{Y|X}(y|x)$ with i.i.d. input $X \sim p_X(x)$

Simulate this channel:



such that

$$\|p_{X,Y} - \hat{p}_{[X^m,Y^m]}\|_{\mathrm{TV}} \xrightarrow{n\to\infty} 0, \text{ a.s.}$$

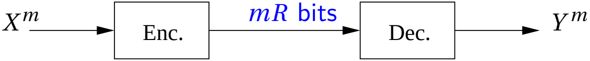
If $R > I(X;Y)$, such family of codes exists.

Since $R = C = \max_{p(x)} I(X;Y)$, such a code always exists.

Channel simulation

Channel $p_{Y|X}(y|x)$ with i.i.d. input $X \sim p_X(x)$

Simulate this channel:



such that

$$\|p_{X,Y} - \hat{p}_{[X^m,Y^m]}\|_{TV} \stackrel{n\to\infty}{\longrightarrow} 0, \text{ a.s.}$$

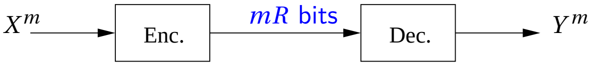
If $R > I(X;Y)$, such family of codes exists.

Since $R = C = \max_{p(x)} I(X;Y)$, such a code always exists.

Channel simulation

Channel $p_{Y|X}(y|x)$ with i.i.d. input $X \sim p_X(x)$

Simulate this channel:



such that

$$\|p_{X,Y} - \hat{p}_{[X^m,Y^m]}\|_{TV} \xrightarrow{n\to\infty} 0, \text{ a.s.}$$

If $R > I(X;Y)$, such family of codes exists.

Since $R = C = \max_{p(x)} I(X;Y)$, such a code always exists.

Results

So far we proved separation of lossy source-network coding and channel coding

	multi-source	demands	correlated sources	lossless	lossy	continuous channels
[Borade 2002][Song et al. 2006]	no	multicast	no	yes	no	no
[Koetter et al. 2009]	yes	arbitrary	no	yes	no	yes
[SJ et al. 2010]	yes	arbitrary	yes	no	yes	no

Lossless vs. $D = 0$

A family of lossless codes is also zero-distotion

Lossless reconstruction:

$$\mathrm{P}(U^L \neq \hat{U}^L) \to 0$$

For bounded distortion:

$$\mathrm{E}[d(U^L, \hat{U}^L)] \leq d_{\max} \mathrm{P}(U^L \neq \hat{U}^L) \to 0$$

But:

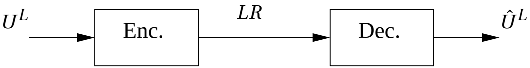
A family of zero-distortion codes is not lossless

$$\mathrm{E}[d(U^L, \hat{U}^L)] \to 0,$$

only implies

$$\frac{\{i : U_i \neq \hat{U}_i\}}{n} \to 0.$$

Lossless vs. $D=0$: point-to-point network



Lossless reconstruction:

$$R \geq H(U)$$

Lossy reconstruction:

$$R(D) = \min_{p(\hat{u}|u):\mathrm{E}\, d(U,\hat{U})\leq D} I(U;\hat{U})$$

- At $D=0$:

$$R(0) = \min_{p(\hat{u}|u):\mathrm{E}[d(U,\hat{U})]=0} I(U;\hat{U}) = I(U;U) = H(U).$$

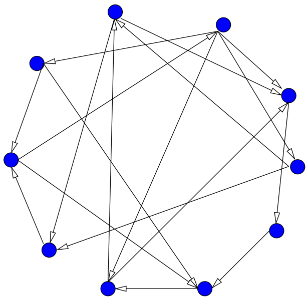
- minimum required rates for lossless reconstruction and $D=0$ coincide.

Lossless vs. $D = 0$: multi-user network

Explicit characterization of the rate-region is unknown for general multi-user networks.

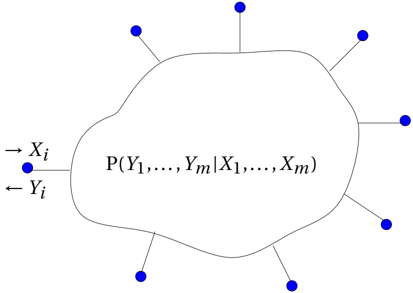
[Gu et al. 2010] proved the equivalence of zero-distortion and lossless reconstruction in error-free wireline networks:

$$\mathscr{R}(D)|_{D=0} = \mathscr{R}_L$$



Lossless vs. $D = 0$: multi-user network

In a general memoryless network [wired or wireless]:



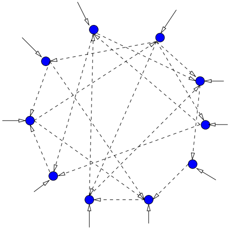
Theorem (SJ, Effros 2015)

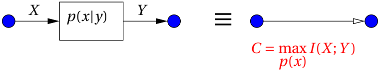
If for any $s \in \mathscr{S}$, $H(U_s | U_{\mathscr{S} \setminus s}) > 0$, then achievability of zero-distortion is equivalent to achievability of lossless reconstruction.

Recap

Wireline networks:

Proved that we can replace noisy point-to-point channels with error-free bit pipes



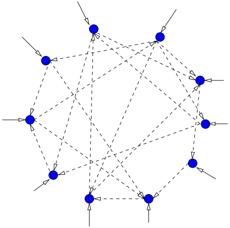


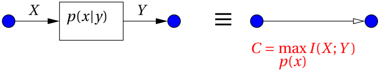
What about wireless networks?

Recap

Wireline networks:

Proved that we can replace noisy point-to-point channels with error-free bit pipes



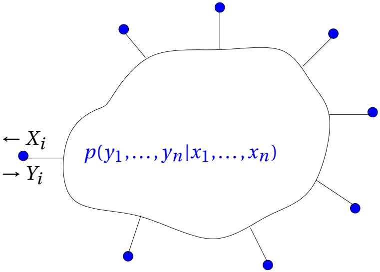


What about wireless networks?

Noisy wireless networks

Wireless networks

General multi-user network:



Separation of channel coding and source-network coding fails

The proof techniques can be extended to derive outer and inner bounding networks of bit pipes

[Jalali, Effros 2011]

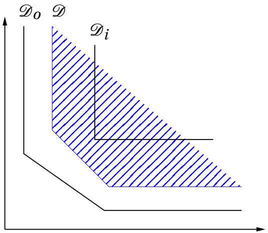
Outer/inner bounding network

Network $\mathscr{N}_o$ is an outer bounding network for $\mathscr{N}$ iff

$$\mathscr{D}(\kappa, \mathscr{N}) \subseteq \mathscr{D}(\kappa, \mathscr{N}_o)$$

Network $\mathscr{N}_i$ is an inner bounding network for $\mathscr{N}$ iff

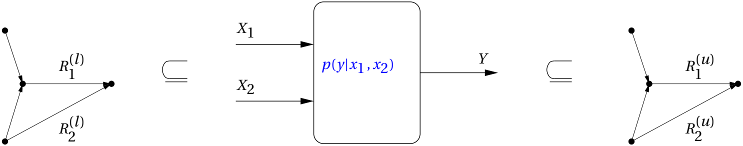
$$\mathscr{D}(\kappa, \mathscr{N}_i) \subseteq \mathscr{D}(\kappa, \mathscr{N})$$



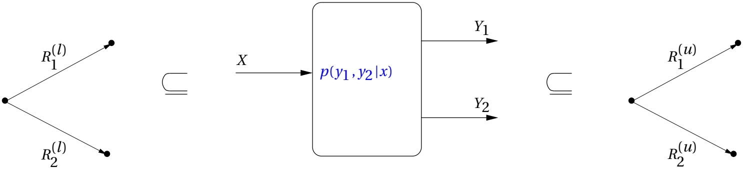
Set of achievable distortions on $\mathscr{N}$, $\mathscr{N}_i$, $\mathscr{N}_o$

Examples

Multiple access channel (MAC):



Broadcast channel (BC):



Recap

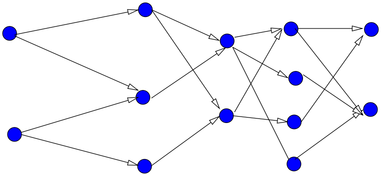
wireline network $\equiv$ network of bit pipes

network of bit pipes $\subset$ wireless network $\subset$ network of bit pipes

Noiseless wired networks

Noisy to noiseless

Acyclic noiseless network represented by a directed graph:



directed edge e = bit-pipe of capacity C_e

Question: What is the set of achievable rates?

Network coding: known results

1. Multicast: each receiver reconstructs all sources
   - Max-flow min-cut bound is tight

     [Ahlswede et al. 2000]
   - Linear codes suffices for achieving capacity

     [Li, Yeung, Cai 2003] [Koetter,Medard 2003]
2. Non-multicast: arbitrary demands
   - Linear codes are insufficient

     [Dougherty, Freiling, Zeger, 2005]
   - Capacity region is an open problem

     [Yeung 2002] [Song, Yeung 2003] [Yeung, Cai, Li, Zhang 2005] [Yan, Yeing, Zhang 2007]

Known bounds

Outer bounds:

- LP outer bound
  i. Tightest outer bound implied by Shannon inequalities
  ii. Software program: Information Theoretic Inequalities Prover (ITIP)

  [Yeung 97]

Inner bounds:

- Optimizing over scalar or vector linear network codes

  [Médard and Koetter 2003] [Chan 2007]

Main challenge:

- computational complexity of evaluating bounds is huge

Known bounds

Outer bounds:

- LP outer bound
  i. Tightest outer bound implied by Shannon inequalities
  ii. Software program: Information Theoretic Inequalities Prover (ITIP)

  [Yeung 97]

Inner bounds:

- Optimizing over scalar or vector linear network codes

  [Médard and Koetter 2003] [Chan 2007]

Main challenge:

- computational complexity of evaluating bounds is huge

Topological operations (component modeling)

Goal:

- find a (inner or outer) bounding network of smaller size

Idea:

- topological simplifications using recursive network operations
- replace a component with another smaller and functionally equivalent component

Functionally equivalent networks

For *any input distribution*, the two networks have identical set of achievable functional demands.

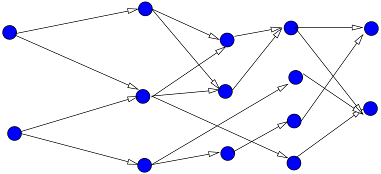
General procedure

Create a library of network simplification operations.

At each step:

i. Select a component in the network.

ii. Replace it by its equivalent or bounding component from the library.



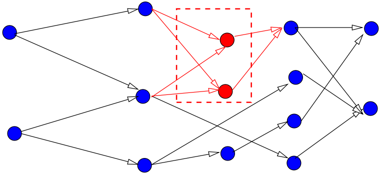
General procedure

Create a library of network simplification operations.

At each step:

i. Select a component in the network.

ii. Replace it by its equivalent or bounding component from the library.



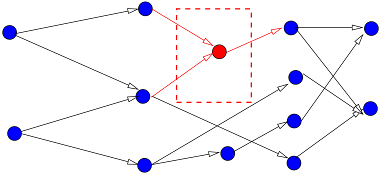
General procedure

Create a library of network simplification operations.

At each step:

i. Select a component in the network.

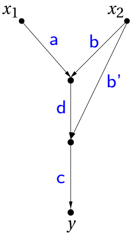
ii. Replace it by its equivalent or bounding component from the library.



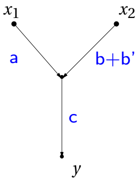
Example

Lemma

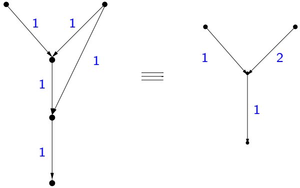
Let $\beta \triangleq \frac{b'}{b+b'}$. *If* $\beta a + (1-\beta)c \leq d$. *networks* $\mathcal{N}_1$ *and* $\mathcal{N}_2$ *are equivalent.*



Network $\mathcal{N}_1$

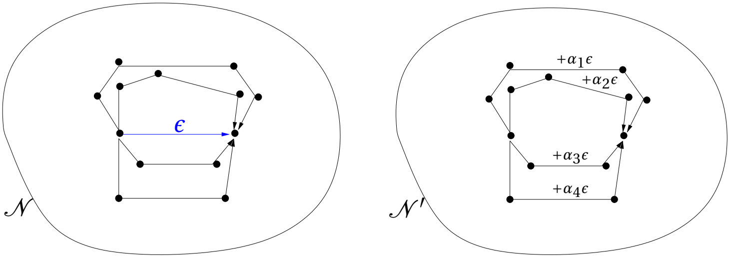


Network $\mathcal{N}_2$



[Ho, Effros, SJ 2010]

Rerouting flow



Removing edge e $\Rightarrow$ lower bounding network

Rerouting flow of edge e over other paths ($\sum \alpha_i = 1$) $\Rightarrow$ upper bounding network

Comparing inner and outer bounds

Consider network $(\mathcal{N}, c)$ and let

- $(\mathcal{N}_o, c_o)$: outer bounding network for $\mathcal{N}$
- $(\mathcal{N}_i, c_i)$: inner bounding network for $\mathcal{N}$

Question: How to compare the bounds?

Assume $\mathcal{N}_o$ and $\mathcal{N}_i$ have identical topologies.

Difference factor between $\mathcal{N}_i$ and $\mathcal{N}_o$ is defined:

$$\Delta = \Delta(c_i, c_o) \triangleq \max_{e \in \mathcal{E}} \frac{c_{e,o}}{c_{e,i}} \geq 1$$

Multiplicative bound

$$\mathcal{R}_i \subseteq \mathcal{R}_o \subseteq \Delta \mathcal{R}_i.$$

Comparing inner and outer bounds

Consider network $(\mathcal{N}, c)$ and let

- $(\mathcal{N}_o, c_o)$: outer bounding network for $\mathcal{N}$
- $(\mathcal{N}_i, c_i)$: inner bounding network for $\mathcal{N}$

Question: How to compare the bounds?

Assume $\mathcal{N}_o$ and $\mathcal{N}_i$ have identical topologies.

Difference factor between $\mathcal{N}_i$ and $\mathcal{N}_o$ is defined:

$$\Delta = \Delta(c_i, c_o) \triangleq \max_{e \in \mathcal{E}} \frac{c_{e,o}}{c_{e,i}} \geq 1$$

Multiplicative bound

$$\mathscr{R}_i \subseteq \mathscr{R}_o \subseteq \Delta \mathscr{R}_i.$$

Comparing inner and outer bounds

Consider network $(\mathcal{N}, c)$ and let

- $(\mathcal{N}_o, c_o)$: outer bounding network for $\mathcal{N}$
- $(\mathcal{N}_i, c_i)$: inner bounding network for $\mathcal{N}$

Question: How to compare the bounds?

Assume $\mathcal{N}_o$ and $\mathcal{N}_i$ have identical topologies.

Difference factor between $\mathcal{N}_i$ and $\mathcal{N}_o$ is defined:

$$\Delta = \Delta(c_i, c_o) \triangleq \max_{e \in \mathcal{E}} \frac{c_{e,o}}{c_{e,i}} \geq 1$$

Multiplicative bound

$$\mathcal{R}_i \subseteq \mathcal{R}_o \subseteq \Delta \mathcal{R}_i.$$

Comparing inner and outer bounds

Consider network $(\mathcal{N}, c)$ and let

- $(\mathcal{N}_o, c_o)$: outer bounding network for $\mathcal{N}$
- $(\mathcal{N}_i, c_i)$: inner bounding network for $\mathcal{N}$

Question: How to compare the bounds?

Assume $\mathcal{N}_o$ and $\mathcal{N}_i$ have identical topologies.

Difference factor between $\mathcal{N}_i$ and $\mathcal{N}_o$ is defined:

$$\Delta = \Delta(c_i, c_o) \triangleq \max_{e \in \mathcal{E}} \frac{c_{e,o}}{c_{e,i}} \geq 1$$

Multiplicative bound

$$\mathcal{R}_i \subseteq \mathcal{R}_o \subseteq \Delta \mathcal{R}_i.$$

Hierarchical network simplification (HNS)

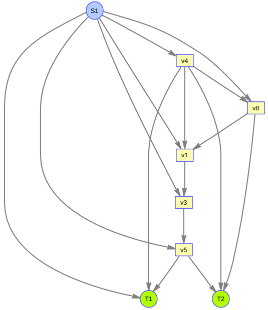
Given:

- network $G = (\mathcal{V}, \mathcal{E})$
- edge capacities $(C_e)_{e \in \mathcal{E}}$

HNS: heuristic algorithm

Output of HNS:

i. simpler feasible bounding network
ii. capacities of upper and lower bounding network



Original network: $|\mathcal{V}| = 8$ and $|\mathcal{E}| = 16$

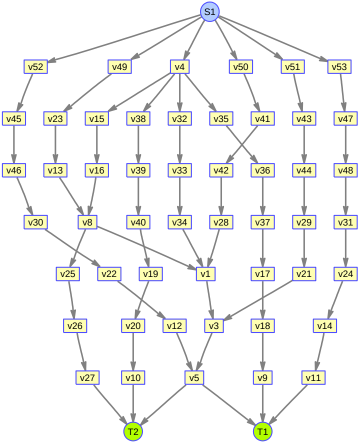
HNS Step 1: layering

Add extra nodes

- sources at top level
- sinks at the bottom
- relay nodes at the intermediate layers

Number of layers:

- length of longest path from a source to a sink



HNS Step 2: find and merge parallel paths

Find set of all parallel paths

Consider two such parallel paths:

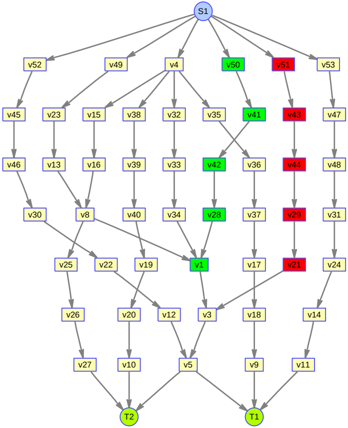
- $\mathscr{P}: v_0 \to v_1 \to v_2 \to \dots v_{\ell-1} \to v_\ell$
- $\mathscr{P}': v_0 \to v'_1 \to v'_2 \to \dots v'_{\ell-1} \to v_\ell$

Coalesce $\mathscr{P}$ and $\mathscr{P}'$ iff

i. $\{v'_2, \dots, v'_{\ell-1}\}$ are all SISO nodes

ii. for $i = 1, \dots, \ell-1$,

$$\frac{C_{v'_i \to v'_{i+1}}}{C_{v_i \to v_{i+1}}} \leq \gamma.$$



HNS Step 3: simplify

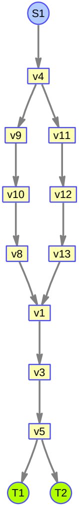
As the last topological step:

i. remove all SISO nodes
ii. combine parallel paths

Repeat the whole process (if necessary)

Output:

- candidate bounding network of smaller size



HNS Step 3: simplify

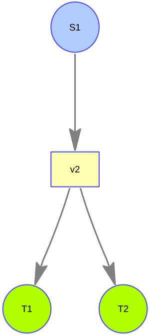
As the last topological step:

i. remove all SISO nodes
ii. combine parallel paths

Repeat the whole process (if necessary)

Output:

- candidate bounding network of smaller size



LP bounds

Given:

- Network $\mathscr{N}$ with edge capacities $c = (c_e)_{e \in \mathscr{E}}$
- bounding topology $\mathscr{B}$

Goal: find edge capacities $c_i = (c_{i,e})$ and $c_o = (c_{o,e})$ such that

$$\mathscr{B}(c_i) \subseteq \mathscr{N}(c) \subseteq \mathscr{B}(c_o)$$

Solution: characterize a set of LPs for finding c_i and c_o

[Effros, Ho, SJ 2010] [Effros, Ho, SJ, Xia 2012]

HNS Step 4: Linear Programming

LP 1:

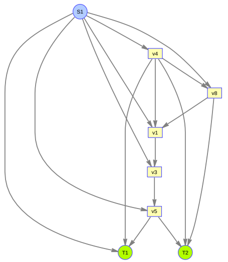
$$\begin{aligned} \min \quad & c_m \\ \text{s.t.} \quad & c_{e_2} \leq c_m, \forall\, e_2 \in \mathscr{E}_2 \\ & (c_2, f, r) \in \mathscr{M}(c_1) \end{aligned}$$

Let c_m^* solution of LP 1

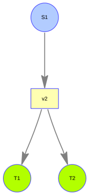
LP 2:

$$\begin{aligned} \min \quad & \sum_{e_2 \in \mathscr{E}_2} c_{e_2} \\ \text{s.t.} \quad & c_{e_2} \leq c_m^*, \forall\, e_2 \in \mathscr{E}_2 \\ & (c_2, f, r) \in \mathscr{M}(c_1). \end{aligned}$$

$\mathscr{N}(c) \subseteq \mathscr{B}(c')$



Original network: $|\mathscr{V}| = 8$ and $|\mathscr{E}| = 16$



Simplified network: $|\mathscr{V}| = 4$ and $|\mathscr{E}| = 3$

HNS Step 4: Linear Programming

LP 1:

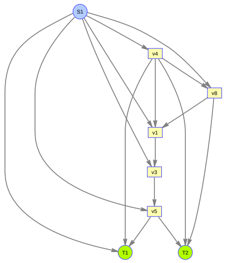
$$\begin{aligned} \min \quad & c_m \\ \text{s.t.} \quad & c_{e_2} \le c_m, \forall\, e_2 \in \mathcal{E}_2 \\ & (c_2, f, r) \in \mathcal{M}(c_1) \end{aligned}$$

Let c_m^* solution of LP 1

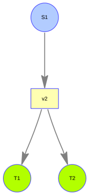
LP 2:

$$\begin{aligned} \min \quad & \sum_{e_2 \in \mathcal{E}_2} c_{e_2} \\ \text{s.t.} \quad & c_{e_2} \le c_m^*, \forall\, e_2 \in \mathcal{E}_2 \\ & (c_2, f, r) \in \mathcal{M}(c_1). \end{aligned}$$

$\mathcal{N}(c) \subseteq \mathcal{B}(c')$



Original network: $|\mathcal{V}| = 8$ and $|\mathcal{E}| = 16$



Simplified network: $|\mathcal{V}| = 4$ and $|\mathcal{E}| = 3$

HNS Step 4: Linear Programming

LP 1:

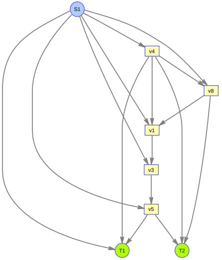
$$\begin{aligned} \min \quad & c_m \\ \text{s.t.} \quad & c_{e_2} \le c_m, \forall\, e_2 \in \mathscr{E}_2 \\ & (c_2, f, r) \in \mathscr{M}(c_1) \end{aligned}$$

Let c_m^* solution of LP 1

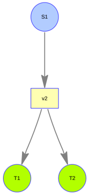
LP 2:

$$\begin{aligned} \min \quad & \sum_{e_2 \in \mathscr{E}_2} c_{e_2} \\ \text{s.t.} \quad & c_{e_2} \le c_m^*, \forall\, e_2 \in \mathscr{E}_2 \\ & (c_2, f, r) \in \mathscr{M}(c_1). \end{aligned}$$

$\mathscr{N}(c) \subseteq \mathscr{B}(c')$



Original network: $|\mathscr{V}| = 8$ and $|\mathscr{E}| = 16$



Simplified network: $|\mathscr{V}| = 4$ and $|\mathscr{E}| = 3$

HNS Step 4: Linear Programming

LP 1:

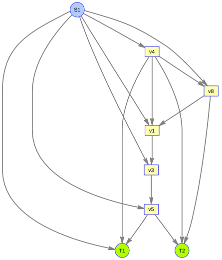
$$\begin{aligned} \min \quad & c_m \\ \text{s.t.} \quad & c_{e_2} \le c_m, \forall\, e_2 \in \mathscr{E}_2 \\ & (c_2, f, r) \in \mathscr{M}(c_1) \end{aligned}$$

Let c_m^* solution of LP 1

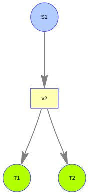
LP 2:

$$\begin{aligned} \min \quad & \sum_{e_2 \in \mathscr{E}_2} c_{e_2} \\ \text{s.t.} \quad & c_{e_2} \le c_m^*, \forall\, e_2 \in \mathscr{E}_2 \\ & (c_2, f, r) \in \mathscr{M}(c_1). \end{aligned}$$

$\mathscr{N}(c) \subseteq \mathscr{B}(c')$



Original network: $|\mathscr{V}| = 8$ and $|\mathscr{E}| = 16$

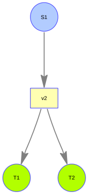


Simplified network: $|\mathscr{V}| = 4$ and $|\mathscr{E}| = 3$

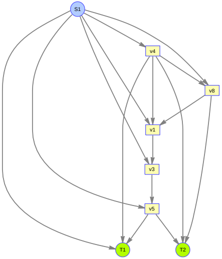
HNS Step 4: Linear Programming

LP 3:

$$\begin{aligned} \min \quad & k \\ \text{s.t.} \quad & (kc, f, r) \in \mathcal{M}(c') \end{aligned}$$



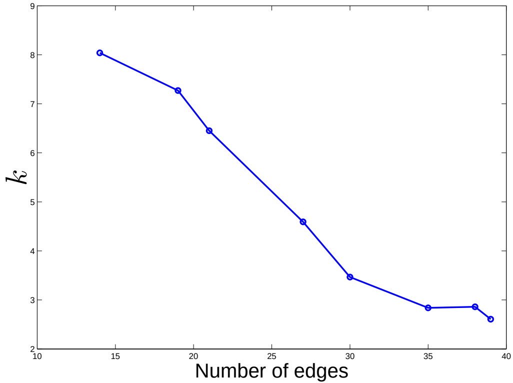
Simplified network: $|\mathcal{V}| = 4$ and $|\mathcal{E}| = 3$



Original network: $|\mathcal{V}| = 8$ and $|\mathcal{E}| = 16$

HNS performance

Performance achieved by varying γ:



Original network: $|\mathcal{V}| = 20$ and $|\mathcal{E}| = 40$

Summary

Wireline networks:
Separation of source-network coding and channel coding is optimal.

Wireless networks:
Find outer and inner bounding noiseless networks.

New approach to analyzing noiseless networks:

- iterative method
- step-by-step reduces the size of the graph
- at each step: one component is replaced by an equivalent or bounding component