Linear flow equations for network coding in the multiple unicast case

Niranjan Ratnakar

Joint work with Ralf Koetter, Tracey Ho

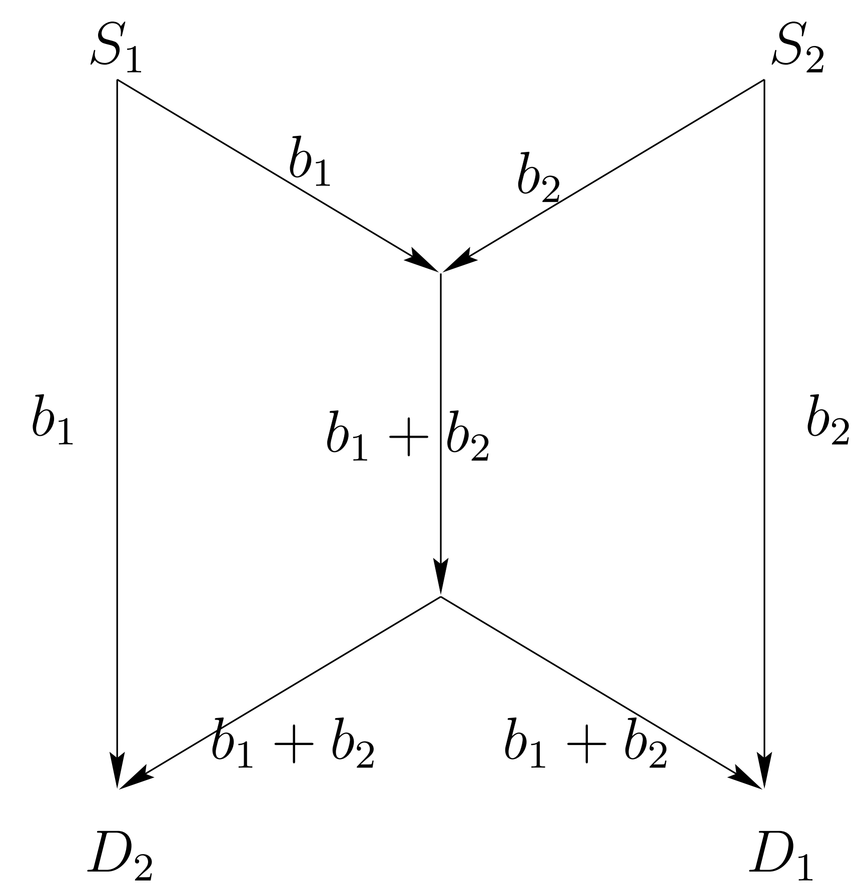
University of Illinois

January 27, 2005

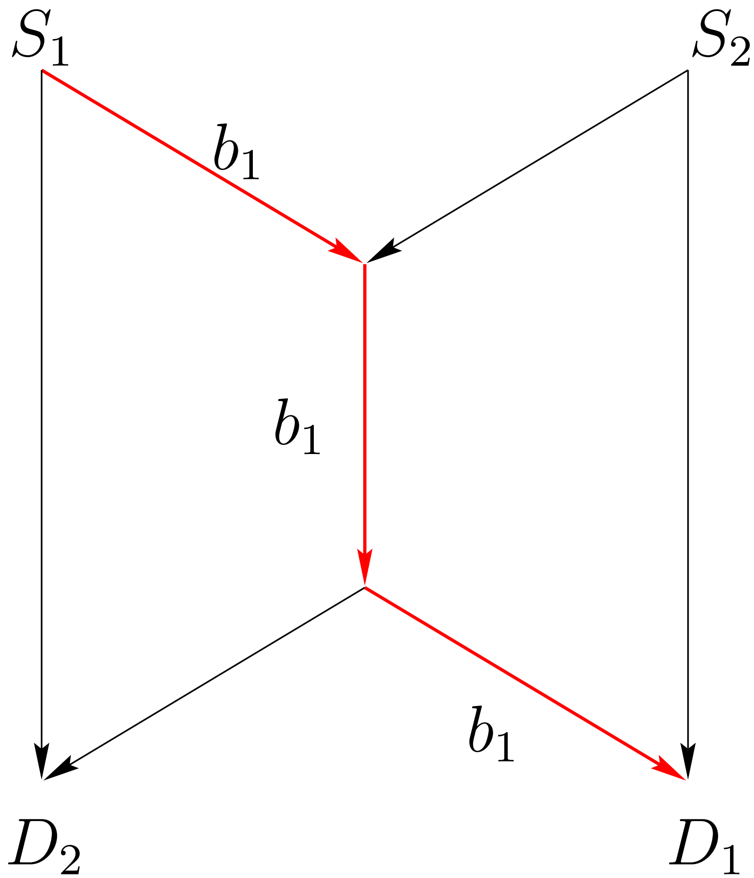
Outline

- Butterfly structure and some observations.
- Generalizing the lessons from the butterfly case.
- Linear equations for general networks.
- Extensions.

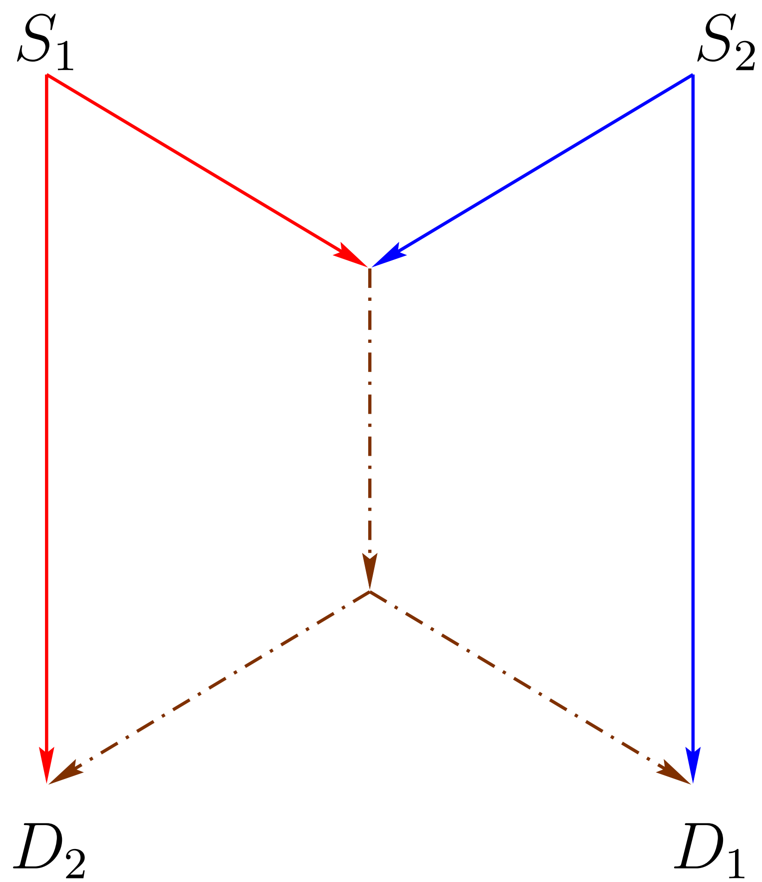
Butterfly Network



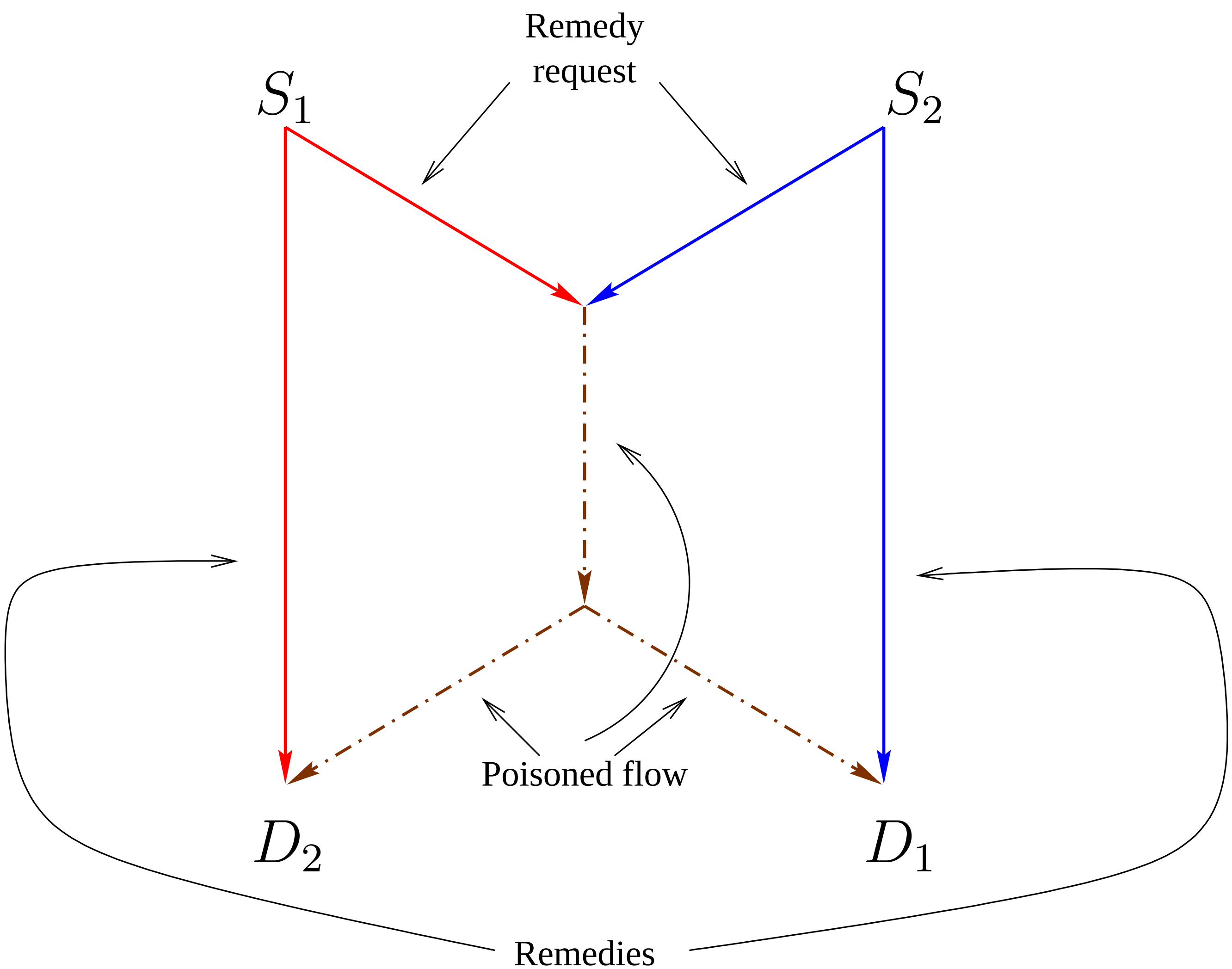
Butterfly Network



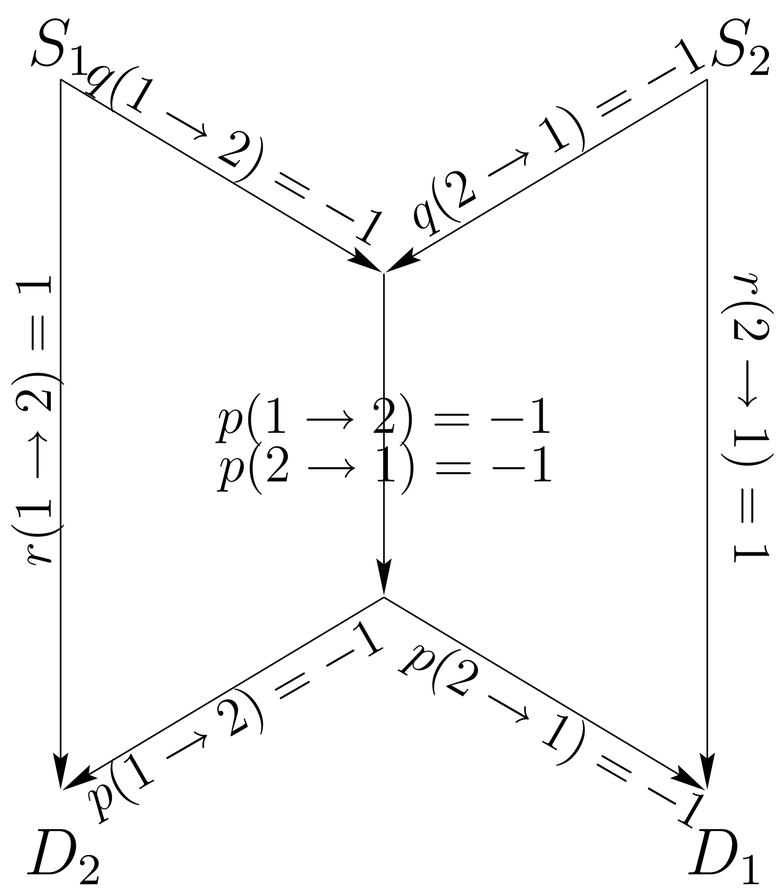
Butterfly Network



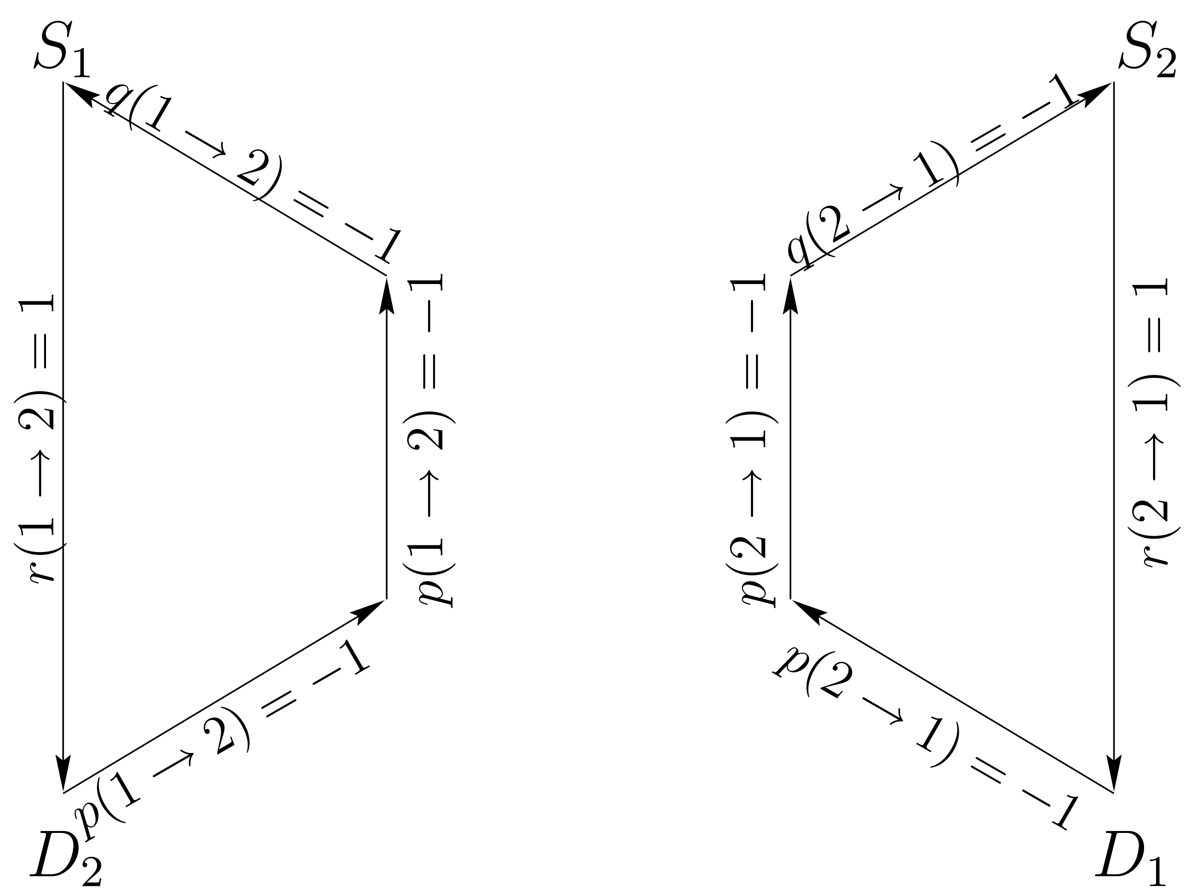
Butterfly Network



Butterfly Network



Butterfly Network



Observations

- $x(1)$ is a flow from S_1 to D_1.
- $x(2)$ is a flow from S_2 to D_2.

Observations

- $x(1)$ is a flow from S_1 to D_1.
- $x(2)$ is a flow from S_2 to D_2.
- $p(1 \longrightarrow 2) + q(\cdot) + r(\cdot)$ forms a loop.

Observations

- $x(1)$ is a flow from S_1 to D_1.
- $x(2)$ is a flow from S_2 to D_2.
- $p(1 \longrightarrow 2) + q(\cdot) + r(\cdot)$ forms a loop.
- $p(1 \longrightarrow 2)$ is a 'virtual' flow with a 'host' $x(2)$.
- $q(1 \longrightarrow 2)$ is a 'virtual' flow with a 'host' $x(1)$.
- $r(1 \longrightarrow 2)$ is a flow that consumes resources and does not need a host.

Observations

- $x(1)$ is a flow from S_1 to D_1.
- $x(2)$ is a flow from S_2 to D_2.
- $p(1 \rightarrow 2) + q(\cdot) + r(\cdot)$ forms a loop.
- $p(1 \rightarrow 2)$ is a 'virtual' flow with a 'host' $x(2)$.
- $q(1 \rightarrow 2)$ is a 'virtual' flow with a 'host' $x(1)$.
- $r(1 \rightarrow 2)$ is a flow that consumes resources and does not need a host.
- $p(\cdot), q(\cdot)$, and $r(\cdot)$ are unbroken paths.

Generalizing the idea

- $p_e(m \to n, u)$ for every edge e. u keeps track of the 'origin' of the poison.
- Similarly $q_e(m \to n, u)$ and $r_e(m \to n, u)$.
- We will search for butterfly structures.

Searching for Butterfly structures

Allow loops made of $p(m \to n, u)$, $q(m \to n, u)$, and $r(m \to n, u)$. For all nodes v, u, and for all flows m and n.

$$\sum_{e:\text{head}(e)=v} p_e(m \to n, u) + q_e(m \to n, u) + r_e(m \to n, u)$$

$$= \sum_{e:\text{tail}(e)=v} p_e(m \to n, u) + q_e(m \to n, u) + r_e(m \to n, u)$$

Searching for Butterfly structures

Ensure that each of $p(m \to n, u)$, $q(m \to n, u)$, and $r(m \to n, u)$ is an unbroken path. At node u

$$\sum_{e:\text{head}(e)=u} q_e(m \to n, u) \geq \sum_{e:\text{tail}(e)=u} q_e(m \to n, u) \qquad (1)$$

At any other node v,

$$\sum_{e:\text{head}(e)=v} q_e(m \to n, u) \leq \sum_{e:\text{tail}(e)=v} q_e(m \to n, u) \qquad (2)$$

Searching for Butterfly structures

Ensure that each of $p(m \to n, u)$, $q(m \to n, u)$, and $r(m \to n, u)$ is an unbroken path. At node u

$$\sum_{e:\mathrm{head}(e)=u} p_e(m \to n, u) \leq \sum_{e:\mathrm{tail}(e)=u} p_e(m \to n, u) \qquad (3)$$

At any other node v,

$$\sum_{e:\mathrm{head}(e)=v} p_e(m \to n, u) \geq \sum_{e:\mathrm{tail}(e)=v} p_e(m \to n, u) \qquad (4)$$

Searching for Butterfly structures

- If m-th and n-th flows overlap, "generate" poison (and consequently the loops).
- Ensure that a maximum of two flows are overlapping. This ensures that the butterfly structures are disjoint.

List of equations

$x_e(n)$ is a flow of the desired rate from S_n to D_n.

$$p_e(n \rightarrow m, u) = p_e(m \rightarrow n, u) \text{ if } \mathrm{tail}(e) = u \tag{5}$$

$$\begin{aligned}\sum_u \sum_m \max(p_e(m \rightarrow n, u), p_e(n \rightarrow m, u)) + \sum_{i=1}^{n} x_e(i) \quad &\leq \quad z_e \\ + \sum_u \sum_m (r_e(m \rightarrow n, u) + r_e(n \rightarrow m, u)) & \end{aligned} \tag{6}$$

Virtual hosts

$$x_e(n) + \sum_u \sum_m p_e(m \to n, u) + q_e(m \to n, u) \geq 0 \qquad (7)$$

A solution to these equations can be used to identify the butterfly structures and a network coding solution can be computed.

Extensions

- Investigate the packing of more complicated structures.
- Allow for multiple poisoning (might need coding over greater field sizes).
- Investigate the performance of the solution on practical networks.