Network Coding for Line Networks with Broadcast Channels

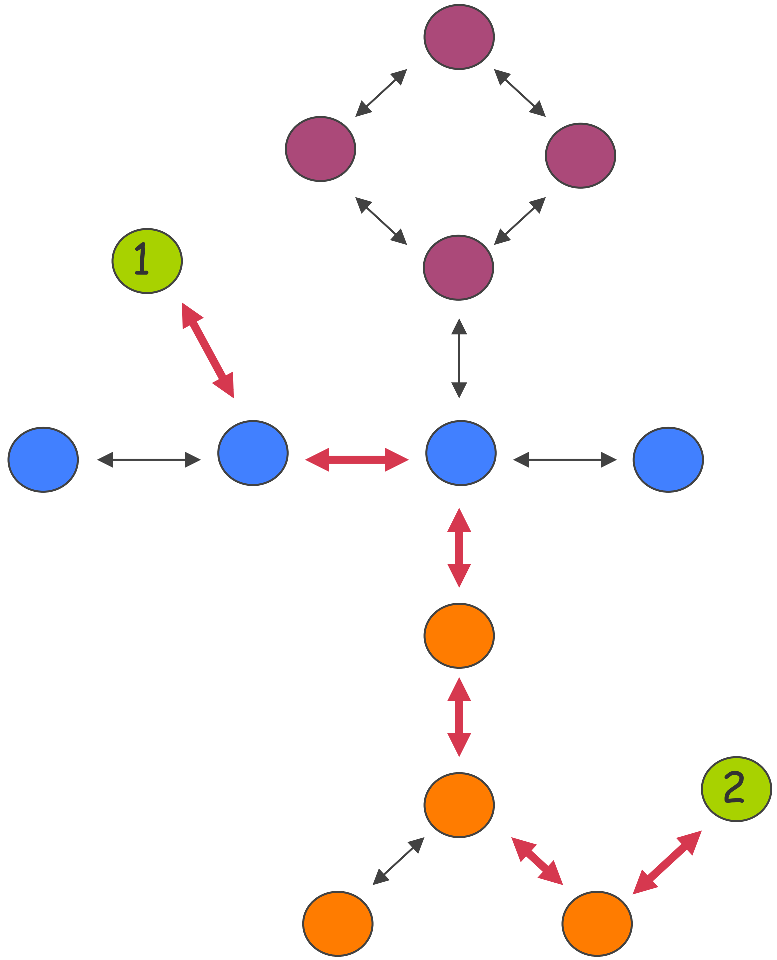
Gerhard Kramer* and Sadegh Tabatabaei Yazdi**
*Technische Universität München, Germany
**Corporate R&D, Qualcomm, San Diego, CA

Talk at the 2014 Sino-German Workshop
Shenzhen, China
March 4, 2014

Unterstützt von / Supported by
Alexander von Humboldt
Stiftung / Foundation

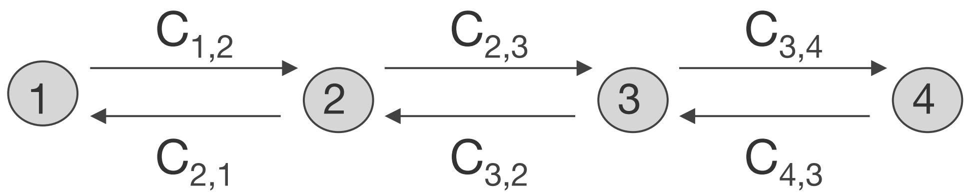
Network Communication

- Network: represent by a graph
- Example: network of 3 networks
- **Line**, e.g., an ethernet bus
  - Single-path routing common for simplicity, control, security
- Star, e.g., a radio network cell
  - Hub controls and monitors
  - Scalable, can isolate failures
- Ring, e.g., an optical network
  - Two paths protect against failures

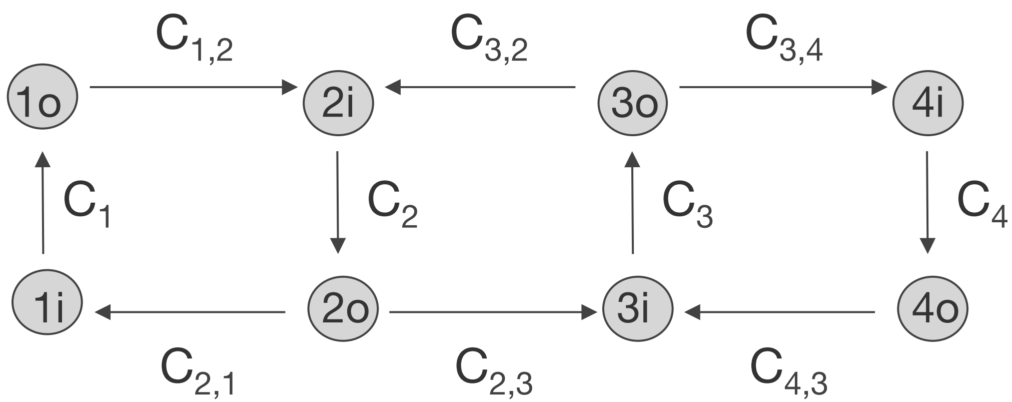


1) Models

- Wireline: edge capacity constraints $C_{u,v}$ for edge (u,v)

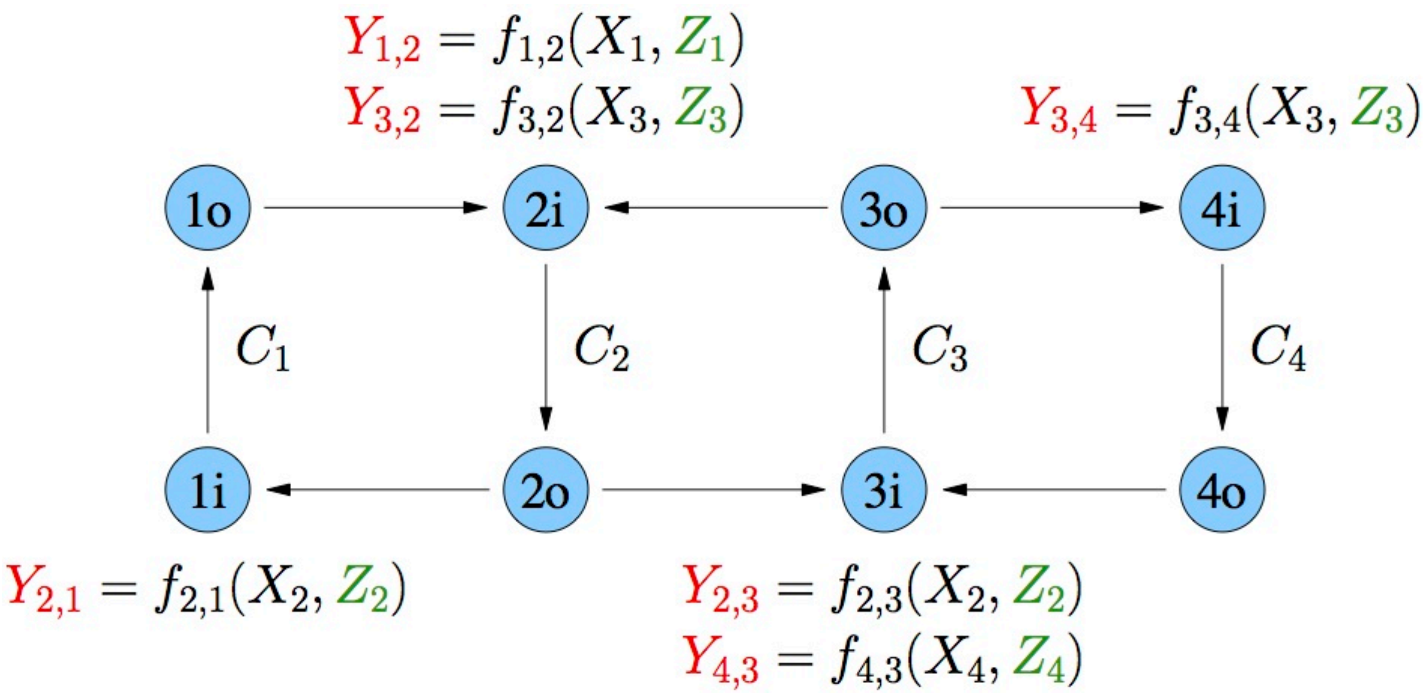


- Nodes can also be bottlenecks, e.g., processor energy, speed, bus bandwidth constraints. Capacity C_u for node u



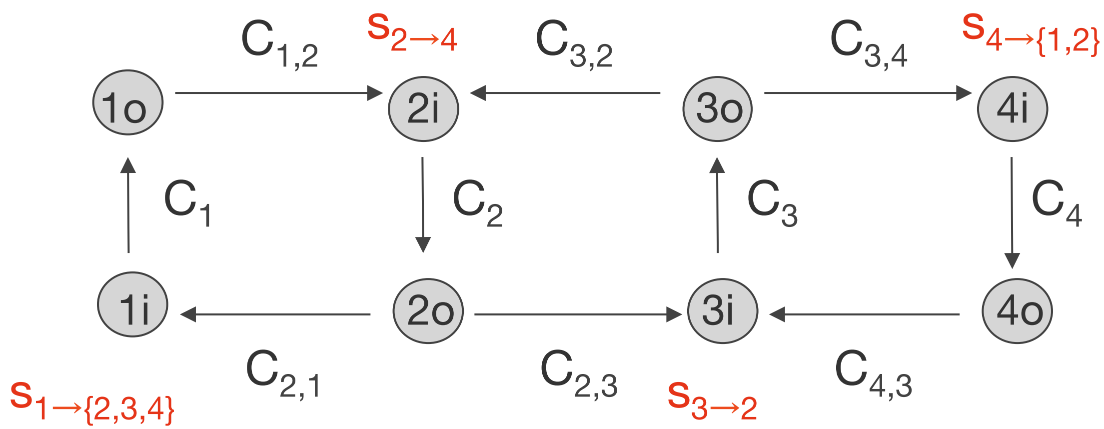
Wireless with Broadcast Channels (BCs)

- Broadcast constraint via X_u rather than $X_{u,u-1}$ and $X_{u,u+1}$
- Time-frequency slots: no interference
- General: add interference via Y_u rather than $Y_{u-1,u}$ and $Y_{u+1,u}$



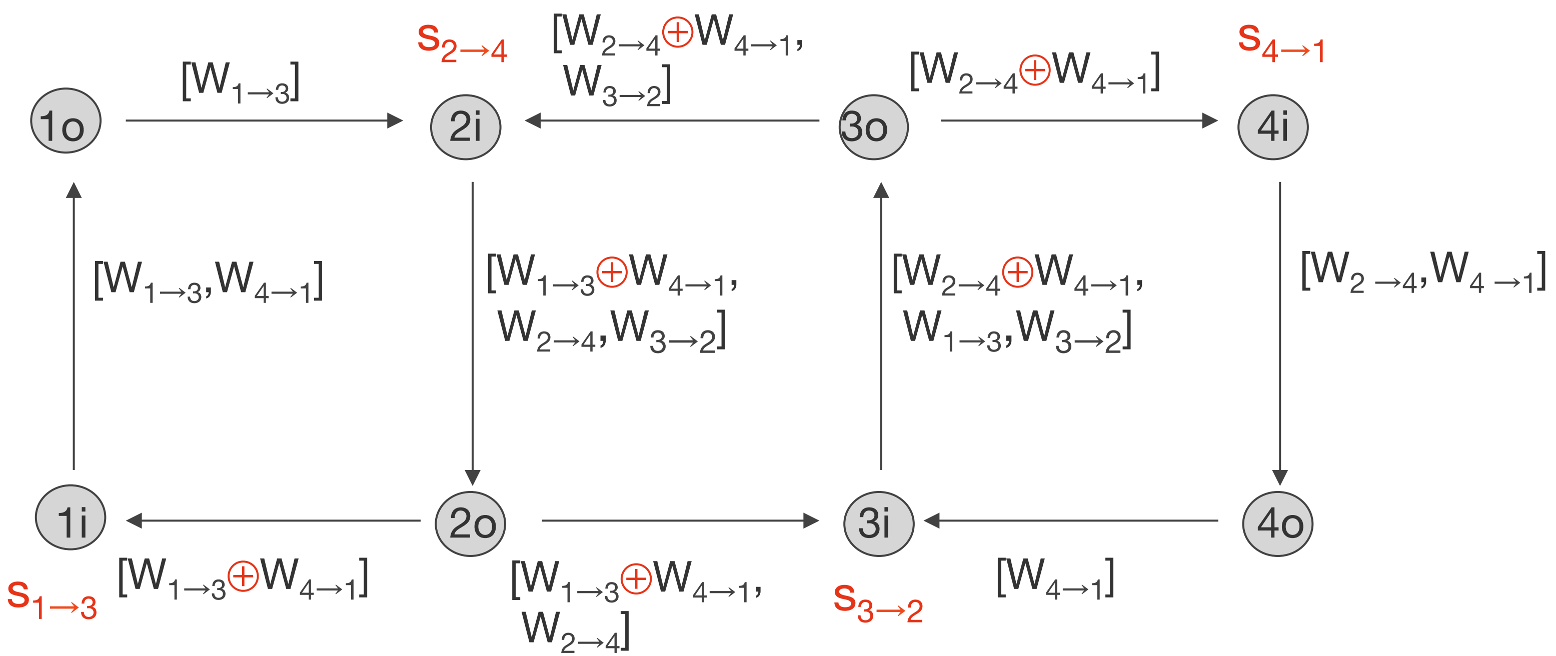
Traffic Sessions

- Traffic Sessions: $u \rightarrow D(u)=\{v(1),\ldots,v(L)\}$, rate $R(u \rightarrow D(u))$
  - Unicast: up to n(n-1) sessions between node-pairs
  - Broadcast: n sessions (one node to all other nodes)
  - Multicast: $n(2^{n-1}-1)$ sessions (one node u to a node set D(u))
- Node constraints: can place sources & sinks at different sub-nodes for different problems



2) Wireline: How to Communicate?

- Network coding helps: many "butterflies"
- Guess: Routing, copying, and "butterfly" binary linear network coding is optimal. For equal-length packets:

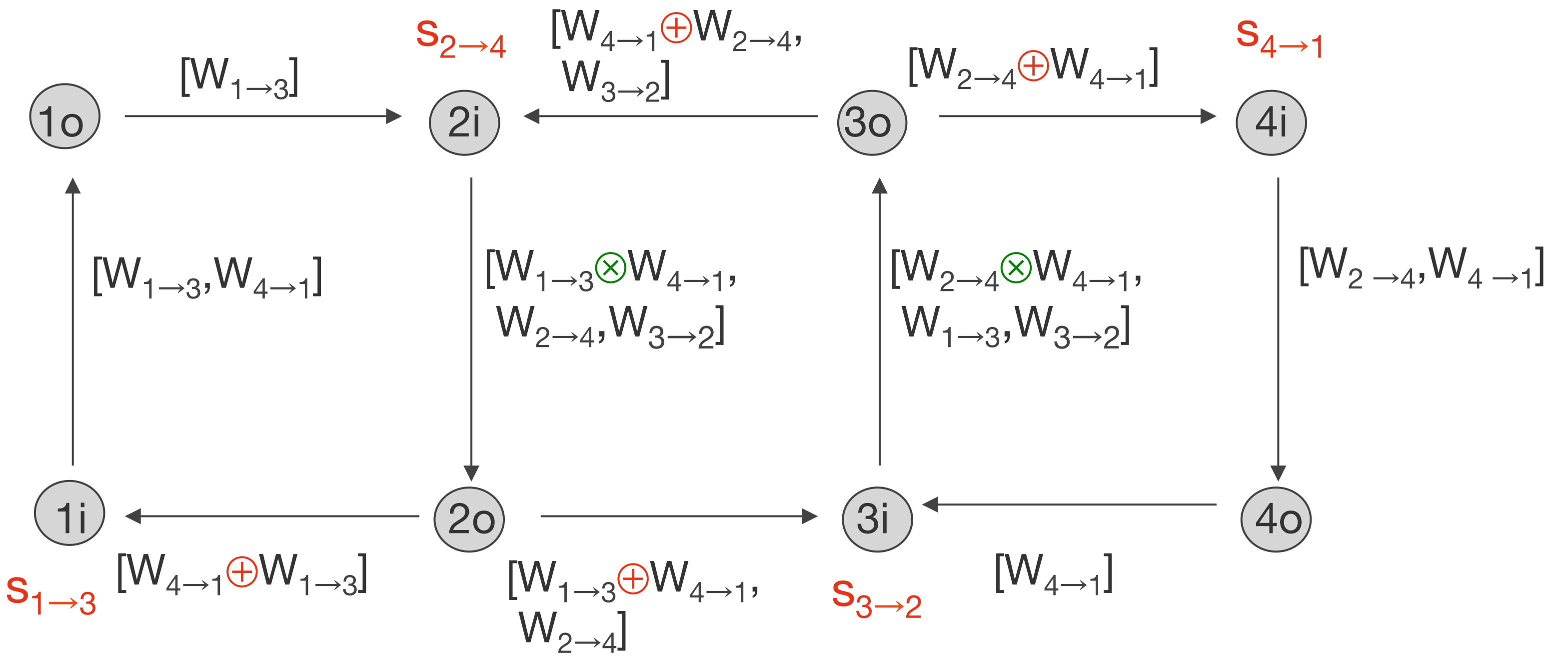


Non-uniform Packet Lengths and Rates

- Let:

$$A \oplus B = \begin{cases} [A_1 \oplus B_1, \ldots, A_m \oplus B_m] & m \le n \\ [A_1 \oplus B_1, \ldots, A_n \oplus B_n, A_{n+1}, \ldots, A_m] & m > n \end{cases}$$

$$A \otimes B = \begin{cases} [A_1 \oplus B_1, \ldots, A_m \oplus B_m, B_{m+1}, \ldots, B_n] & m \le n \\ [A_1 \oplus B_1, \ldots, A_n \oplus B_n, A_{n+1}, \ldots, A_m] & m > n \end{cases}$$



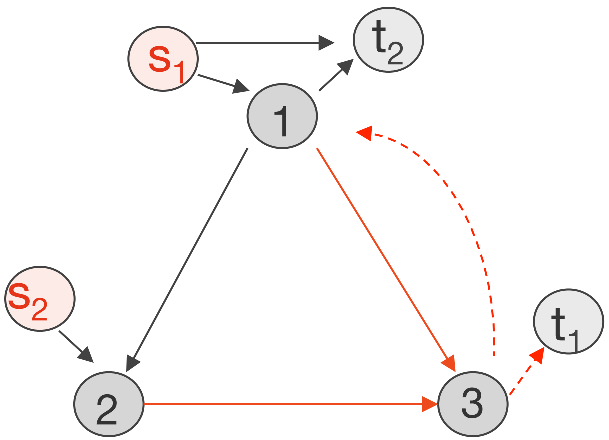
Notes

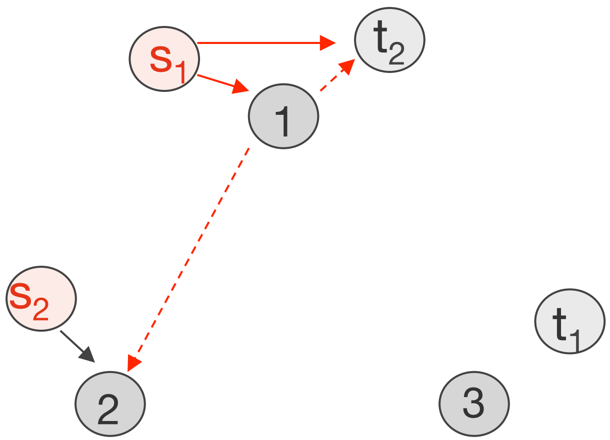
- Method seems simple but requires careful control. Each node u treats 8 sets of messages differently
  1) Left-to-right (LR) messages through node u
  2) Right-to-left (RL) messages through node u
  3) Left-to-right (LRu) messages also destined for u
  4) Right-to-left (RLu) messages also destined for u
  5) L-to-R and R-to-L messages “stopping” at node u (u)
  6) Node u messages going to left and right (u,LR)
  7) Node u messages going to right (u,R)
  8) Node u messages going to left (u,L)

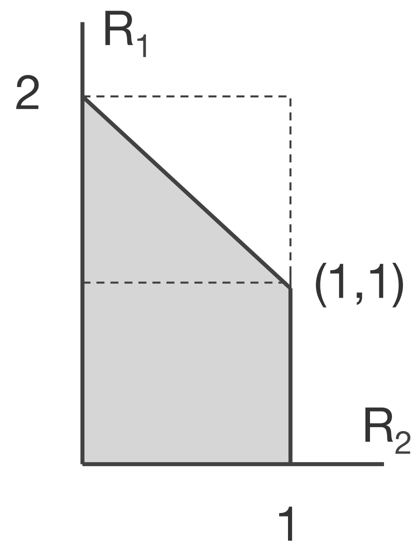
- Converse:
  - Classic cut bounds insufficient
  - Progressive edge-cut bounds give the capacity (and include classic cut bounds)

3) Progressive Edge Cuts (Kramer-Savari '06)

- Consider a general edge set E and session set S
- Initialize: remove (1) edges in E; (2) edges of sources not in S; (3) edges out of nodes directed-sense[1] disconnected from S
- Repeat: test if an s in S is undirected-sense[2] disconnected from any of its sinks. If so, remove s and then edges out of nodes directed-sense[1] disconnected from the remaining sources.
- Successful removal of all sources: $\Sigma_{k \in S} R_k \leq \Sigma_{e \in E} C_e$
- Example: $E=\{(1,3),(2,3)\}$ and $S=\{s_1,s_2\}$: $\mathbf{R_1+R_2 \leq 2}$ if $C_e=1$ for all e







[1]based on functional dependencies
[2]based on fd-separation in functional dependence graphs (Kramer '98)

Line Network Rate Constraints

- **Edges:** get basic routing rates (cf. classic cut-set bound)

$$\Sigma_{i=1..u}\ \Sigma_{D(i)\text{ with a node in }\{u+1..n\}}\ R(i \rightarrow D(i)) \leq C_{u,u+1} \qquad (L\rightarrow R)$$

$$\Sigma_{i=u..n}\ \Sigma_{D(i)\text{ with a node in }\{1..u-1\}}\ R(i \rightarrow D(i)) \leq C_{u,u-1} \qquad (R\rightarrow L)$$

- **Nodes:** node u incoming and outgoing rates <u>plus</u> max(L→R rates, R→L rates) (see graph on p. 8):

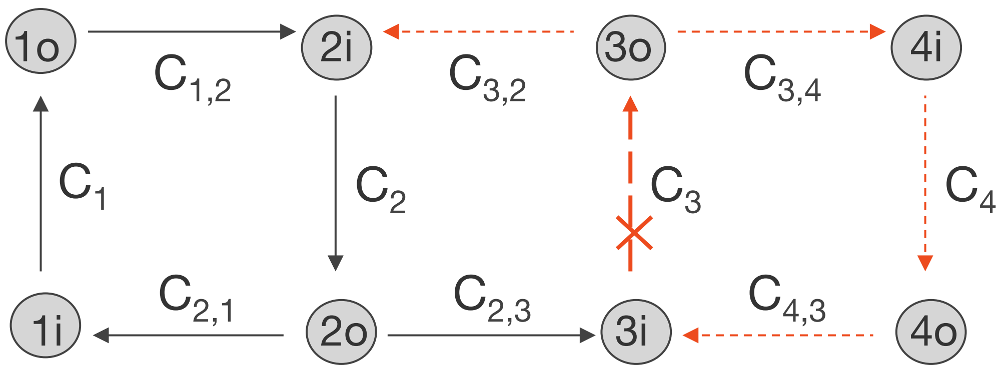
$$\Sigma_{D(u)}\ R(u \rightarrow D(u)) + \Sigma_{v}\ \Sigma_{\text{Traffic stops at }u}\ R(v \rightarrow D(v))$$

$$+ \max\Big(\Sigma_{i=1..u-1}\ \Sigma_{D(i)\text{ with a node in }\{u+1..n\}}\ R(i \rightarrow D(i)),$$

$$\Sigma_{i=u+1..n}\ \Sigma_{D(i)\text{ with a node in }\{1..u-1\}}\ R(i \rightarrow D(i))\Big) \leq C_u$$

Application of Edge Cuts to Lines

- Bound for node u: choose E={(ui,uo)} and S={L→R sources across u} U {u incoming and outgoing sources}
- Example: u=3 with E={(3i,3o)}
  - Remove (3i,3o); s right of node 3 and s left of node 3 having sinks left of node 3 only; edges right of u and (3o,2i)
  - Can remove all sources <u>including</u> node 3 outgoing sources
  - Gives new (non-classic) L→R bounds; similarly get new R→L bounds; these bounds, combined with the classic cut bounds, define the multiple-multicast capacity region



4) What about Wireless?

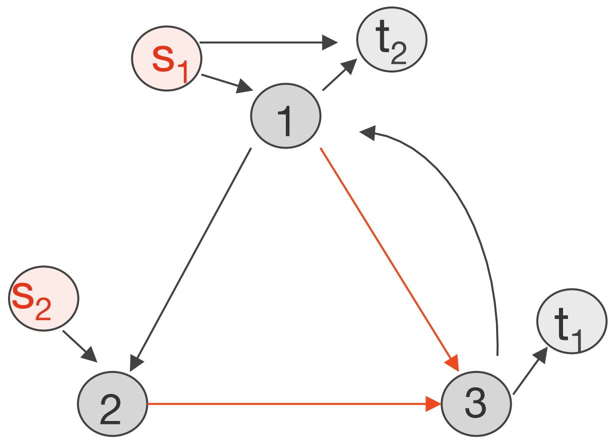
- Problem: capacity of BCs with feedback is unknown
- Partial resolution: capacity is known for some cases
  - orthogonal channels
  - deterministic channels
  - physically degraded channels, including physically degraded Gaussian BCs [El Gamal, 1978)
- Do the coding/converse methods extend to our networks?
- Answer: yes! See our paper "Network coding for line networks with broadcast channels," Entropy, vol. 14, 2012
- Paper gives a general achievable region, and converses for the above cases and for packet erasure channels

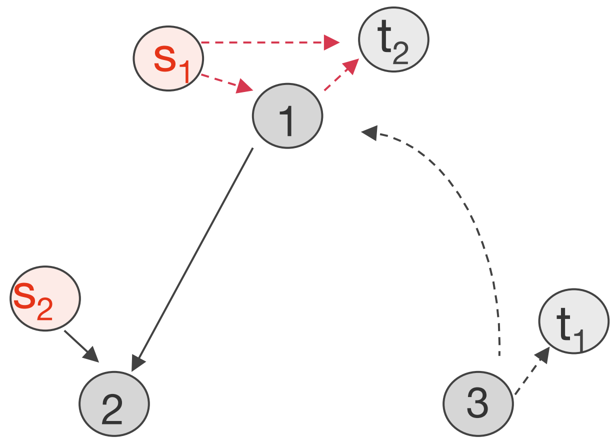
Notes: the Progressive Edge Cut Tool

- Includes classic cuts as special cases
- Applies to network coding (a classic edge-cut bound does not)
- Generalizes naturally to wireless networks to include any coding
- Wireless example below: using E={(1,3),(2,3)} and S={1,2} gives

$$R_1 \le \min[I(X_1;Y_2Y_3|X_2X_3), I(X_1X_2;Y_3|X_3)]$$
$$R_2 \le \min[I(X_3;Y_1|X_1), I(X_2;Y_3|X_1X_3)]$$
} Classic cuts

$$R_1+R_2 \le I(X_1X_2;Y_3|X_3)$$
← Progressive cut





Summary

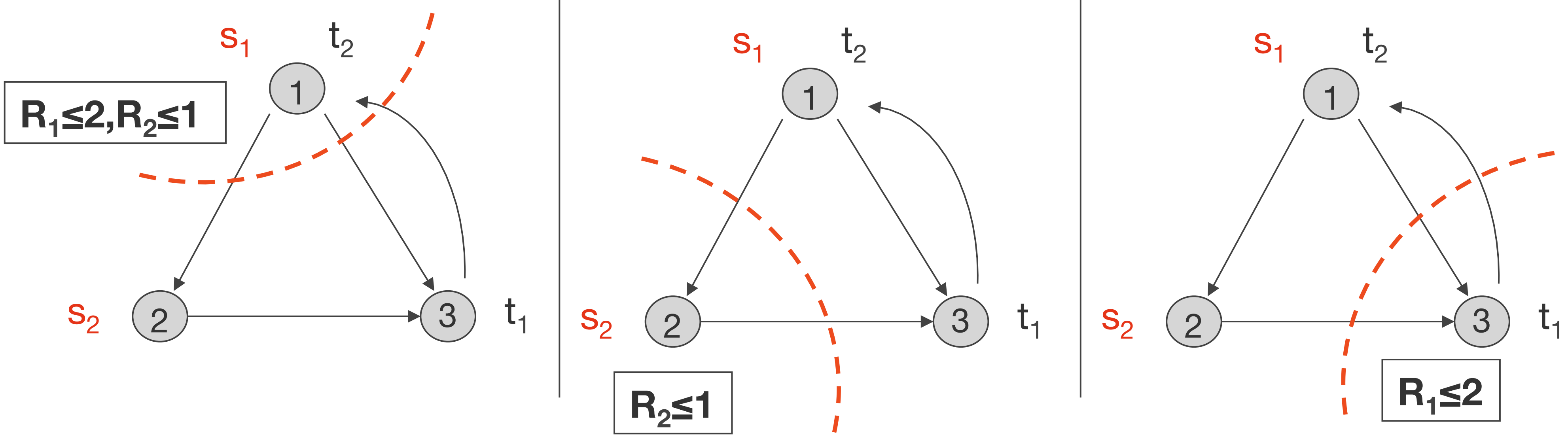
Line Networks:

- even wireline problems require careful coding and have sophisticated capacity regions;
- ideas extend to certain broadcasting scenarios;
- for general BCs: we first need the capacities of BCs with (generalized) feedback;
- including interference will be even tougher!

Extra Slides

Classic Cut-Set Bound

- Partition nodes into two sets N and N^C
- Let S be the set of sessions originating in N with a sink in N^C
- Cut E is the set of edges starting in N and ending in N^C
- Classic cut bound: $\sum_{k \varepsilon S} R_k \leq \sum_{e \varepsilon E} C_e$
- Example: ring with 2 unicast sessions and unit-edge capacities.
  We have: $\mathbf{R_1 \leq 2, R_2 \leq 1}$



Line Networks with Edge Constraints Only

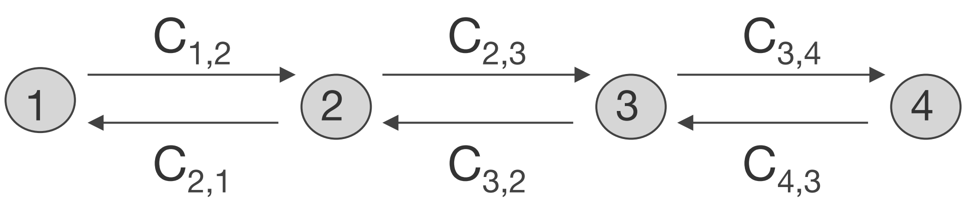
- Routing: bounds for (u,u+1) and (u+1,u):

  $\sum_{i=1..u} \sum_{D(i) \text{ with a node in } \{u+1..n\}} R(i \to D(i)) \leq C_{u,u+1}$ (L→R)

  $\sum_{i=u+1..n} \sum_{D(i) \text{ with a node in } \{1..u\}} R(i \to D(i)) \leq C_{u+1,u}$ (R→L)

- Classic cut-set bound
  - For cut {(u,u+1)} is just (L→ R)
  - For cut {(u+1,u)} is just (R→ L)

- So routing (+ copying for multicast) is rate-optimal



fd-Separation (Kramer '98)

- Let A, B and C be vectors whose entries are RVs (vertices) of a FDG
- Success after the following implies I(A ; B | C) = 0 (cf. Pearl 1988)
  - Consider only vertices and edges met when moving backward from the vertices in A, B, or C ("causality")
  - Remove the outgoing edges of vertices disconnected from the sources in a directed sense
  - Check if there is no undirected path from "A" to "B"

- Ex: $I(W_1 ; \hat{W}_1 \mid Y_{2,3}\ Y_{1,3}\ Z_{3,1}) = 0$

  $I(W_2 ; \hat{W}_2 \mid Y_{2,3}\ Y_{1,3}\ Z_{3,1}\ W_1) = 0$