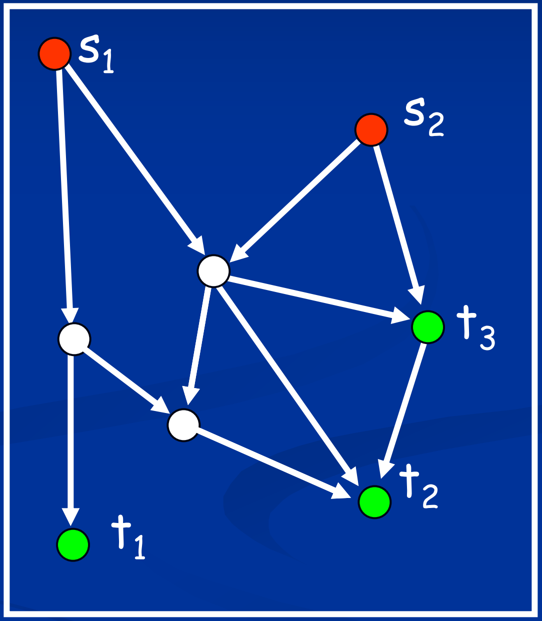
Algorithmic Complexity of Network Coding

Michael Langberg
Open University of Israel

General Network Coding

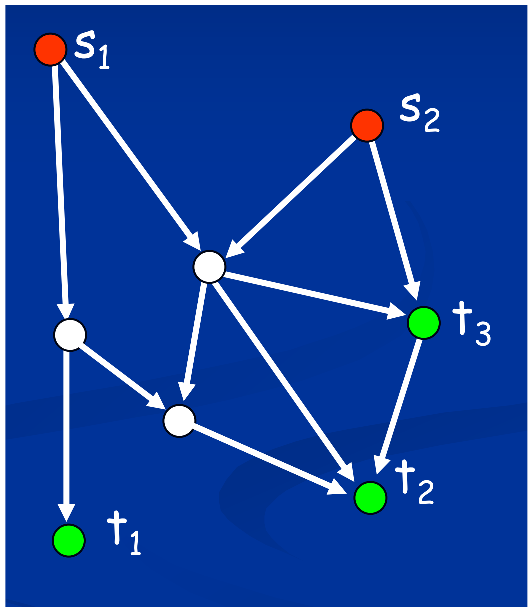
- Directed (acyclic) graph G=(V,E).
  - Assume unit capacity lossless edges.
- Source vertices S.
- Terminal vertices T.
- Requirement:
  - Transfer information from S to T.



G=(V,E)

General Network Coding

- Directed (acyclic) graph G=(V,E).
  - Assume unit capacity lossless edges.
- Source vertices S.
- Terminal vertices T.
- Requirement:
  - Transfer information from S to T.

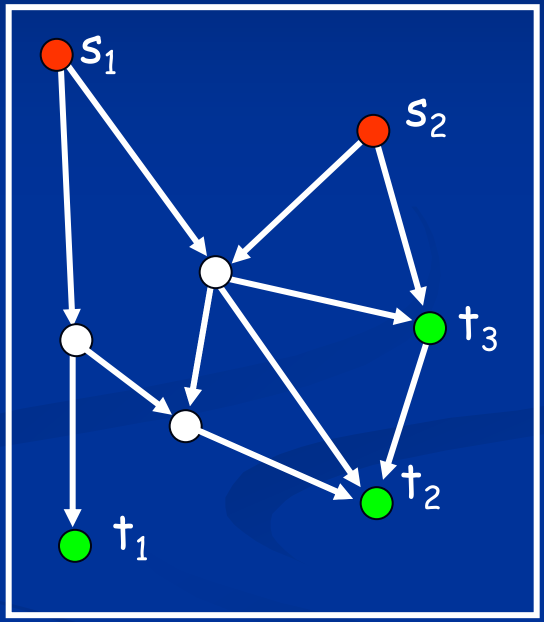


G=(V,E)

- Objectives:
  - Decide if given requirements are feasible using NC.
  - Design information flow using Network Coding.
  - Coding functions may be (Scalar/Vector) Linear or General.

- Multicast:
  - Single source, all terminals want all information at source.
  - Well understood.
  - Efficient solutions.
- General NC:
  - Not well understood.
- This talk:
  - Present evidence that the general problem is "hard".
  - Focus on "negative" results.

...oding

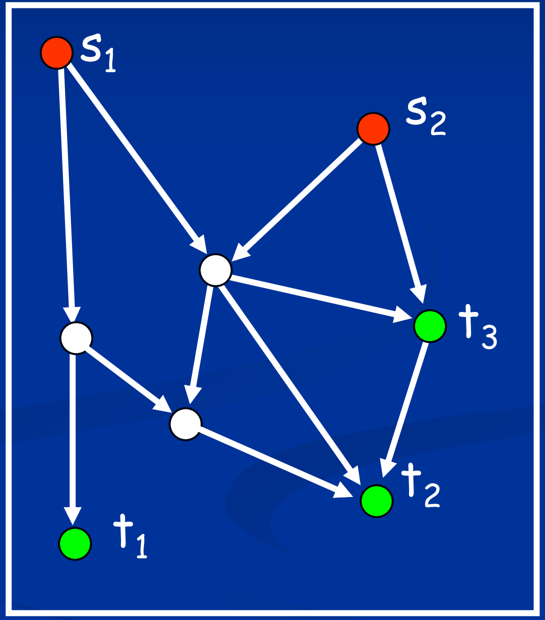


G=(V,E)

- Objectives:
  - Decide if given requirements are feasible using NC.
  - Design information flow using Network Coding.
  - Coding functions may be (Scalar/Vector) Linear or General.

...oding

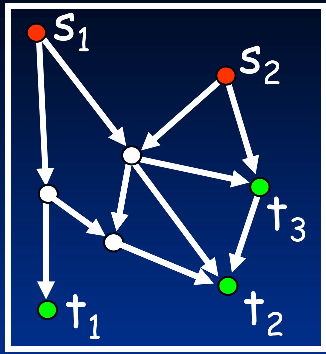
- Multicast:
  - Single source, all terminals want all information at source.
  - Well understood.
  - Efficient solutions.
- General NC:
  - Not well understood.
- This talk:
  - Present evidence that the general problem is "hard".
  - Focus on "negative" results.



G=(V,E)

- Objectives:
  - Decide if given requirements are feasible using NC.
  - Design information flow using Network Coding.
  - Coding functions may be (Scalar/Vector) Linear or General.

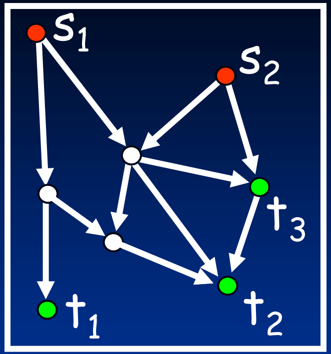
Scalar Linear Capacity



e.g. [Dougherty Freiling Zeger].

- General acyclic NC instance.
- Requirement matrix: $R=[r_{ij}]$
  - $r_{ij} \in \{0,1\}$, $r_{ij}=1 \Leftrightarrow$ terminal t_j wants info. of source s_i.
- Alphabet Σ of size q (here Σ will be a field).
- Scalar Linear: Each source has single character from Σ.
- Objective: design linear network code which is feasible.

Scalar Linear Capacity

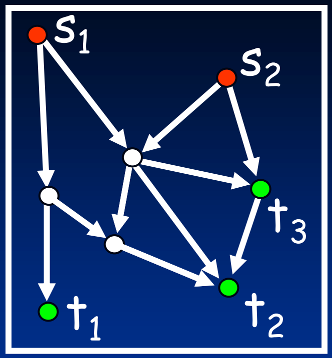


e.g. [Dougherty Freiling Zeger].

- General acyclic NC instance.
- Requirement matrix: $R=[r_{ij}]$
  - $r_{ij} \in \{0,1\}$, $r_{ij}=1 \Leftrightarrow$ terminal t_j wants info. of source s_i.
- Alphabet Σ of size q (here Σ will be a field).
- Scalar Linear: Each source has single character from Σ.
- Objective: design linear network code which is feasible.
- A Network Code enables communication at rate $1/m$:
  - Each link can carry m characters from Σ during transmission.

Capacity: C = maximum rate feasible linear NC.

Problem



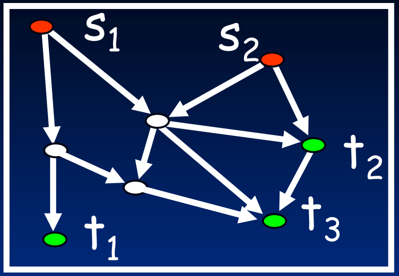
Input:

- Instance to the NC problem ($G,S,T,R=[r_{ij}]$).
- Desired communication rate r.

Output:

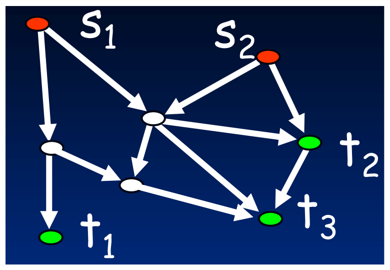
- Decision: Is the instance feasible (scalar linear).
- Construction: If so find corresponding NC.

Scalar Linear Coding



Q: Given an instance G with requirements $R=[r_{ij}]$, can one determine if instance has scalar linear capacity of 1.

Scalar Linear Coding

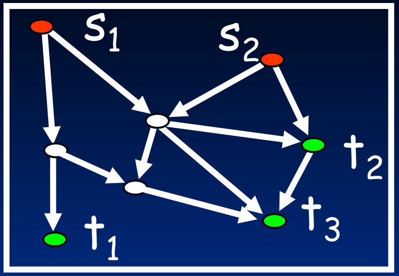


Q: Given an instance G with requirements $R=[r_{ij}]$, can one determine if instance has scalar linear capacity of 1.

A: "No" not efficiently [Lehman Lehman].

- NP-hard to determine scalar linear feasibility ($C=1$).
  - We are not even asking to find a network code!!

Scalar Linear Coding



Q: Given an instance G with requirements $R=[r_{ij}]$, can one determine if instance has scalar linear capacity of 1.

A: "No" not efficiently [Lehman Lehman].

- NP-hard to determine scalar linear feasibility ($C=1$).
  - We are not even asking to find a network code!!

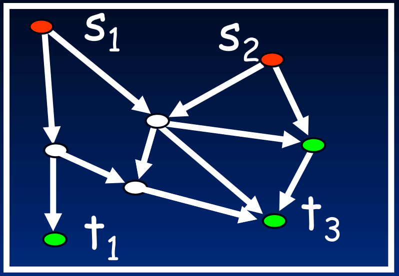
Proof Technique by reduction:

Show that solving the problem at hand efficiently will enable the efficient solution of a "hard" problem.

- Instance to hard problem $\Rightarrow$ Network Coding instance
- Solution to hard problem $\Leftarrow$ Solution to NC problem

Proof Technique by reduction:

Show that solving the problem at hand will solve a problem considered to be hard.



Q: Given an instance G with requirements $R=[r_{ij}]$, can one determine if instance has scalar linear capacity of 1.

A: "No" not efficiently [Lehman Lehman].

- NP-hard to determine scalar linear feasibility ($C=1$).
  - We are not even asking to find a network code!!
- Reduction from the 3-SAT problem.
- 3-SAT: given 3-CNF formula, determine if satisfiable.

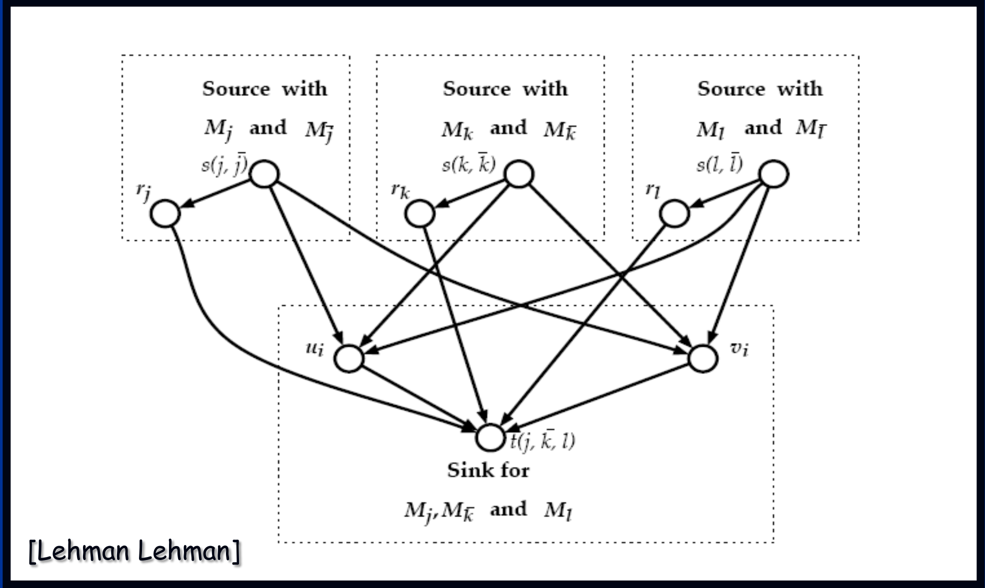
$$(x_1 \cup x_2 \cup x_3) \cap (x_4 \cup \bar{x}_2 \cup x_7) \cap \ldots \cap (\bar{x}_1 \cup \bar{x}_5 \cup x_6) \cap (\bar{x}_4 \cup x_9 \cup \bar{x}_5)$$

- 3-SAT is a classical NP-Complete problem.

Proof Technique by reduction:

- Instance to hard problem $\Rightarrow$ Network Coding instance
- Solution to hard problem $\Leftarrow$ Solution to NC problem

- Given 3-SAT instance φ [Lehman Lehman] construct network coding instance (G,R) such that:
  - Associate 2 sources with each variable corr. to TRUE and FALSE.
  - Single terminal with each clause.
  - With each clause associate a subgraph and terminal requirements.
  - For $(x_j \cup \bar{x}_k \cup x_l)$

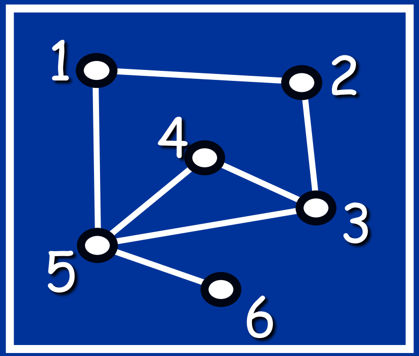


- Reduction works: φ is satisfiable iff (G,R) is feasible.

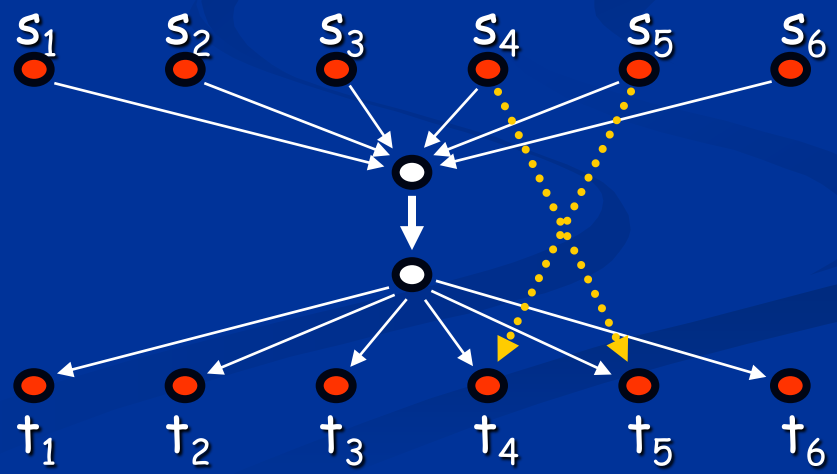
Index Coding

- Special instances of the general NC problem.
- Instances defined by an undirected base graph *G*.
- Combinatorial properties of *G* now pass on to the NC instance.
- Reduction, enables to establish hardness results via standard and well know combinatorial problems.

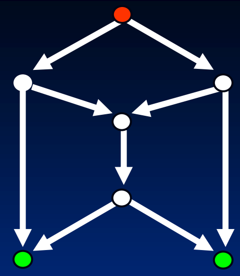
G



⇒

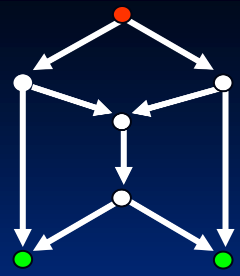


Index Coding

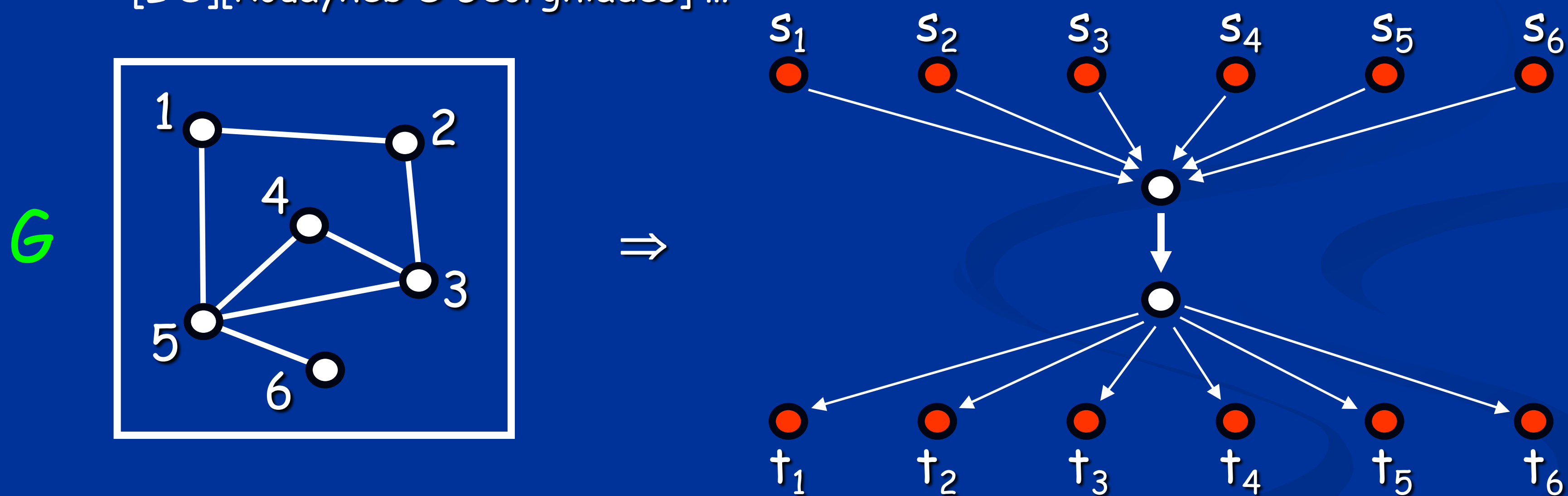


- Very simple construction – generalizes "butterfly".
- Index Coding appears in study of "source coding with side information" [Bar-Yossef et al][Lubetzky Stav][Alon et al][Yazdi et al][L S][Rouayheb S Georghiades] …

Index Coding

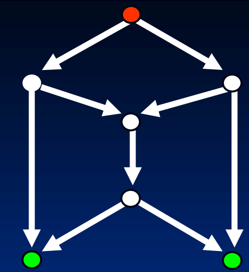


- Very simple construction – generalizes "butterfly".
- Index Coding appears in study of "source coding with side information" [Bar-Yossef et al][Lubetzky Stav][Alon et al][Yazdi et al][L S][Rouayheb S Georghiades] …



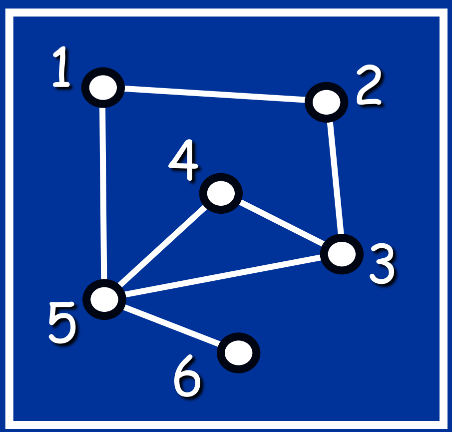
- Source/terminal for each vertex of G.
- Requirements: t_i wants information from s_i only.

Index Coding

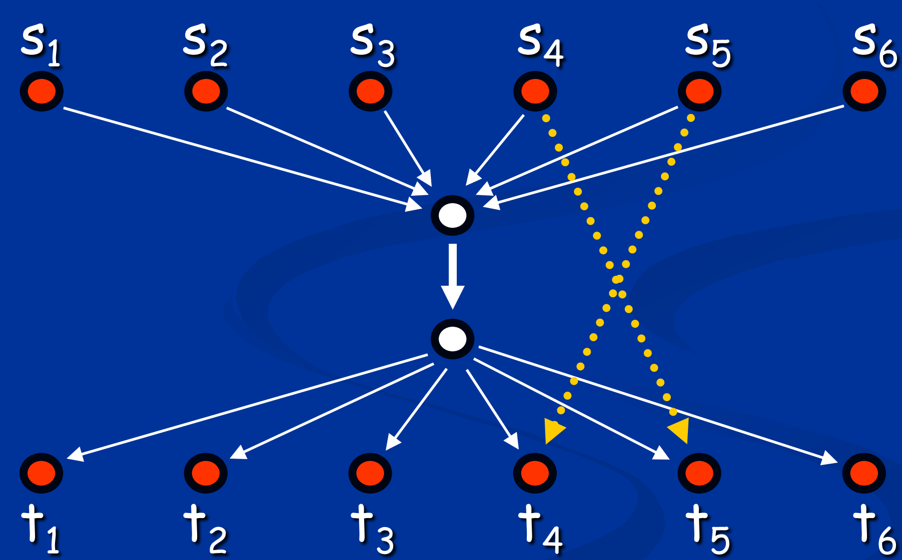


- Very simple construction – generalizes "butterfly".
- Index Coding appears in study of "source coding with side information" [Bar-Yossef et al][Lubetzky Stav][Alon et al][Yazdi et al] [L S][Rouayheb S Georghiades] …

G

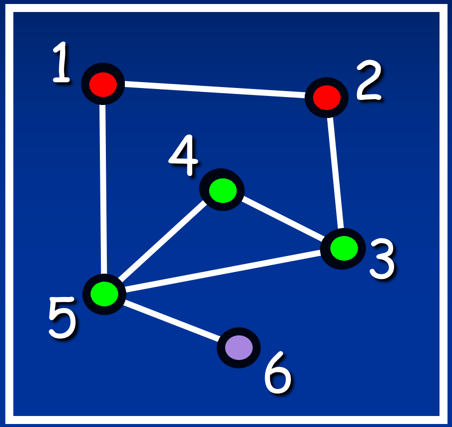


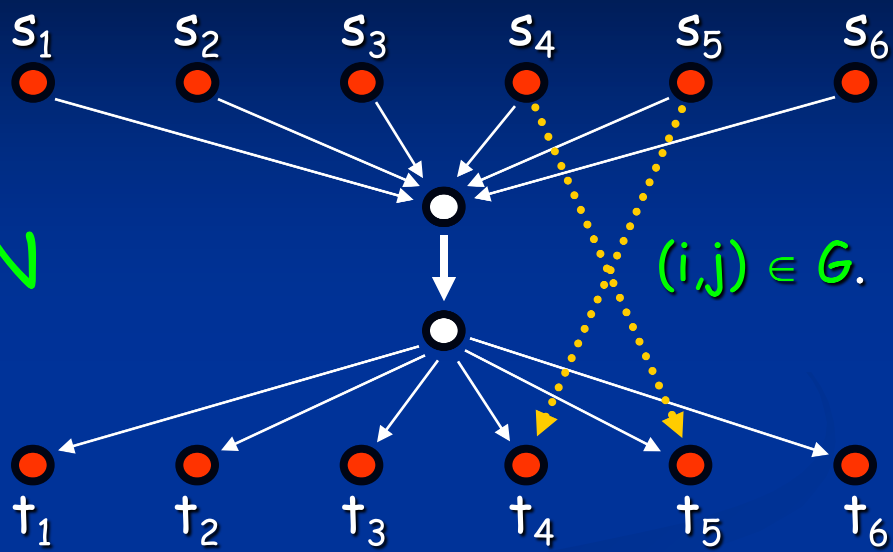
⇒



- Source/terminal for each vertex of G.
- Requirements: t_i wants information from s_i only.
- Side information: s_i connected to t_j ⇔ (i,j) edge in G.

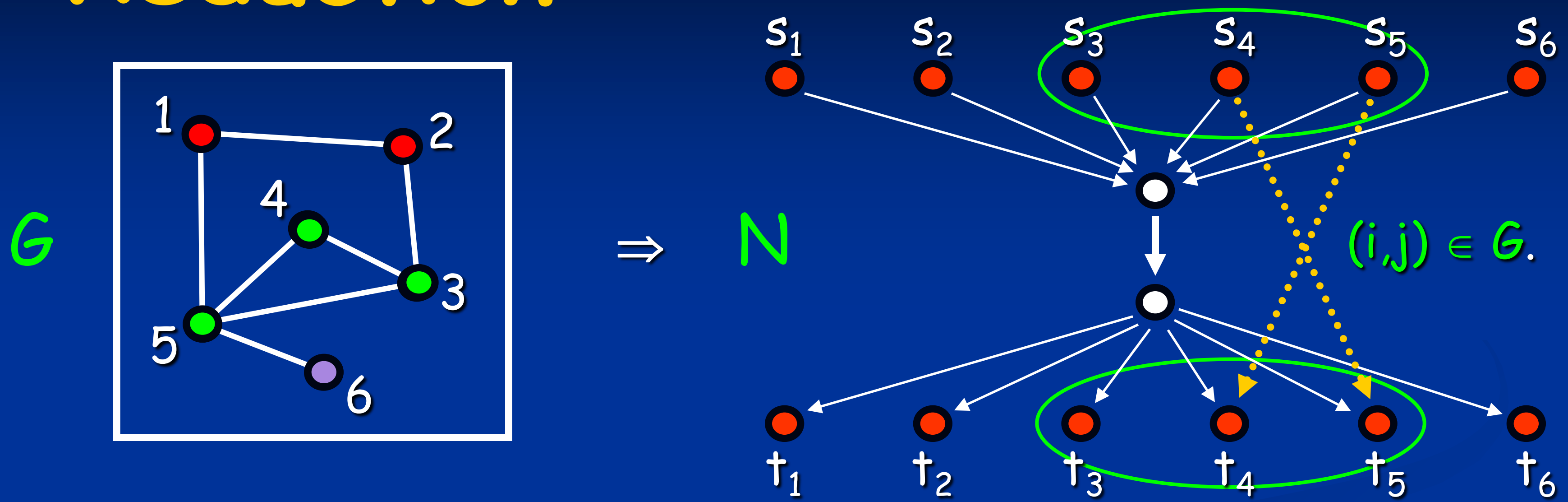
Reduction

G 

⇒ N 

Many beautiful connections between G and network N.

Reduction

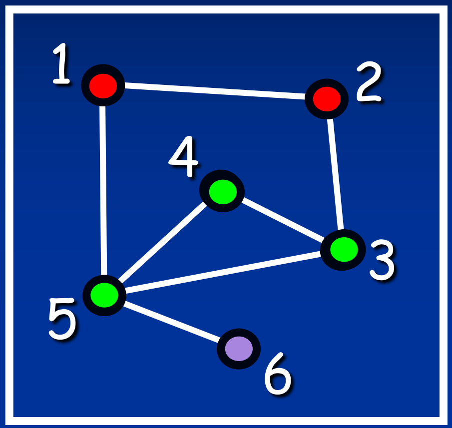


Many beautiful connections between G and network N.

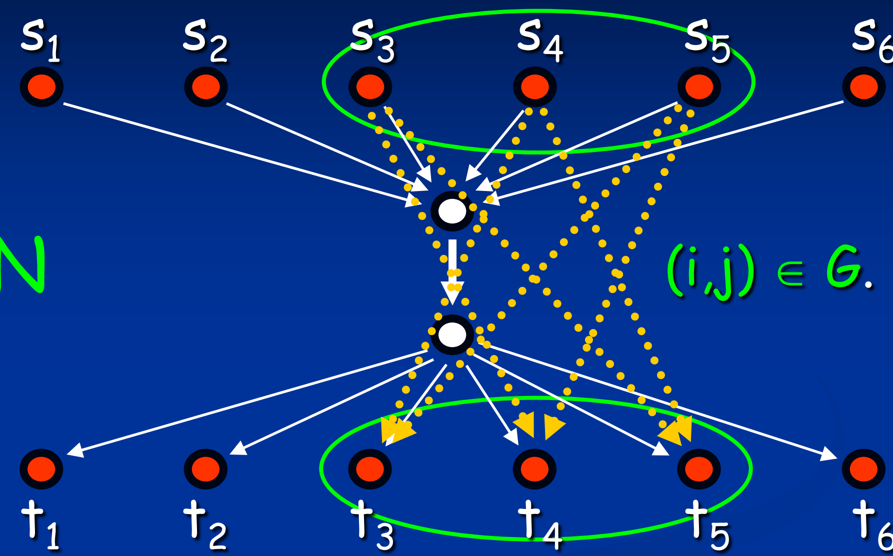
- Simple NC for (s,t) pairs corresponding to any clique in G:
- Consider clique (3,4,5) and pairs (s_3,t_3), (s_4,t_4), (s_5,t_5).

Reduction

G



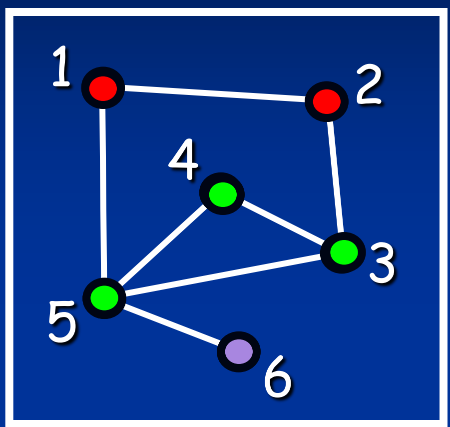
$\Rightarrow$ N

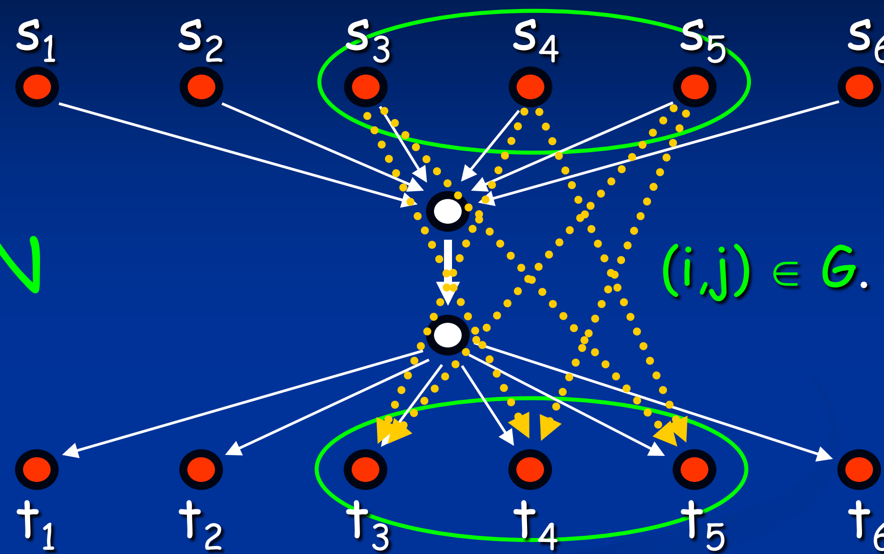


Many beautiful connections between G and network N.

- Simple NC for (s,t) pairs corresponding to any clique in G:
- Consider clique (3,4,5) and pairs (s_3,t_3), (s_4,t_4), (s_5,t_5).
  - Can transmit source info. (s_3,s_4,s_5) on side edges  and XOR $s_3 \oplus s_4 \oplus s_5$ on bottleneck edge.
  - t_3 gets s_4, s_5, $s_3 \oplus s_4 \oplus s_5 \Rightarrow$ can deduce s_3! (same for t_4, t_5).

Reduction

G 

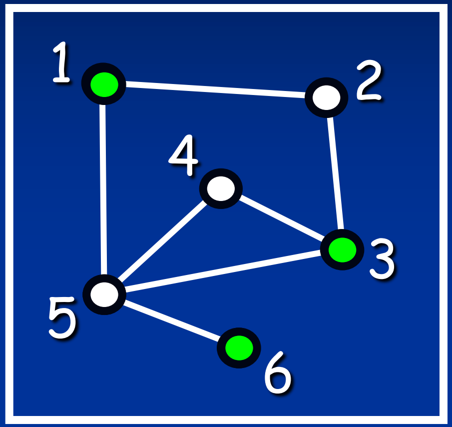
$\Rightarrow$ N 

Many beautiful connections between G and network N.

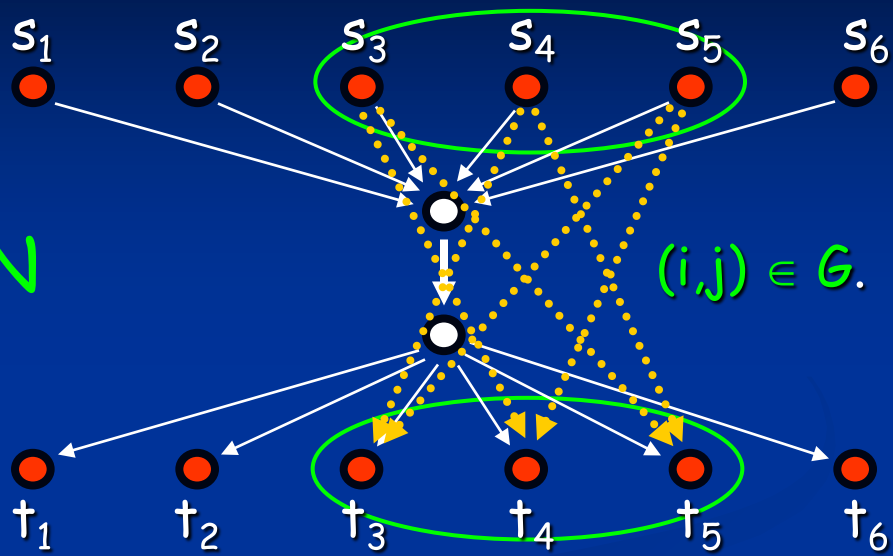
- Simple NC for (s,t) pairs corresponding to any clique in G:
- Consider clique $(3,4,5)$ and pairs $(s_3,t_3), (s_4,t_4), (s_5,t_5)$.
  - Can transmit source info. (s_3,s_4,s_5) on side edges and XOR $s_3 \oplus s_4 \oplus s_5$ on bottleneck edge.
  - t_3 gets $s_4,\ s_5,\ s_3 \oplus s_4 \oplus s_5 \Rightarrow$ can deduce s_3! (same for t_4,t_5).
- Can do the same with each clique of G (using time sharing).
- This example: scalar linear capacity $\geq 1/3$.

Sandwich

G



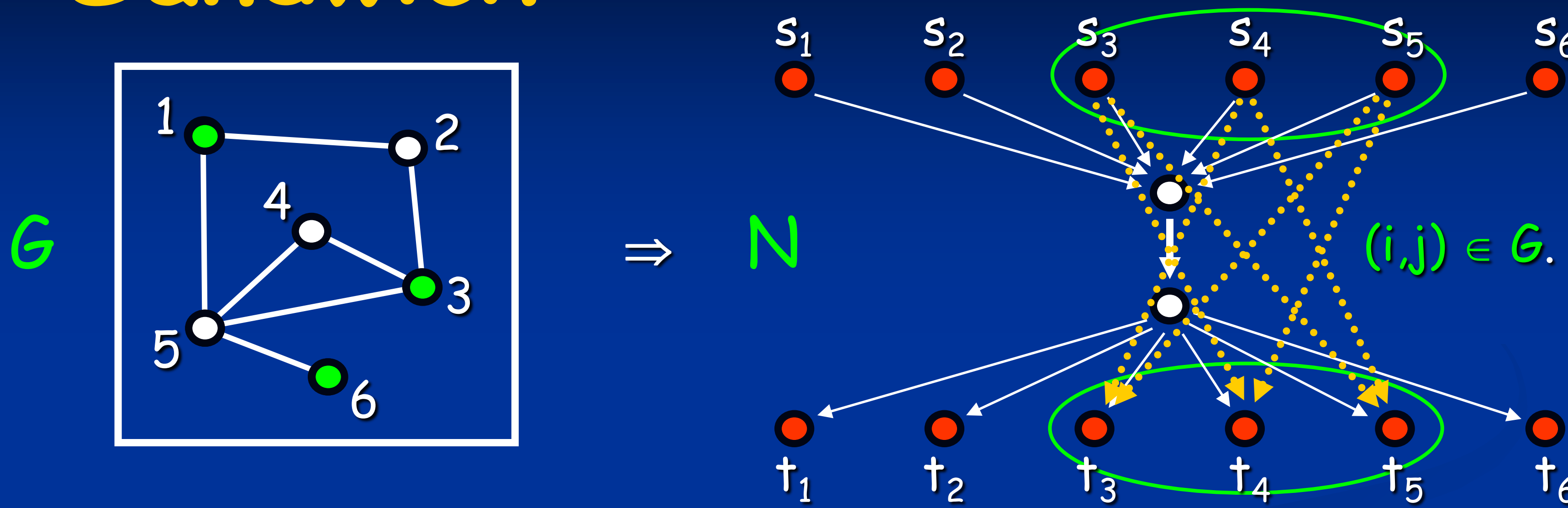
⇒ N



Sandwich

Clique cover: a cover of the vertices of G with cliques.

Independent set: a set of vertices of G with no edges.



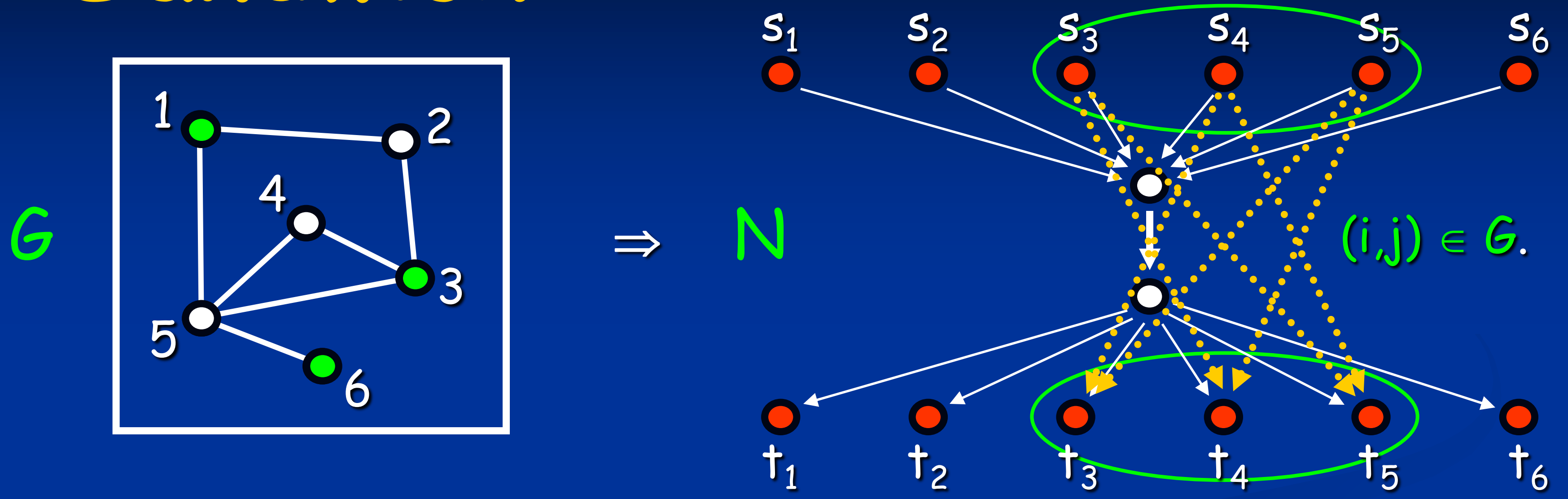
- Let ω be the size of a "clique cover" of G.
- Let α be the size of an "independent set" in G.

Clique cover: a cover of the vertices of G with cliques.

Sandwich

Clique cover: a cover of the vertices of G with cliques.

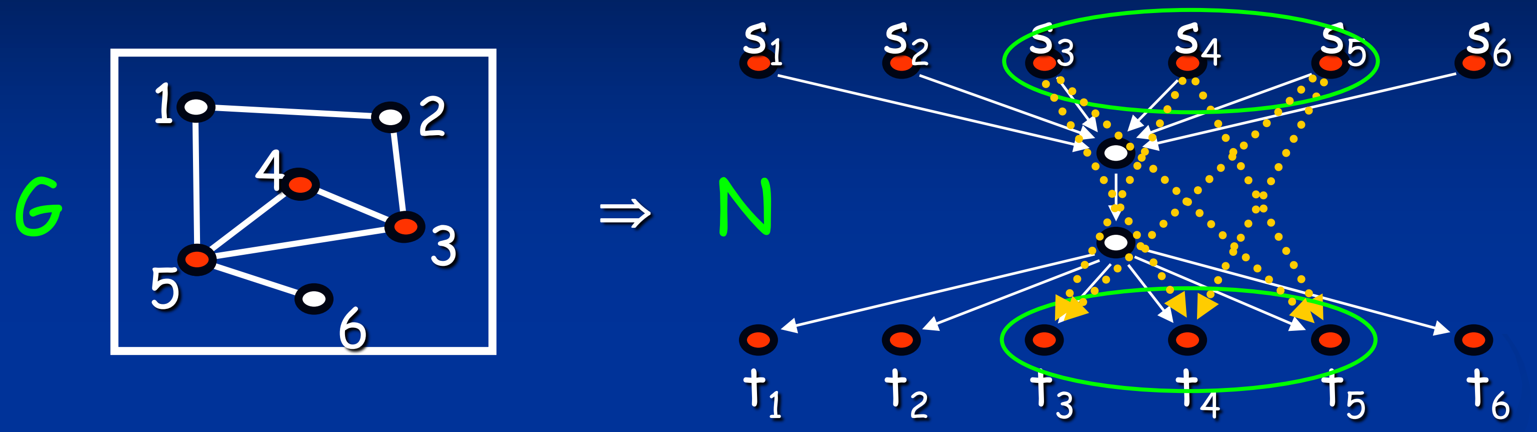
Independent set: a set of vertices of G with no edges.



- Let ω be the size of a "clique cover" of G.
- Let α be the size of an "independent set" in G.
- We just saw: Scalar Linear capacity $\geq 1/\omega$.
- Also holds that: Scalar Linear capacity $\leq 1/\alpha$ [Haemers,Peeters].

$$1/\omega \leq C \leq 1/\alpha$$

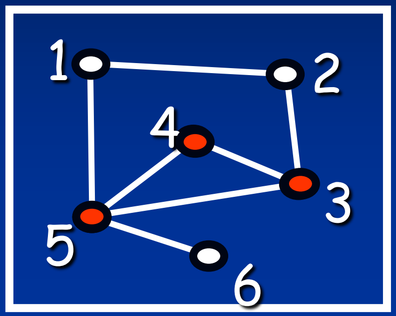
Proof II for Scalar Linear



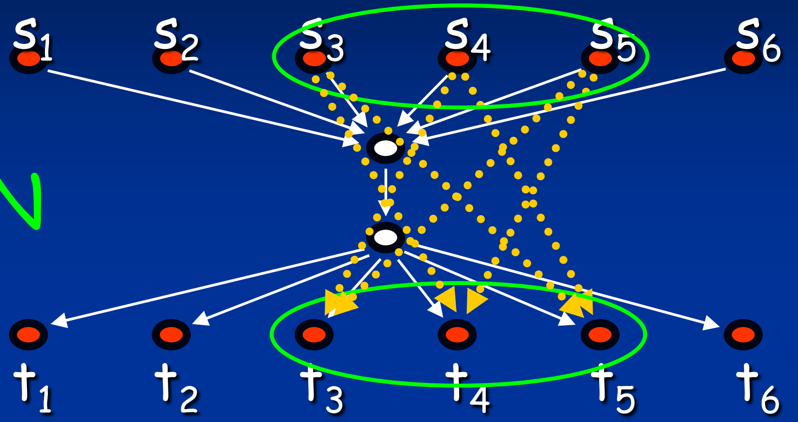
Proof Technique by reduction:

- Instance to hard problem ⇒ Network Coding instance
- Solution to hard problem ⇐ Solution to NC problem

G

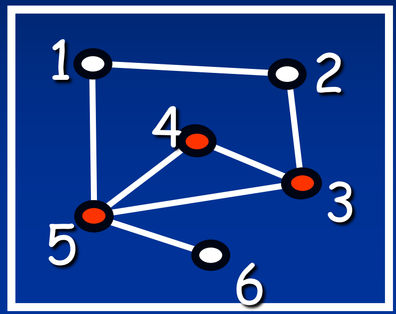


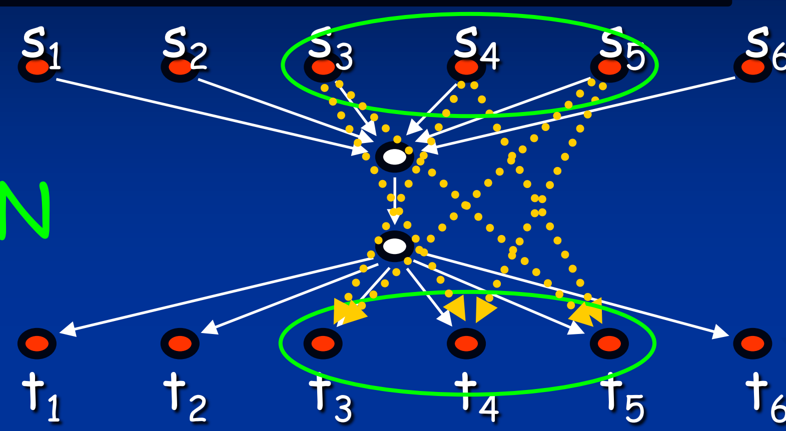
⇒ N



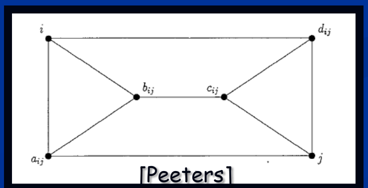
Proof Technique by reduction:

- Instance to hard problem ⇒ Network Coding instance
- Solution to hard problem ⇐ Solution to NC problem

G 

⇒ N 

- Known: NP-hard to decide if G has clique cover of size 3 [Gary et al.].
- In the 90's [Peeters] presented a reduction taking any G to a graph f(G) such that
  - G has clique cover of size 3 iff f(G) does.
  - f(G) has clique cover of size 3 iff Index Coding instance for f(G) has Scalar Linear capacity 1/3.
- "Gadgets" in proof:



[Peeters]

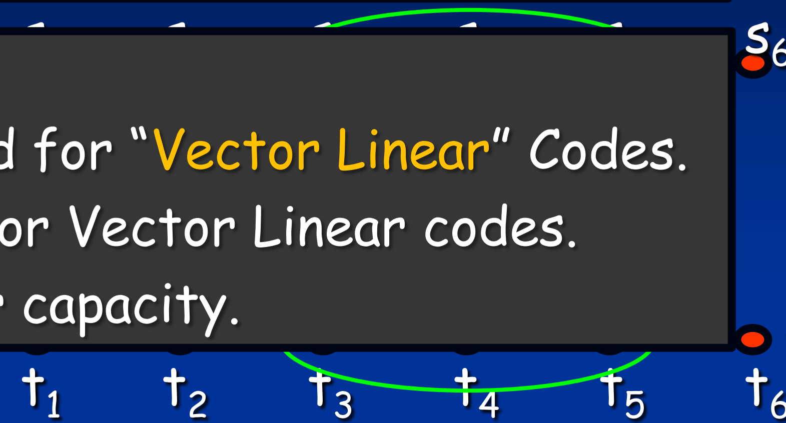
i	*	0	0	*	0	0
a_{ij}	0	*	0	0	*	0
b_{ij}	0	0	*	0	0	*
c_{ij}	*	0	0	*	0	0
d_{ij}	0	*	0	0	*	0
j	0	0	*	0	0	*

i	*	0	0	0	0	*
a_{ij}	0	*	0	*	0	0
b_{ij}	0	0	*	0	*	0
c_{ij}	0	*	0	*	0	0
d_{ij}	0	0	*	0	*	0
j	*	0	0	0	0	*

[Peeters]

Proof Technique by reduction:

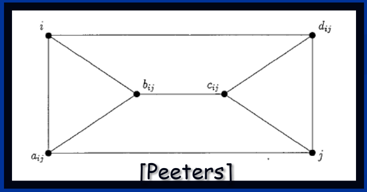
- Instance to hard problem ⇒ Network Coding instance
- Solution to hard problem ⇐ Solution to NC problem

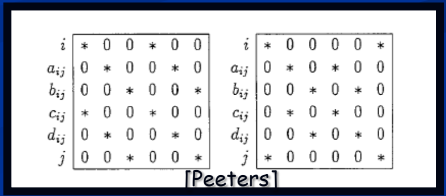


- Remark:
  - Both reductions do not hold for "Vector Linear" Codes.
  - Later: will show hardness for Vector Linear codes.

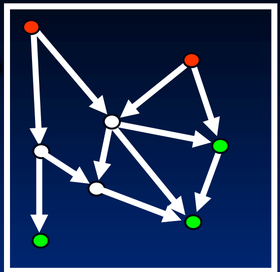
Now: estimation of Scalar Linear capacity.

- Known: NP-hard to decide if G has clique cover of size 3 [Gary et al.].
- In the 90's [Peeters] presented a reduction taking any G to a graph $f(G)$ such that
  - G has clique cover of size 3 iff $f(G)$ does.
  - $f(G)$ has clique cover of size 3 iff Index Coding instance for $f(G)$ has Scalar Linear capacity 1/3.
- "Gadgets" in proof:



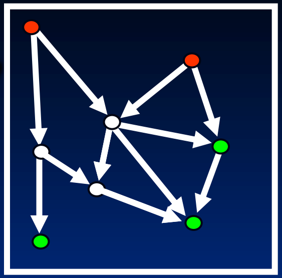


What about approximately finding capacity?



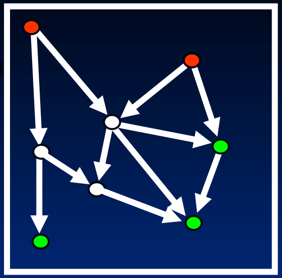
- Up to now: Finding Scalar-Linear NC that obtains capacity is NP-hard.
- Question: Is it easy to find a Scalar Linear NC that enables communication at rate 50% the capacity?

What about approximately finding capacity?



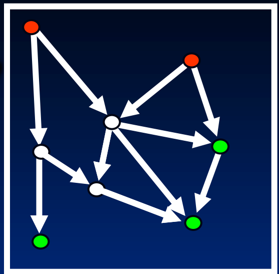
- Up to now: Finding Scalar-Linear NC that obtains capacity is NP-hard.
- Question: Is it easy to find a Scalar Linear NC that enables communication at rate 50% the capacity?
- NO! “Hard” to find a Scalar Linear NC that enables communication within any constant factor of capacity.

What about approximately finding capacity?



- Up to now: Finding Scalar-Linear NC that obtains capacity is NP-hard.
- Question: Is it easy to find a Scalar Linear NC that enables communication at rate ~~50%~~ the capacity?
- NO! "Hard" to find a Scalar Linear NC that enables communication within any constant factor of capacity.

What about approximately finding capacity?



- Up to now: Finding Scalar-Linear NC that obtains capacity is NP-hard.
- Question: Is it easy to find a Scalar Linear NC that enables communication at rate 0.001% the capacity?
- NO! "Hard" to find a Scalar Linear NC that enables communication within any constant factor of capacity.
- Main idea: Use Index Coding and connection to the clique cover [LSprintson].
  - Previous two constructions do not extend when trying to find NC that approximately meet capacity.

Structure

Will show: "Hard" to find a Network Code that enables communication within any constant factor of capacity.

Proof Structure (reduction):

Polynomial algorithm for approximately solving the NC problem $\Rightarrow$ Poly. alg. for long standing open problem in field of Graph Coloring.

Plan:

- Present "Hard" problem in Graph Coloring.
- Sketch proof (correctness of reduction).

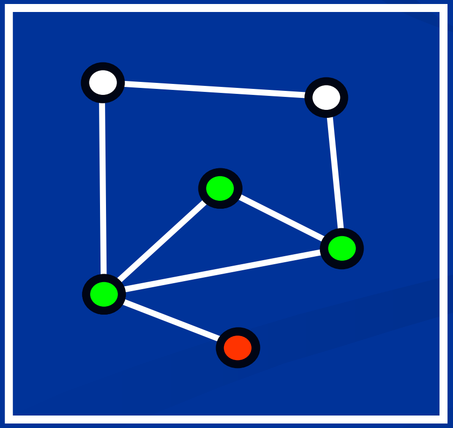
What is the "hard" problem?

A k-clique cover of G is an assignment of k colors to vertices of G such that each color class is a clique.

- $\omega(G)$ the minimum k such that G has k-clique cover.

Question: Can one efficiently find a minimum clique cover of G?
Answer: NO! Problem NP-Hard (implies solving SAT).

G



What is the "hard" problem?

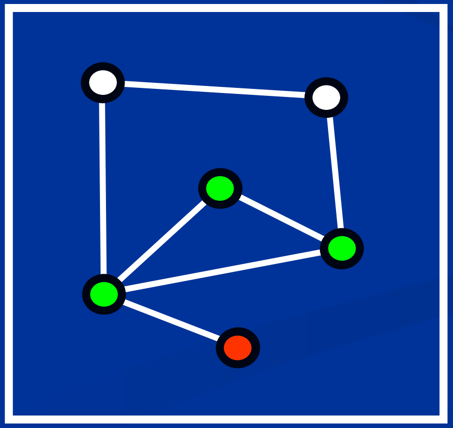
A k-clique cover of G is an assignment of k colors to vertices of G such that each color class is a clique.

- $\omega(G)$ the minimum k such that G has k-clique cover.

Question: Can one efficiently find a minimum clique cover of G?
Answer: NO! Problem NP-Hard (implies solving SAT).

Question: Can one eff. find a clique cover of G with $\omega(G) \cdot n^{\frac{1}{2}}$ colors?
Answer: NO! Problem NP-Hard.

G



What is the "hard" problem?

A k-clique cover of G is an assignment of k colors to vertices of G such that each color class is a clique.

- $\omega(G)$ the minimum k such that G has k-clique cover.

Question: Can one efficiently find a minimum clique cover of G?
Answer: NO! Problem NP-Hard (implies solving SAT).

Question: Can one eff. find a clique cover of G with $\omega(G) \cdot n^{\frac{1}{2}}$ colors?
Answer: NO! Problem NP-Hard.

Assume that one knows that G has 3-clique cover.
Question: Can one efficiently find such a 3-clique cover of G?
Answer: NO! Problem NP-Hard.

What is the "hard" problem?

A k-clique cover of G is an assignment of k colors to vertices of G such that each color class is a clique.

- $\omega(G)$ the minimum k such that G has k-clique cover.

Question: Can one efficiently find a minimum clique cover of G?
Answer: NO! Problem NP-Hard (implies solving SAT).

Question: Can one eff. find a clique cover of G with $\omega(G) \cdot n^{\frac{1}{2}}$ colors?
Answer: NO! Problem NP-Hard.

Assume that one knows that G has 3-clique cover.
Question: Can one efficiently find such a 3-clique cover of G?
Answer: NO! Problem NP-Hard.

Question: Can one efficiently find a clique cover of G with $3 \cdot \theta$ colors?
Answer: "NO". Long standing open problem for $\theta \leq \log(n)$, provably "hard" when θ is a universal constant.

What is the “hard” problem?

Will show:

Given undirected G that has a clique cover of size 3, one can efficiently construct a Network Coding instance such that:

- Any Scalar Linear Network Code over F_q with rate $\geq \rho C$ implies a clique cover of G with $q^{3/\rho}$ colors (if q,ρ are constant $\Rightarrow$ “hard”).

Answer: NO! Problem NP-Hard (implies solving SAT).

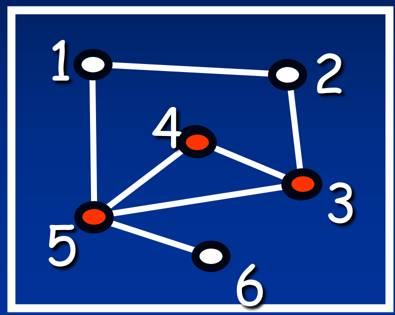
Question: Can one eff. find a clique cover of G with $\omega(G) \cdot n^{\frac{1}{2}}$ colors?
Answer: NO! Problem NP-Hard.

Assume that one knows that G has 3-clique cover.
Question: Can one efficiently find such a 3-clique cover of G?
Answer: NO! Problem NP-Hard.

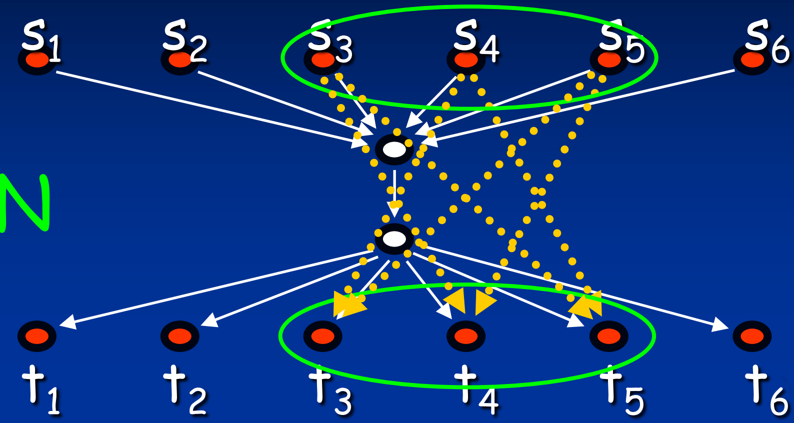
Question: Can one efficiently find a clique cover of G with $3 \cdot \theta$ colors?
Answer: “NO”. Long standing open problem for $\theta \leq \log(n)$, provably “hard” when θ is a universal constant.

Reduction

G

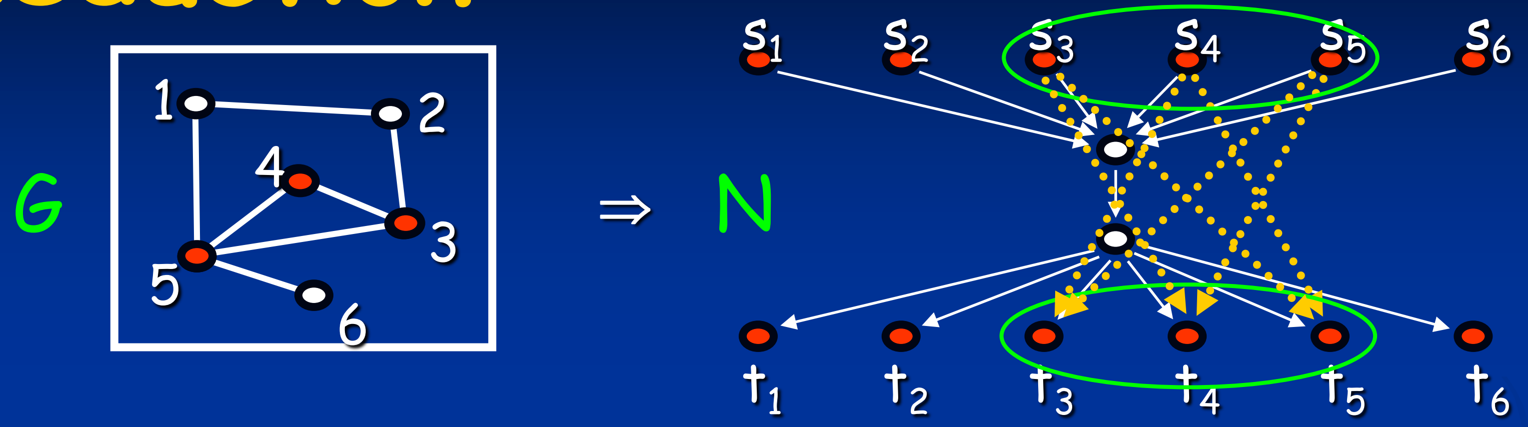


⇒ N



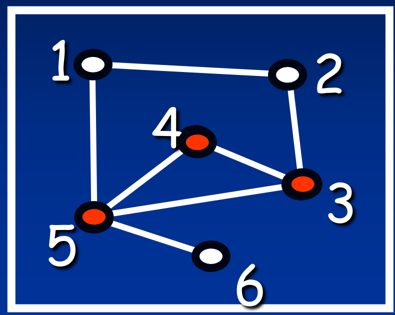
- Let G be a graph which can be covered by 3 cliques.
- Implies C(N) is at least 1/3 (time sharing between cliques).
- Key Lemma: From any scalar linear solution to N over F_q with rate $= 1/m \geq \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that eventually cover all of G.
- Number of cliques $\leq q^{3/\rho}$: We have a "small" clique cover of G!!

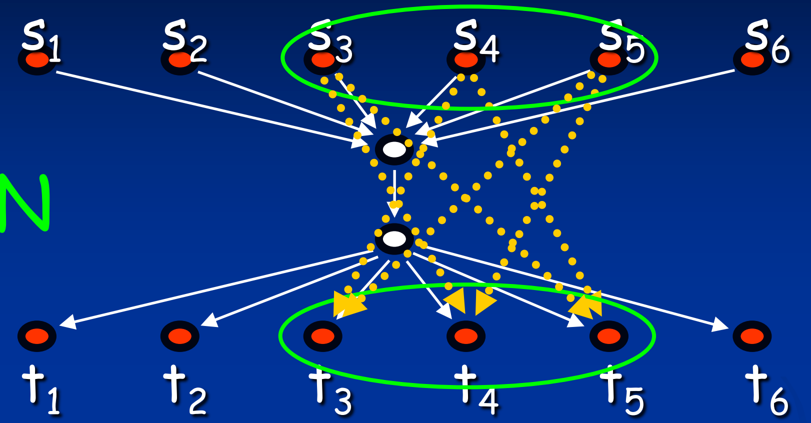
Reduction



- Let G be a graph which can be covered by 3 cliques.
- Implies C(N) is at least 1/3 (time sharing between cliques).
- Key Lemma: From any scalar linear solution to N over F_q with rate = $1/m \geq \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that eventually cover all of G.
- Number of cliques $\leq q^{3/\rho}$: We have a "small" clique cover of G!!

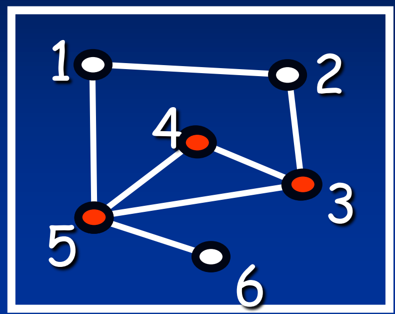
Reduction

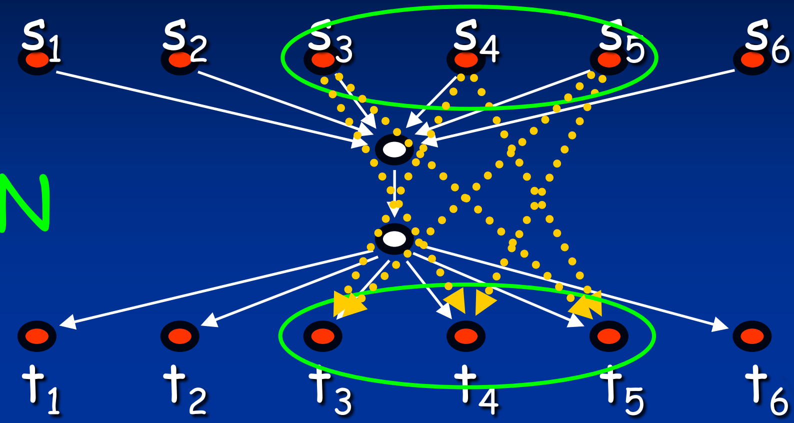
G 

⇒ N 

- Let G be a graph which can be covered by 3 cliques.
- Implies C(N) is at least 1/3 (time sharing between cliques).
- Key Lemma: From any scalar linear solution to N over F_q with rate = $1/m \geq \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that eventually cover all of G.
- Number of cliques $\leq q^{3/\rho}$: We have a "small" clique cover of G!!

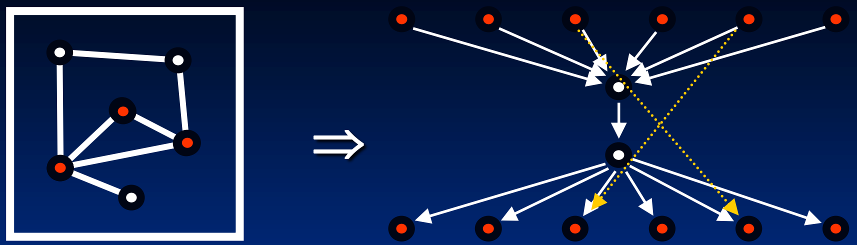
Reduction

G


⇒ N


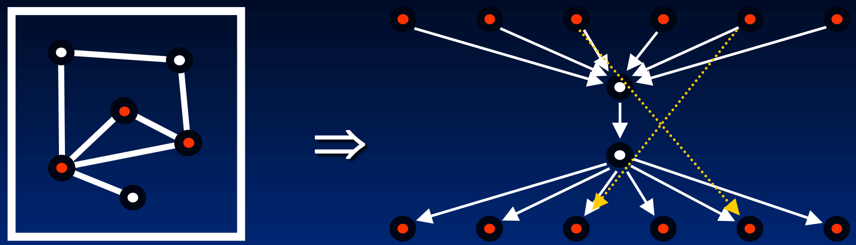
- Let G be a graph which can be covered by 3 cliques.
- Implies $C(N)$ is at least $1/3$ (time sharing between cliques).
- Key Lemma: From any scalar linear solution to N over F_q with rate = $1/m \geq \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that eventually cover all of G.
- Number of cliques $\leq q^{3/\rho}$: We have a "small" clique cover of G!!

Key Lemma



- Key Lemma: From any scalar linear solution to N over F_q with rate = $1/m = \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that eventually cover all of G.

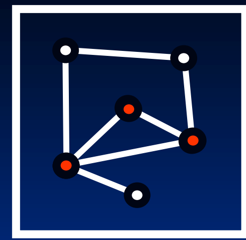
Key Lemma



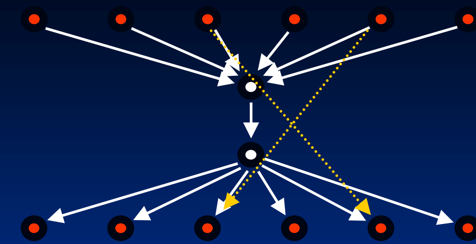
- Key Lemma: From any scalar linear solution to N over F_q with rate = $1/m = \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that eventually cover all of G.

- Consider linear combinations transmitted over bottleneck: $v_1, v_2, v_3, \ldots, v_m$.
  - $v_i = s_1 + s_3 + 2s_5 + s_7$
- NC is feasible, thus each terminal can decode using $\{v_i\}$ and .
- For each terminal let z_i be the combination of v_i's used before side information is used for decoding: $z_i = s_3 + s_5 + 2s_8 + s_9$

Key Lemma



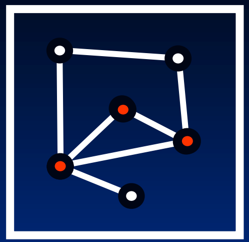
⇒



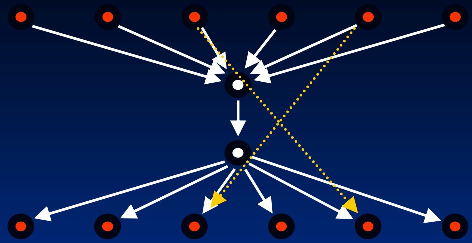
- Key Lemma: From any scalar linear solution to N over F_q with rate = $1/m = \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that eventually cover all of G.

- Consider linear combinations transmitted over bottleneck: $v_1, v_2, v_3, \ldots, v_m$.
  - $v_i = s_1 + s_3 + 2s_5 + s_7$
- NC is feasible, thus each terminal can decode using $\{v_i\}$ and .
- For each terminal let z_i be the combination of v_i's used before side information is used for decoding: $z_i = s_3 + s_5 + 2s_8 + s_9$
- Main observation: if z_i and z_j have same support then there must be an edge in G between vertex i and j. Otherwise side information will not enable decoding.
  - "Equivalence class" in $\{z_i\}$ corresponds to clique in G.
- $\{z_i\}$ includes at most $q^m \leq q^{3/\rho}$ different equivalence classes.
- Implies "small" clique cover.

Key Lemma



⇒

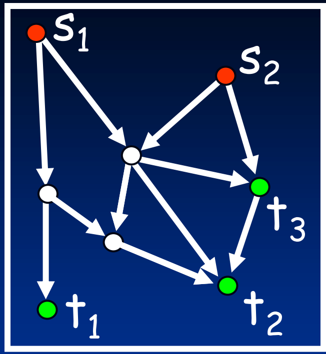


Given undirected G that has a clique cover of size 3, one can efficiently construct a Network Coding instance such that:

- Any Scalar Linear Network Code over F_q with rate $\geq \rho C$ implies a clique cover of G with $q^{3/\rho}$ colors (if q,ρ are constant ⇒ "hard").

- Consider linear combinations transmitted over bottleneck: $v_1,v_2,v_3,\ldots,v_m$.
  - $v_i=s_1+s_3+2s_5+s_7$
- NC is feasible, thus each terminal can decode using $\{v_i\}$ and .
- For each terminal let z_i be the combination of v_i's used before side information is used for decoding: $z_i=s_3+s_5+2s_8+s_9$
- Main observation: if z_i and z_j have same support then there must be an edge in G between vertex i and j. Otherwise side information will not enable decoding.
  - "Equivalence class" in $\{z_i\}$ corresponds to clique in G.
- $\{z_i\}$ includes at most $q^m \leq q^{3/\rho}$ different equivalence classes.
- Implies "small" clique cover.

Recall: scalar Linear



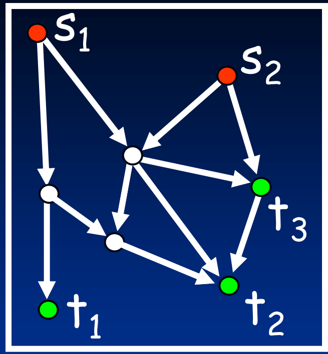
e.g. [Dougherty Freiling Zeger].

- General acyclic NC instance.
- Requirement matrix: $R=[r_{ij}]$
  - $r_{ij} \in \{0,1\}$, $r_{ij}=1 \Leftrightarrow$ terminal t_j wants info. of source s_i.
- Alphabet Σ of size q (here Σ will be a field).
- Scalar Linear: Each source has single character from Σ.
- Objective: design linear network code which is feasible.

- A Network Code enables communication at rate $1/m$:
  - Each link can carry m characters from Σ during transmission.

Capacity: C = maximum rate feasible linear NC.

General capacity



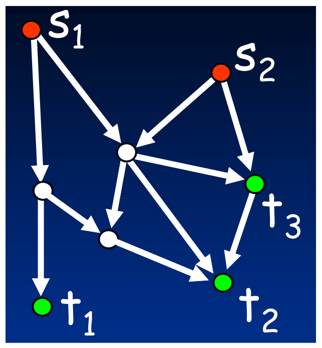
A NC scheme is (k,m) feasible:

- Each source has message of size k characters from Σ.
- Each link can carry m characters from Σ during transmission.
- All requirements satisfied.

- Rate = k/m.

Capacity: C = maximum (k/m) over (k,m) feasible NCs.

Capacity



A NC scheme is (k,m) feasible:

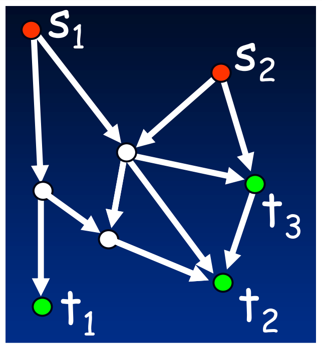
- Each source has message of size k characters from Σ.
- Each link can carry m characters from Σ during transmission.
- All requirements satisfied.

- Rate = k/m.

Capacity: C = maximum (k/m) over (k,m) feasible NCs.

- Scalar Linear NC:
  - Encoding functions are linear AND k=1.
- Vector Linear of dimension k*:
  - Encoding functions are linear AND k =k*.
- General NC:
  - No restrictions!

Capacity



A NC scheme is (k,m) feasible:

- Each source has message of size k characters from Σ.
- Each link can carry m characters from Σ during transmission.
- All requirements satisfied.

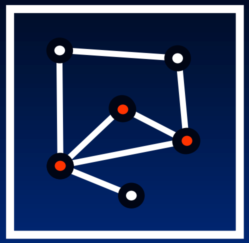
- Rate = k/m.

Capacity: C = maximum (k/m) over (k,m) feasible NCs.

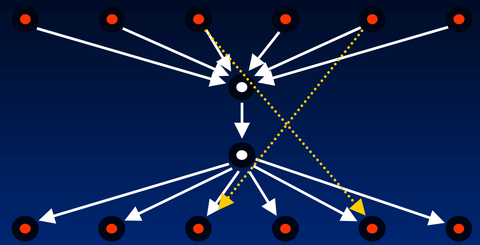
- Scalar Linear NC:
  - Encoding functions are linear AND k=1.
- Vector Linear of dimension k*:
  - Encoding functions are linear AND k =k*.
- General NC:
  - No restrictions!

Single number characterization of the Network Coding instance.

Vector Linear



⇒



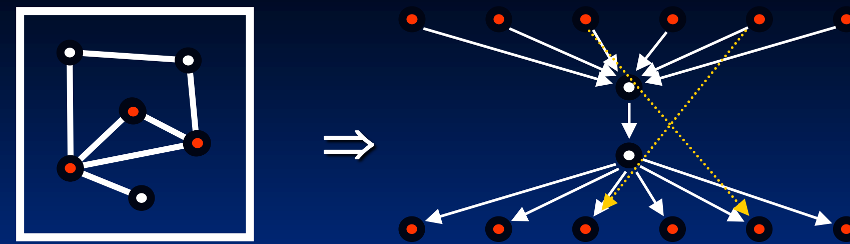
Up to now:

- Exact Scalar linear: NP-Hard.
- Approximate Scalar Linear: "Hard" for const. approx.

- What about Vector Linear?
- Same construction and proof actually hold for Vector Linear.

- Let G be a graph which can be covered by 3 cliques.
- Implies $C(N)$ is at least $1/3$ (time sharing between cliques).
- Key Lemma: From any k-dim. vector linear solution to N over F_q with rate = $k/m \geq \rho \cdot C(N) \geq \rho/3$ one can find "not to many" cliques in G, that cover all of G.
- Number of cliques $\leq q^{3k/\rho}$: We have a "small" clique cover of G!!

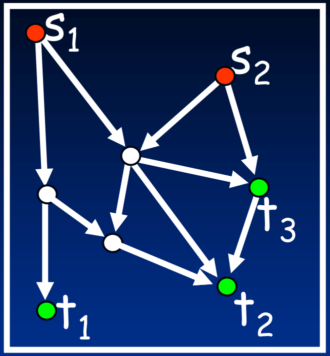
Key Lemma



- Key Lemma: From any scalar linear solution to N with rate = $k/m = \rho \cdot C(N) \geq \rho/3$ one can find "not to many" small cliques in G, that eventually cover all of G.

- Consider linear combinations transmitted: $v_1, v_2, v_3, \ldots, v_m$.
  - $v_1 = s_{11} + s_{32} + 2s_{5k} + s_{74}$
- NC is feasible, thus each terminal can decode using $\{v_i\}$ and $\dashrightarrow$.
- For each terminal let z_i be the combination of v_i's used before side information $\dashrightarrow$ is used: $z_i = s_{32} + s_{55} + 2s_{81} + s_{9k}$
- Main observation: if z_i and z_j have same support in first "generation" then there must be an edge in G between vertex i and j. Otherwise side information will not enable decoding.
  - "Equivalence class" in $\{z_i\}$ corresponds to clique in G.
- $\{z_i\}$ includes at most $q^m \leq q^{3k/\rho}$ different equivalence classes.
- Implies "small" clique cover.

General codes

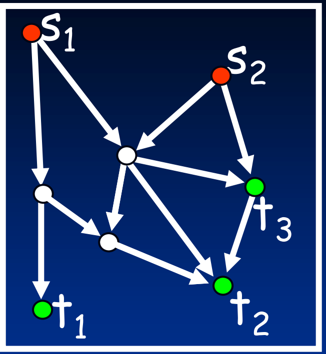


Up to now:

- Exact Scalar Linear: NP-Hard.
- Approximate Scalar Linear: "Hard" for const. approx.
- Approximate Vector Linear: "Hard" for const. approx. and const. dimension.

- What about General Network Coding?

General codes

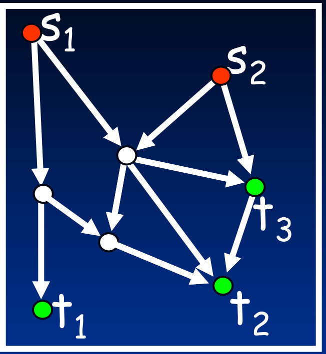


Up to now:

- Exact Scalar Linear: NP-Hard.
- Approximate Scalar Linear: "Hard" for const. approx.
- Approximate Vector Linear: "Hard" for const. approx. and const. dimension.

- What about General Network Coding?
- Define the notion of a "rate vector".

General codes



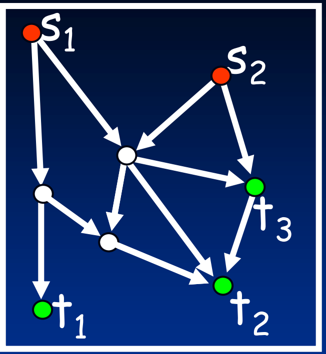
Up to now:

- Exact Scalar Linear: NP-Hard.
- Approximate Scalar Linear: "Hard" for const. approx.
- Approximate Vector Linear: "Hard" for const. approx. and const. dimension.

- What about General Network Coding?
- Define the notion of a "rate vector".

If one can decide whether a certain rate vector is feasible, one can identify "entropic" vectors! [Yeung Li Cai Zhang; Chan Grant].

General codes



Up to now:

- Exact Scalar Linear: NP-Hard.
- Approximate Scalar Linear: "Hard" for const. approx.
- Approximate Vector Linear: "Hard" for const. approx. and const. dimension.

- What about General Network Coding?
- Define the notion of a "rate vector".

If one can decide whether a certain rate vector is feasible, one can identify "entropic" vectors! [Yeung Li Cai Zhang; Chan Grant].

- General coding is necessary to obtain capacity [Dougherty Freiling Zeger].

Summary

- Exact Scalar linear:
  - NP-Hard.
- Approximate Scalar Linear:
  - "Hard" for constant approximation and field size.
- Approximate Vector Linear:
  - "Hard" for constant approximation, field size and dimension.
- General codes:
  - Imply an oracle for the set of entropic vectors.
- Index codes.

- So what else can we do?

Summary

- Exact Scalar linear:
  - NP-Hard.
- Approximate Scalar Li
  - "Hard" for constant app
- Approximate Vector Li
  - "Hard" for constant app
- General codes:
  - Imply an oracle for the
- Index codes.

- So what else can we do?

Open problems:
- Feasibility of General NC
  - Easy or undecidable?
  - Prove NP-Hard.
- Feasibility for Vector Linear.
  - Prove NP-Hard for any dim.

Summary

- Exact Scalar linear:
  - NP-Hard.
- Approximate Scalar l
  - "Hard" for constant a
- Approximate Vector
  - "Hard" for constant a
- General codes:
  - Imply an oracle for th
- Index codes.

- So what else can we do?

The "positive" angle:

- Special networks:
  - Topology: Ring, line, star, planar
  - Few source/sink pairs.
- Loose inner and outer bounds that are computable.

[Yazdi Savari Kramer] [Savari Kramer] [Adler Harvey Jain Kleinberg Lehman] [Erez and Feder] [L S] [Ngai and Yeung] [Adler Harvey Jain Kleinberg Lehman] [Jain Vazirani Yeung Yuval] [Kramer Savari] [Yeung Li Cai Zhang] [Yan Yeung Zhang] [L Medard] [Thakor Grant Chan] [Bakshi Effros Gu Koetter] …

Summary

- Exact Scalar linear:
  - NP-Hard.
- Approximate Scalar l
  - "Hard" for constant a
- Approximate Vector
  - "Hard" for constant a
- General codes:
  - Imply an oracle for th
- Index codes.

- So what else can we do?

The "positive" angle:
- Special networks:
  - Topology: Ring, line, star, planar
  - Few source/sink pairs.
- Loose inner and outer bounds that are computable.

[Yazdi Savari Kramer] [Savari Kramer] [Adler Harvey Jain Kleinberg Lehman] [Erez and Feder] [L S] [Ngai and Yeung] [Adler Harvey Jain Kleinberg Lehman] [Jain Vazirani Yeung Yuval] [Kramer Savari] [Yeung Li Cai Zhang] [Yan Yeung Zhang] [L Medard] [Thakor Grant Chan] [Bakshi Effros Gu Koetter] …

Thanks!