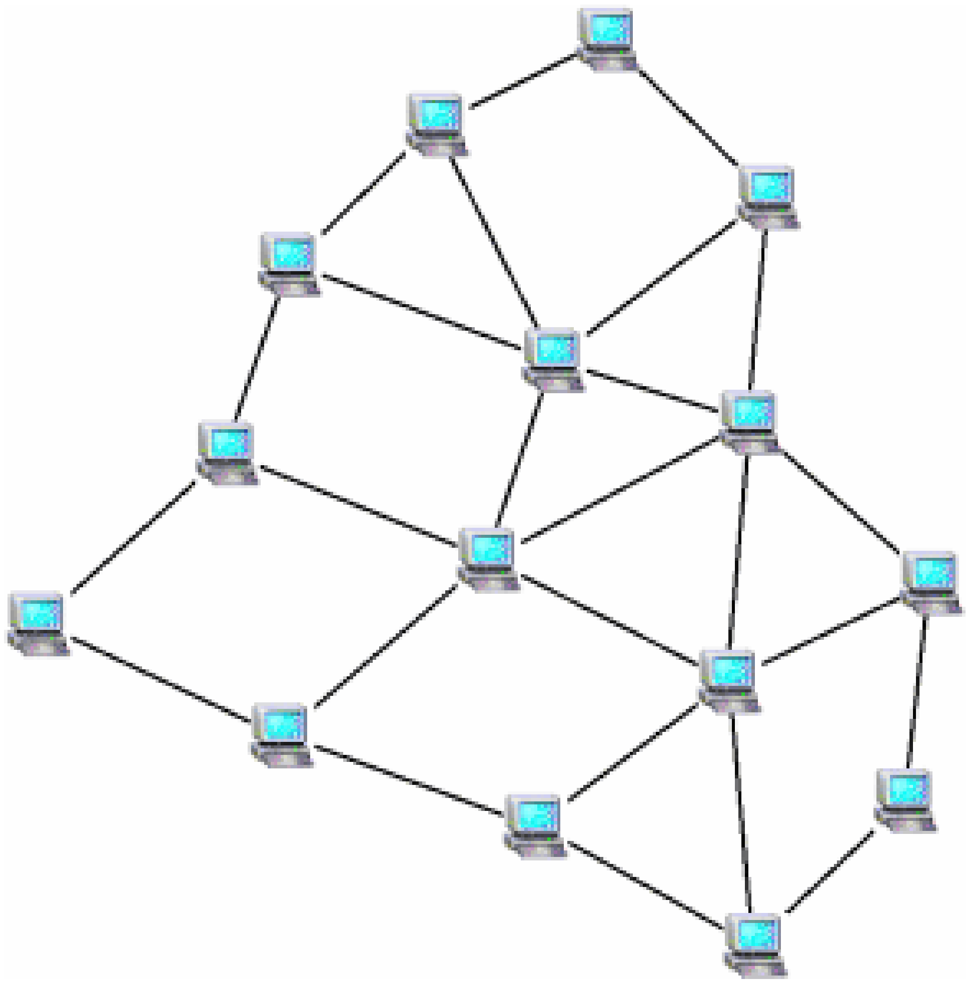
A Brief Introduction to Network Coding

Terry Ferrett

Lane Department of
Computer Science
and
Electrical Engineering

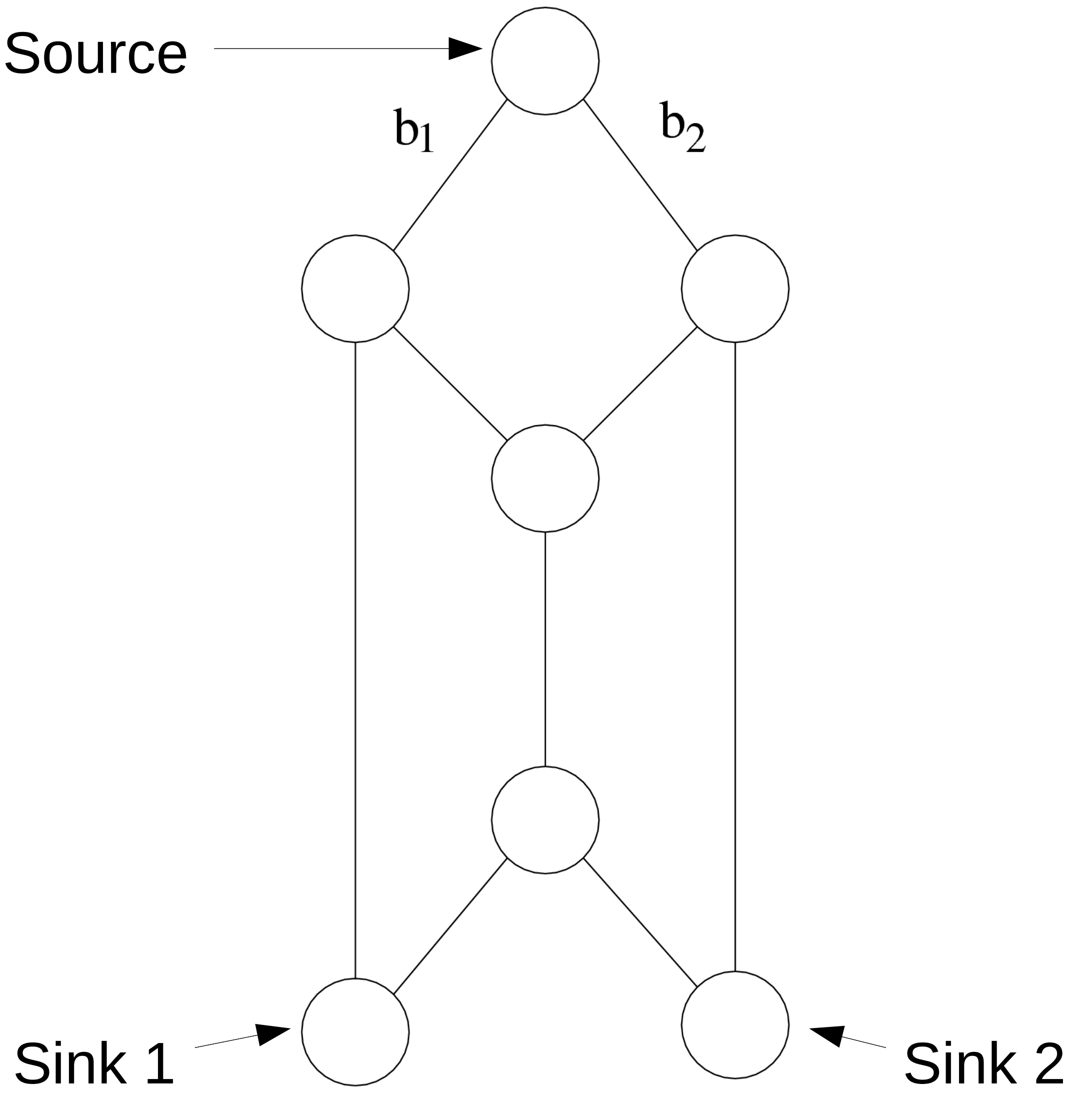
West Virginia University

July 22, 2009

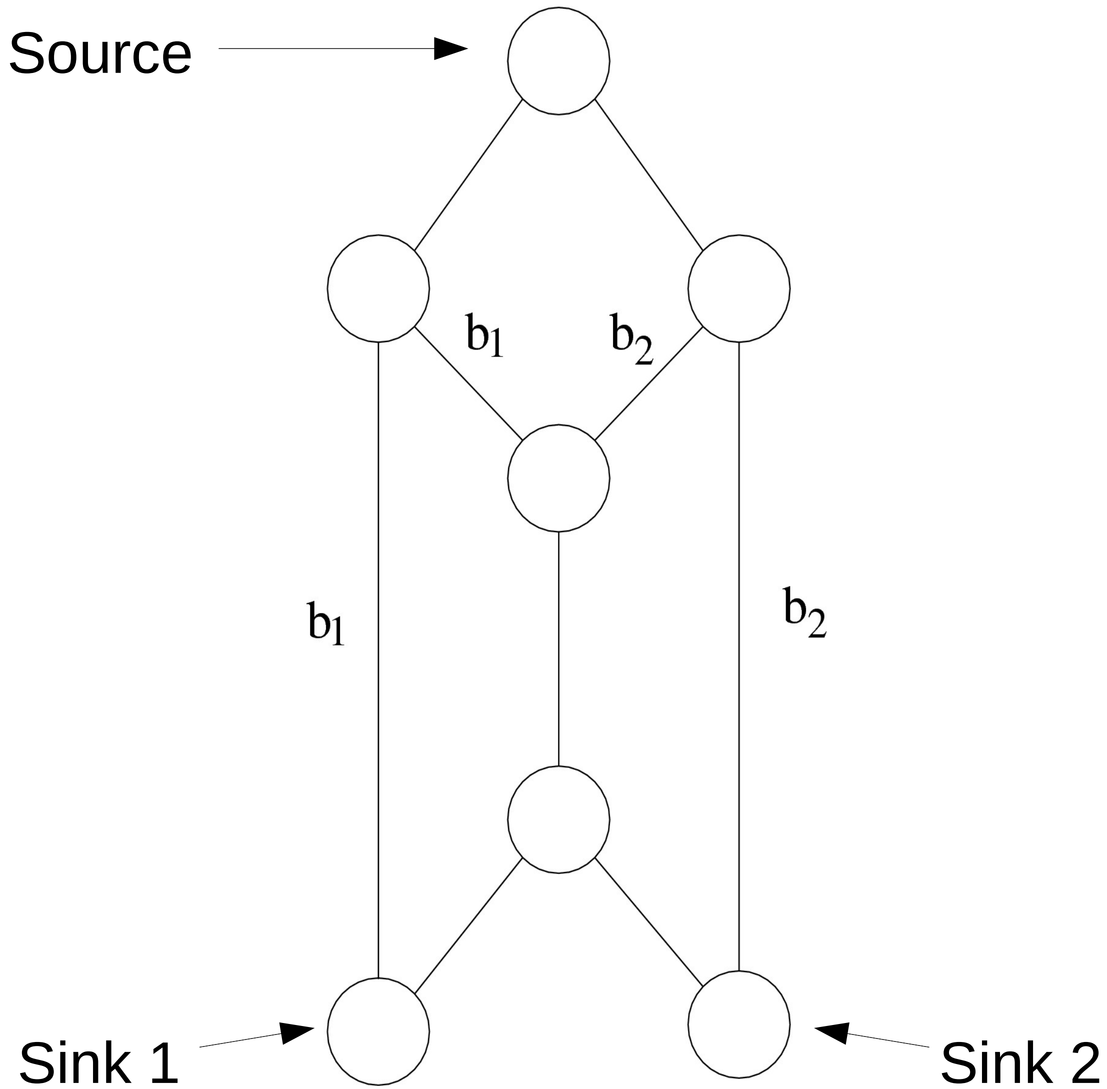


West Virginia University

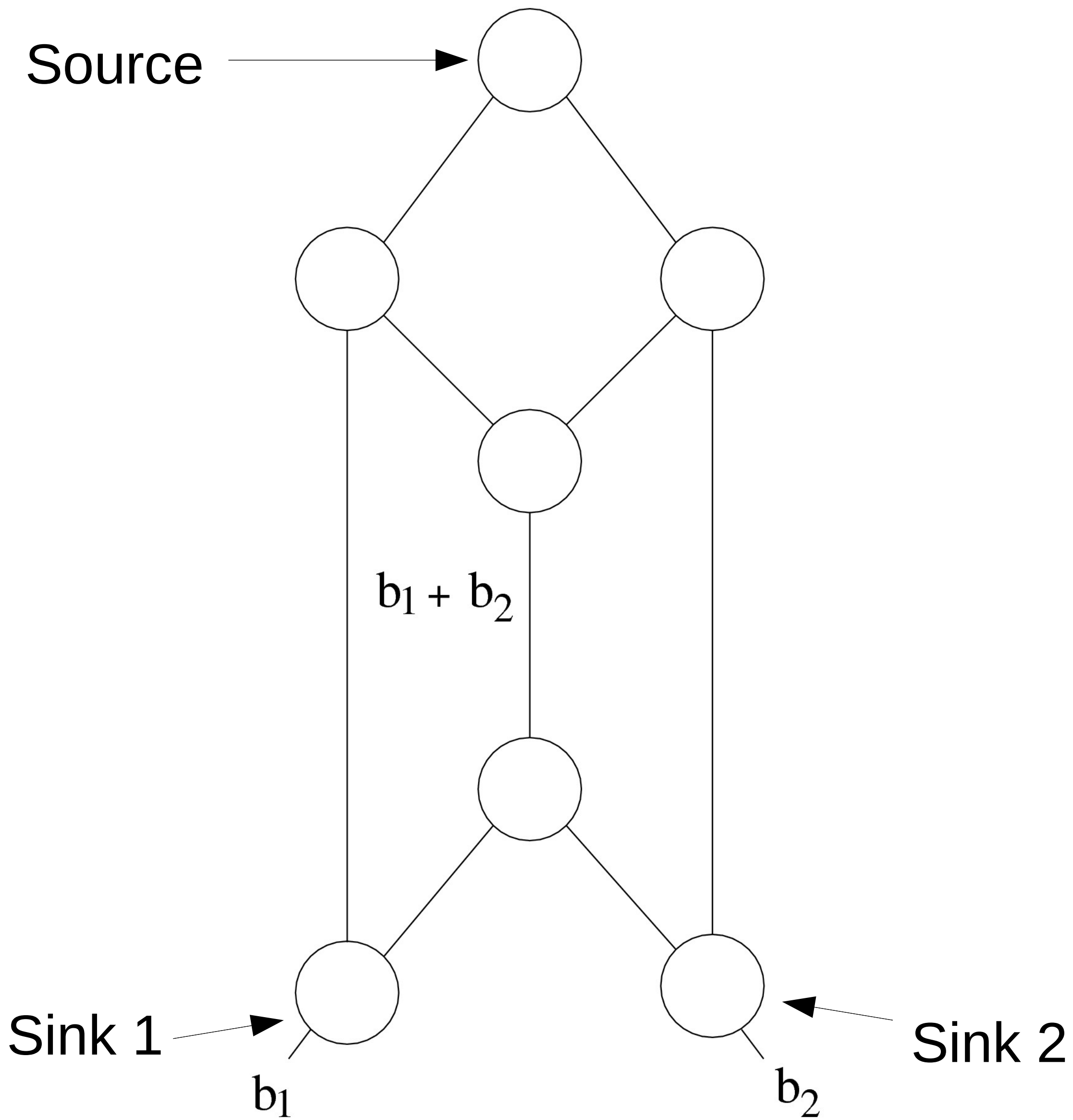
Network coding is an alternative to routing, in which routers forward linear combinations of bits, rather than routing bits separately.



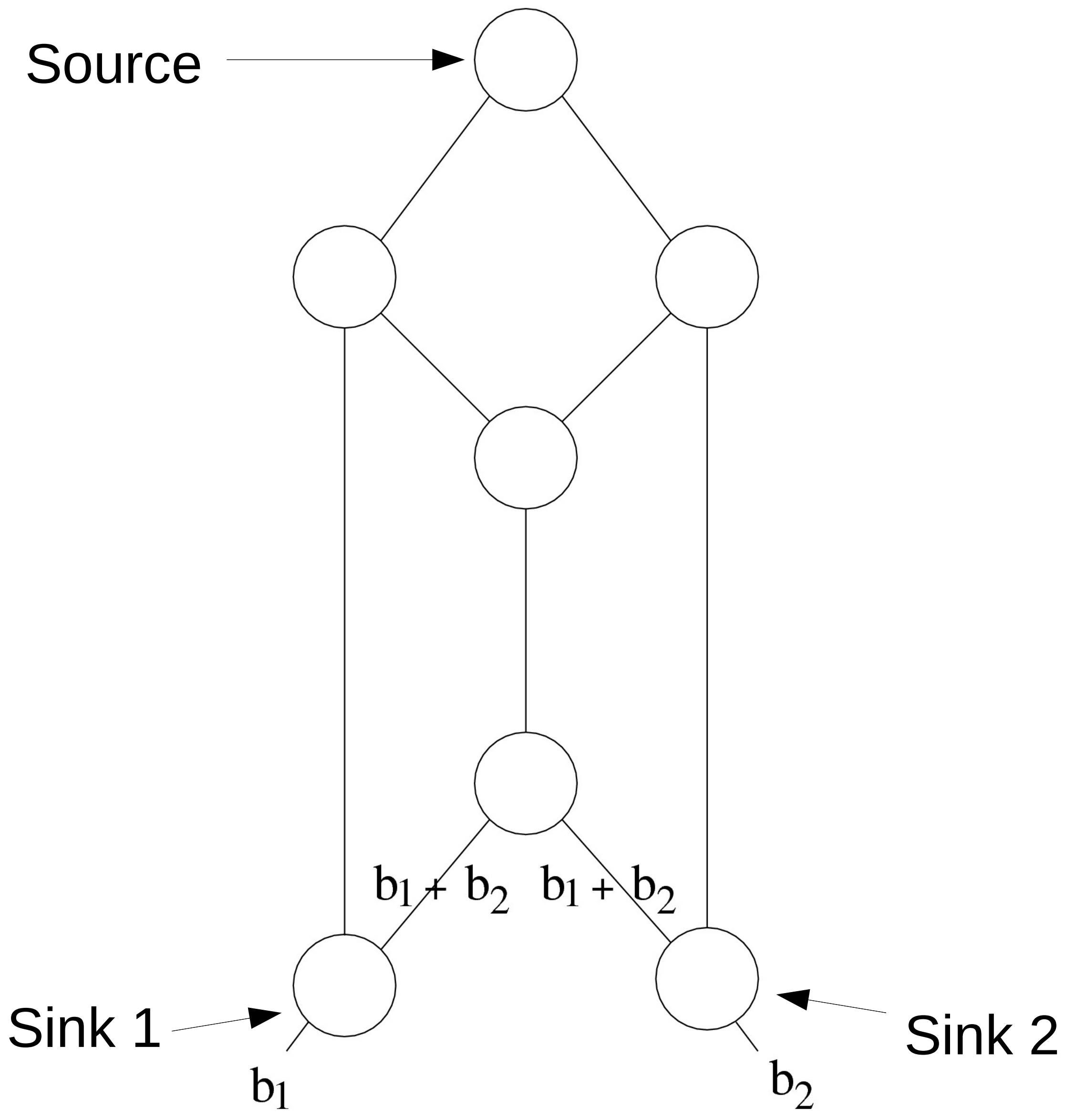
Network coding is an alternative to routing, in which routers forward linear combinations of bits, rather than routing bits separately.



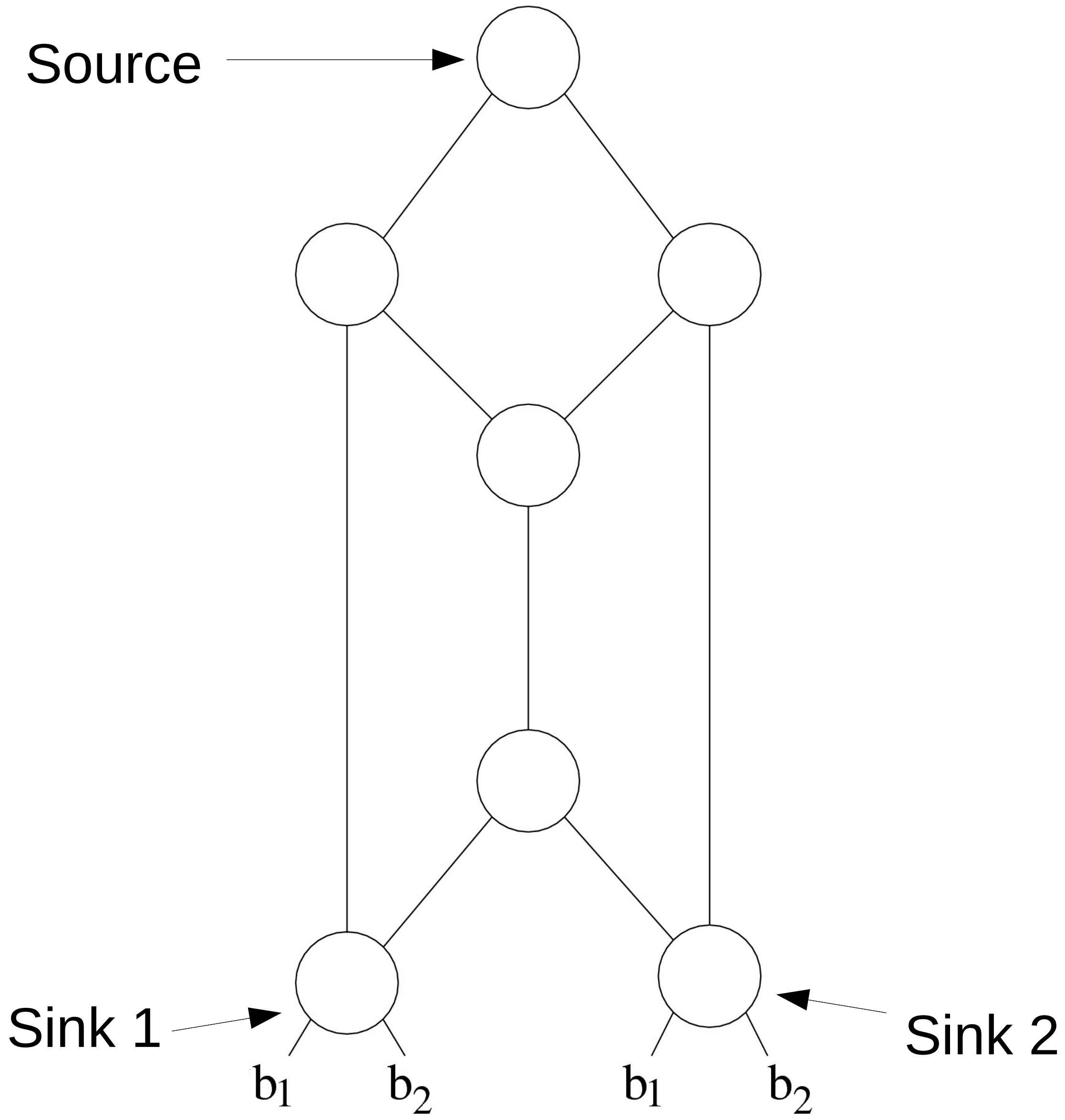
Network coding is an alternative to routing, in which routers forward linear combinations of bits, rather than routing bits separately.



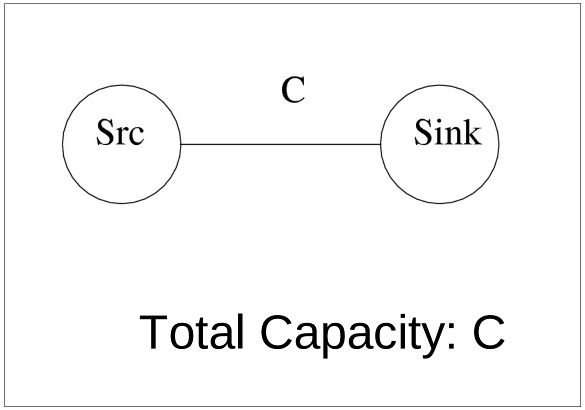
Network coding is an alternative to routing, in which routers forward linear combinations of bits, rather than routing bits separately.

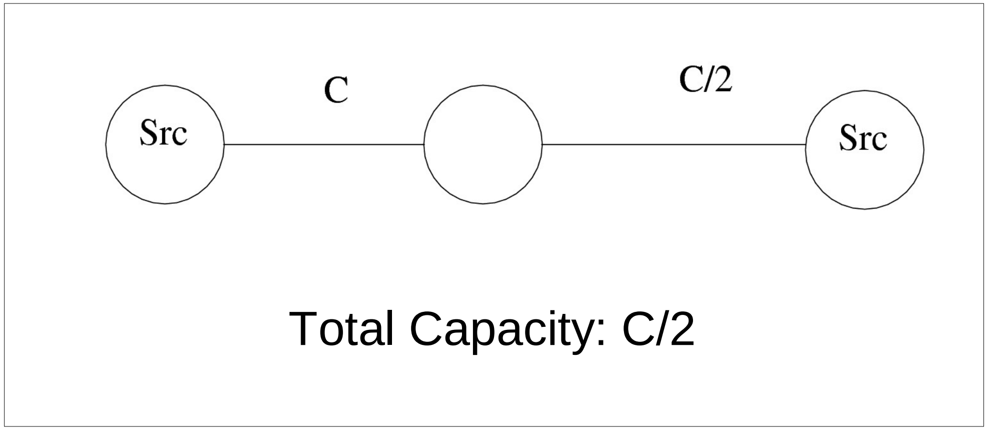


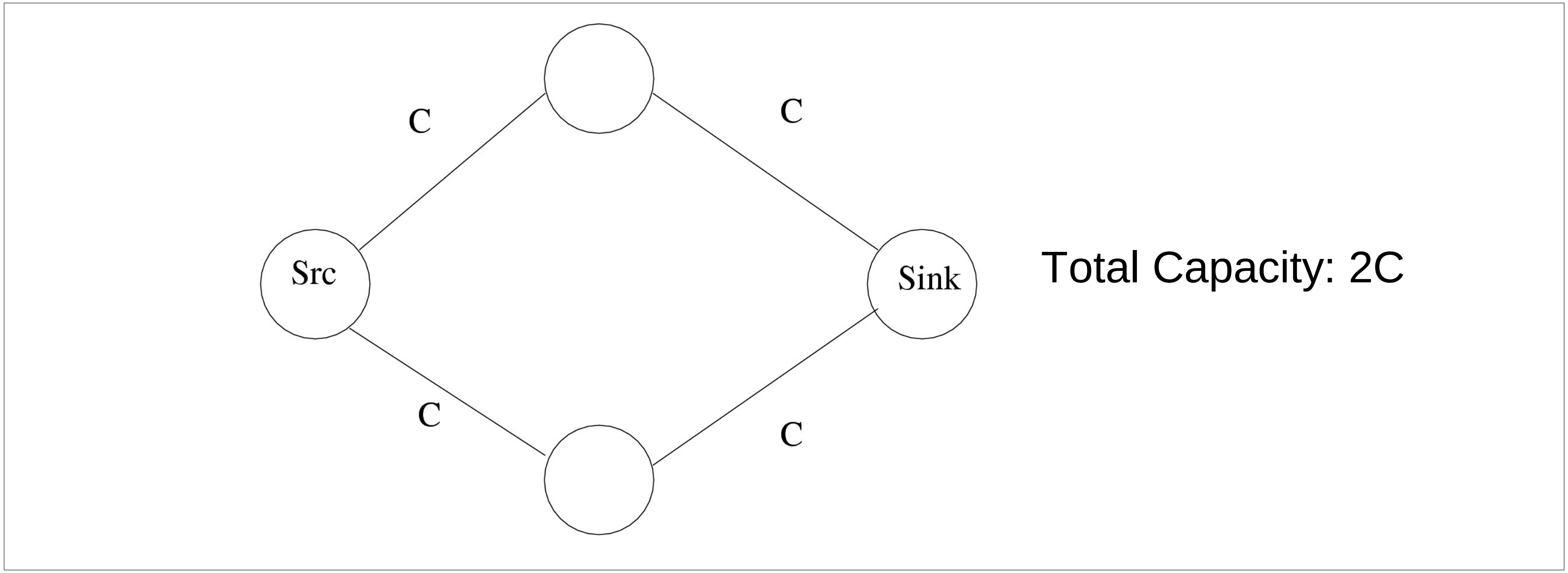
Network coding is an alternative to routing, in which routers forward linear combinations of bits, rather than routing bits separately.



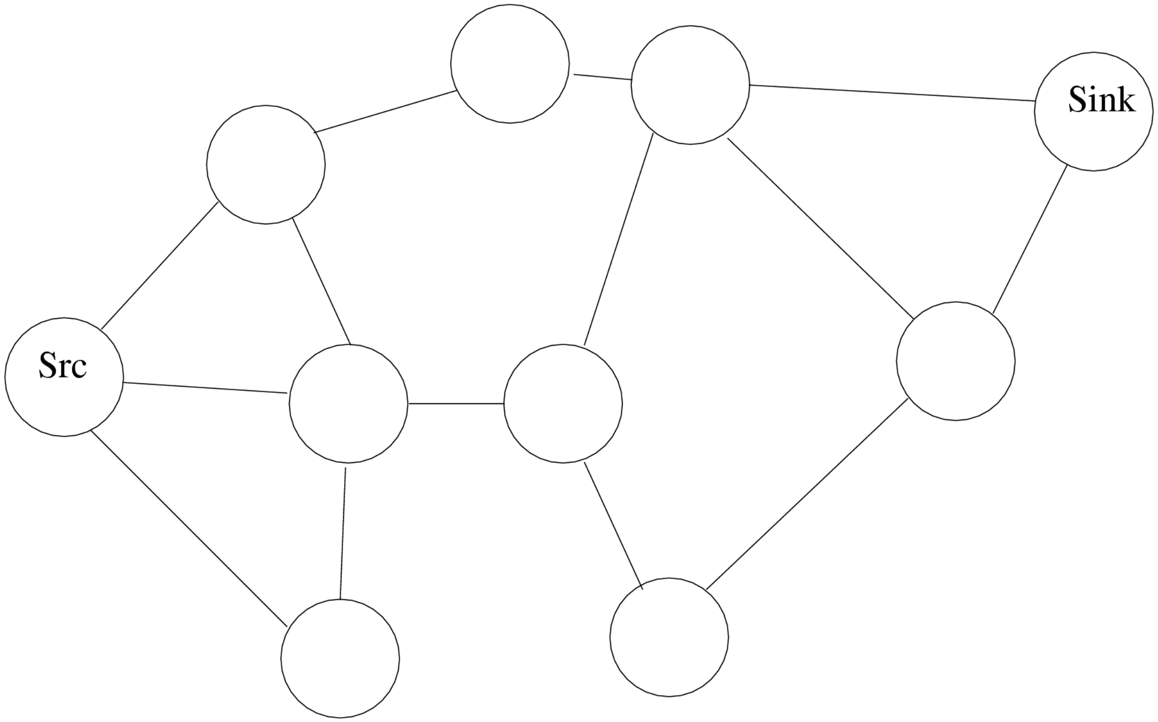
Unicast capacity is dependent on the network topology







Unicast capacity is dependent on the network topology



How to find unicast capacity in the general case?

The answer is provided By the *max-flow min-cut theorem [1].*

The max-flow min-cut theorem is a formal description of the maximum unicast capacity of any network.

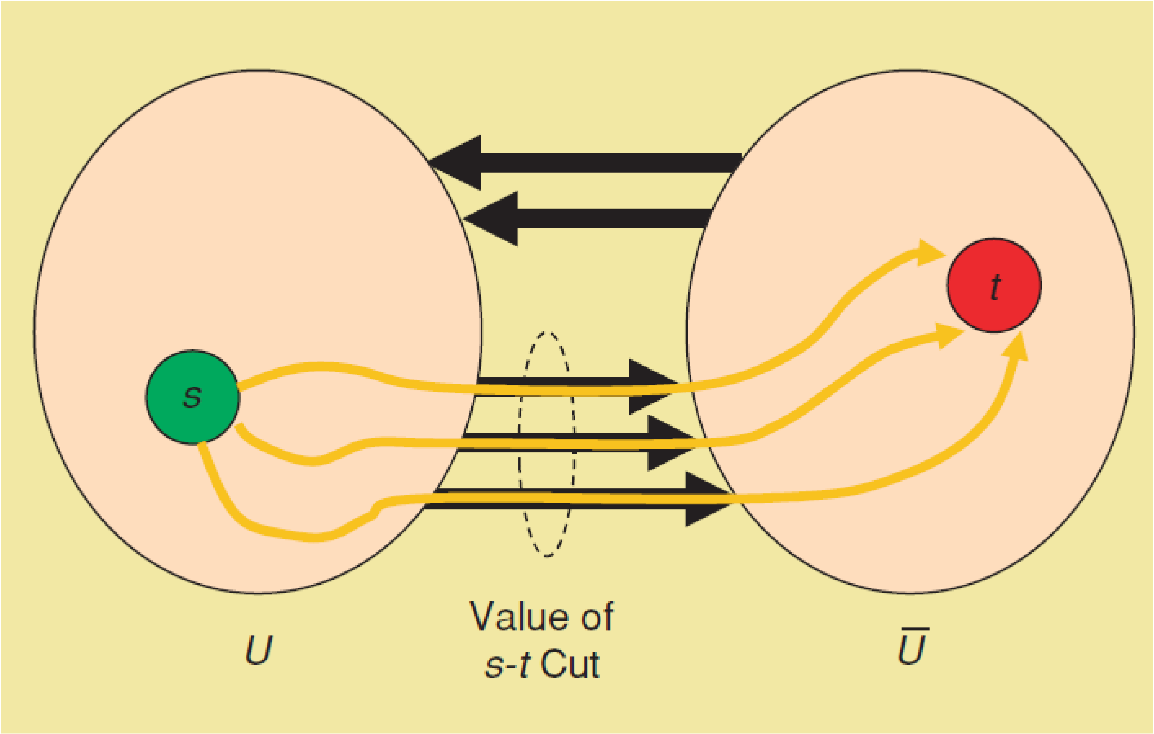


Image courtesy of Chou et. al. [1]

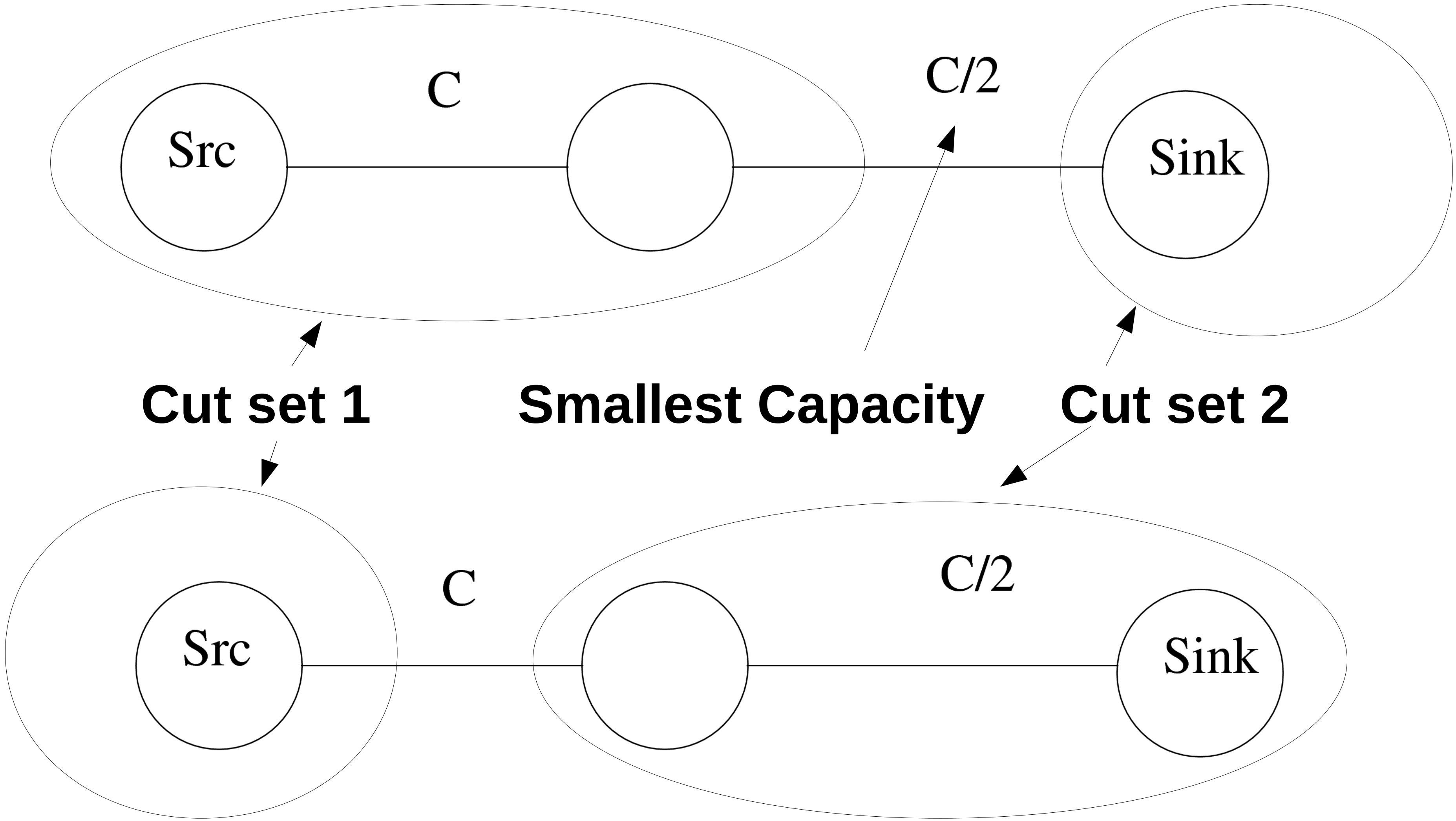
- **Consider** a network containing a source *s* and a sink *t*

- **Partition** the network nodes into two disjoint sets, one containing the source, one containing the sink

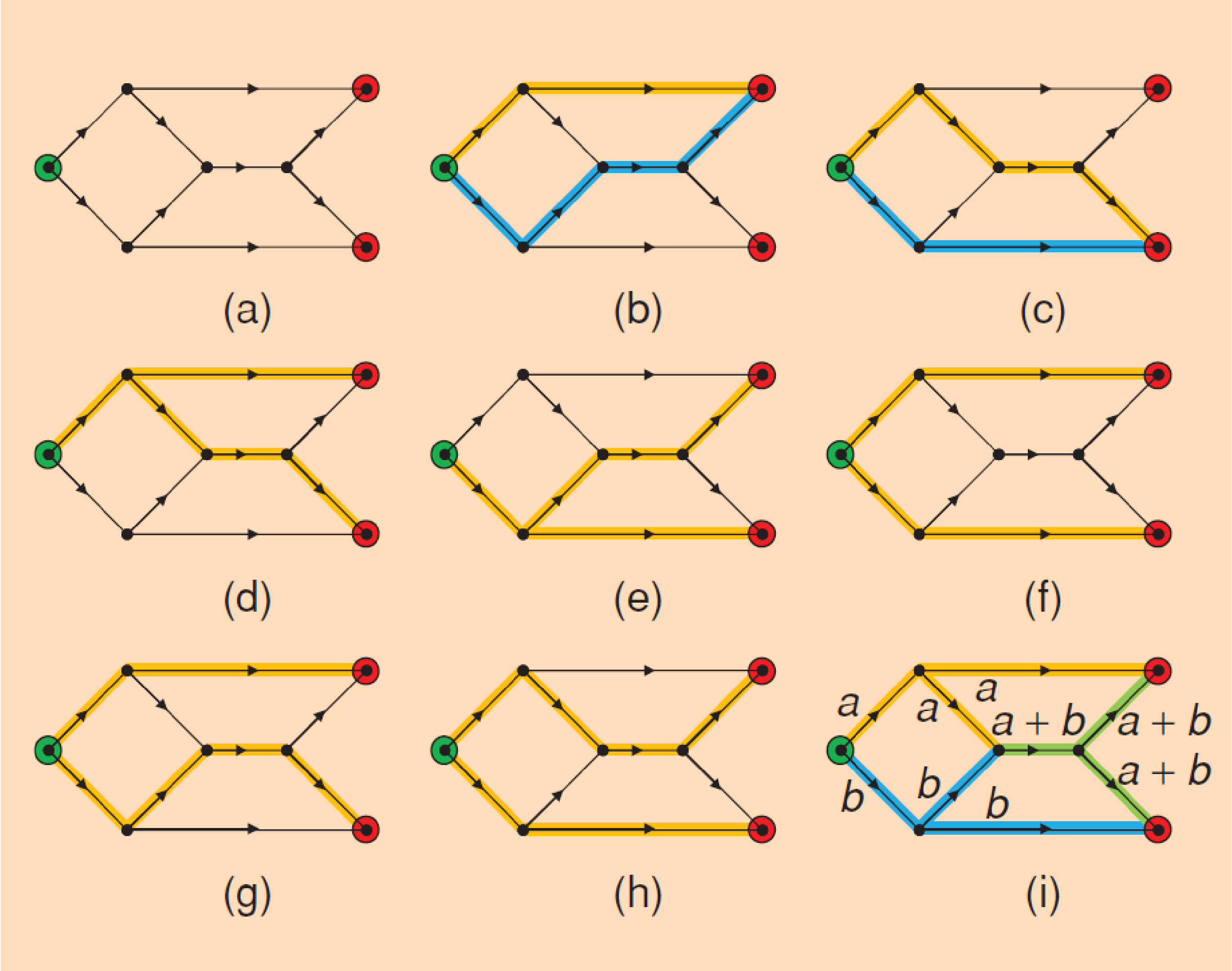
-**The max-flow min-cut theorem** states that the maximum unicast capacity between source and sink is the minimum capacity between disjoint sets.

- **These sets** are referred to as *cuts*

Simple example of max-flow min cut in the linear network



Network coding is necessary to achieve the *multicast* capacity of a network.



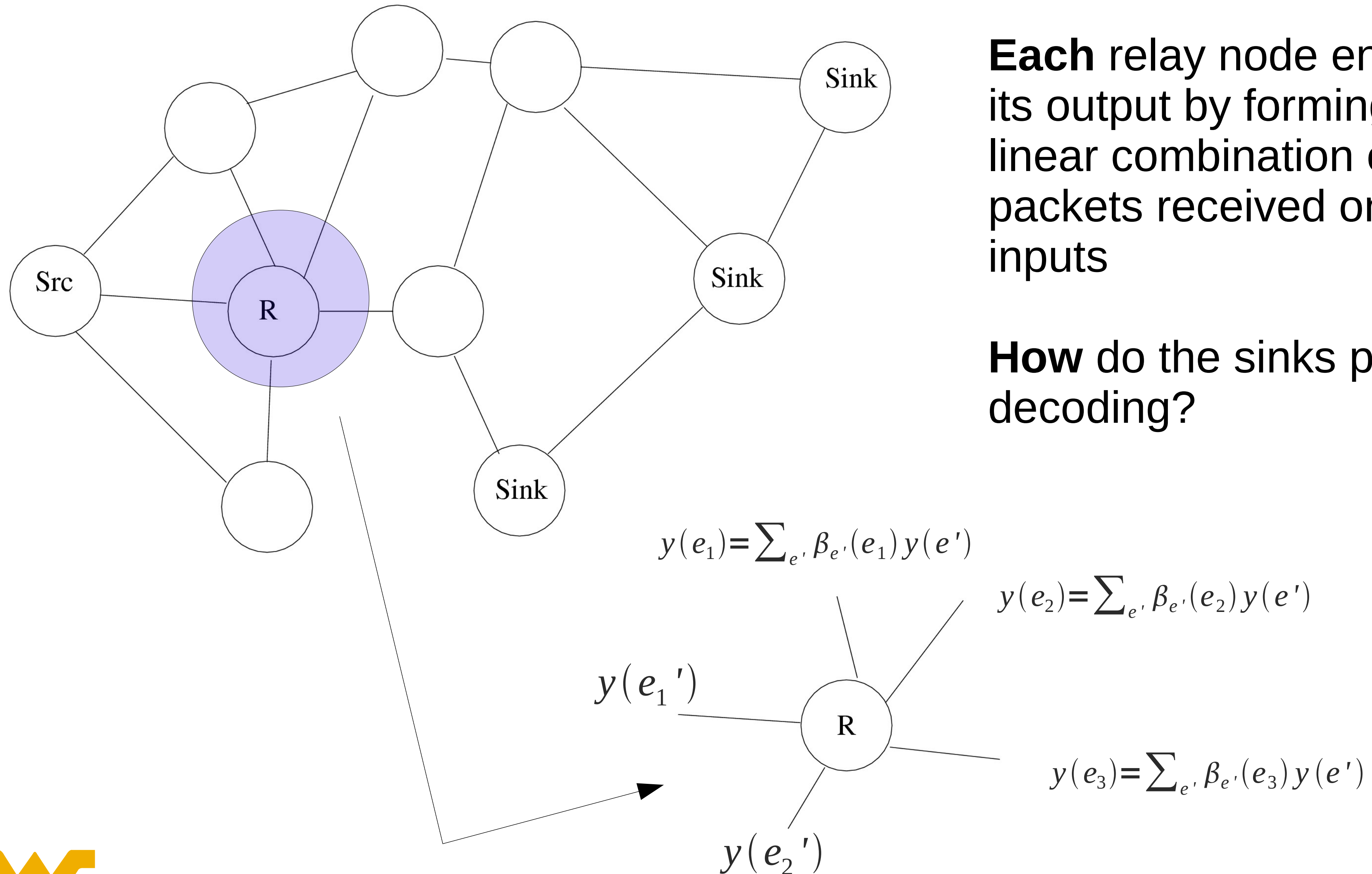
Unicast capacities = 2 bps

Multicast capacities w/ routing = 1 bps

Multicast capacities w/ network coding = 2 bps

Image courtesy of Chou et. al. [1]

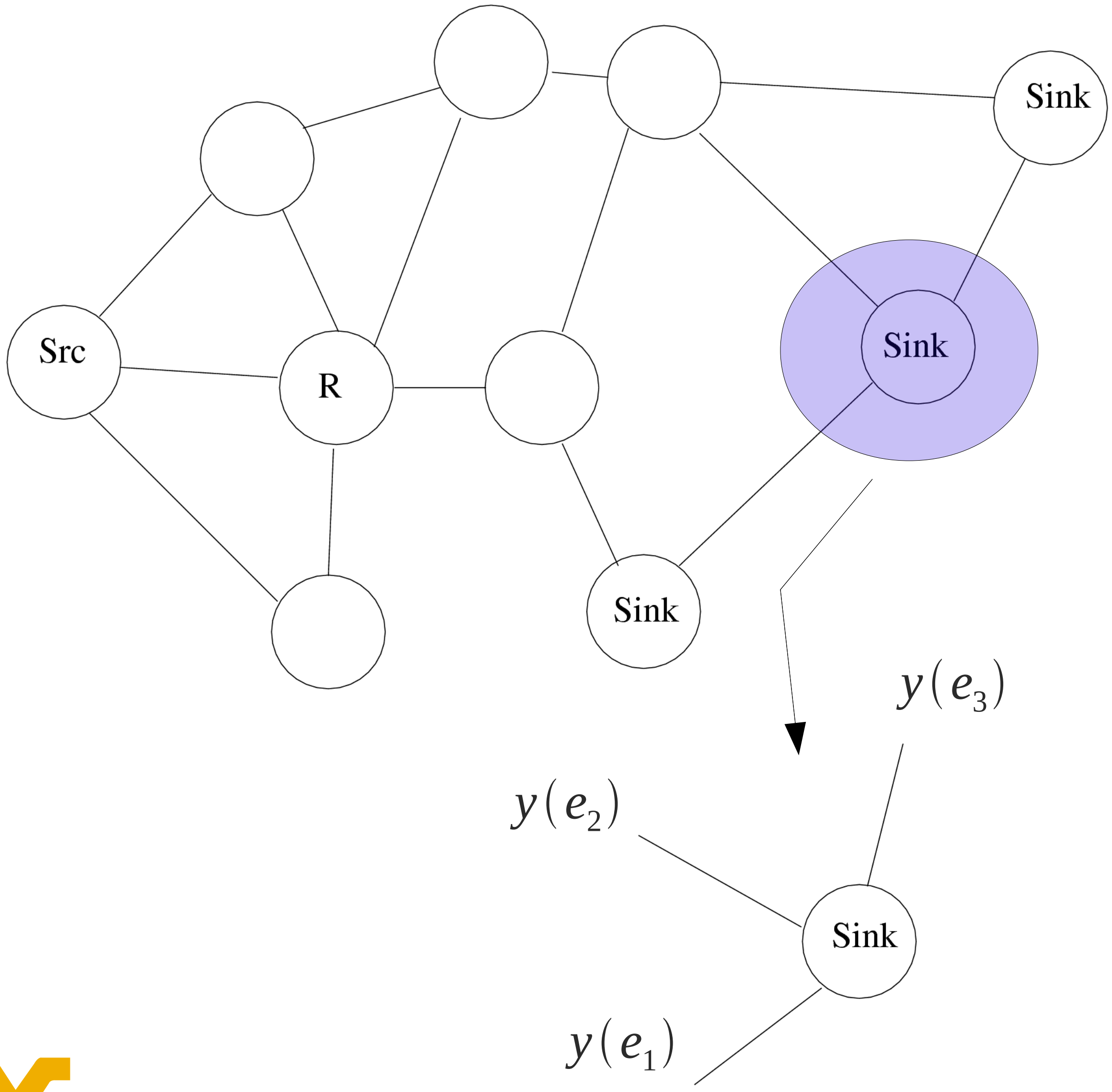
How do we perform network coding in a general topology?



Each relay node encodes its output by forming a linear combination of the packets received on all inputs

How do the sinks perform decoding?

How do we perform network coding in a general topology?

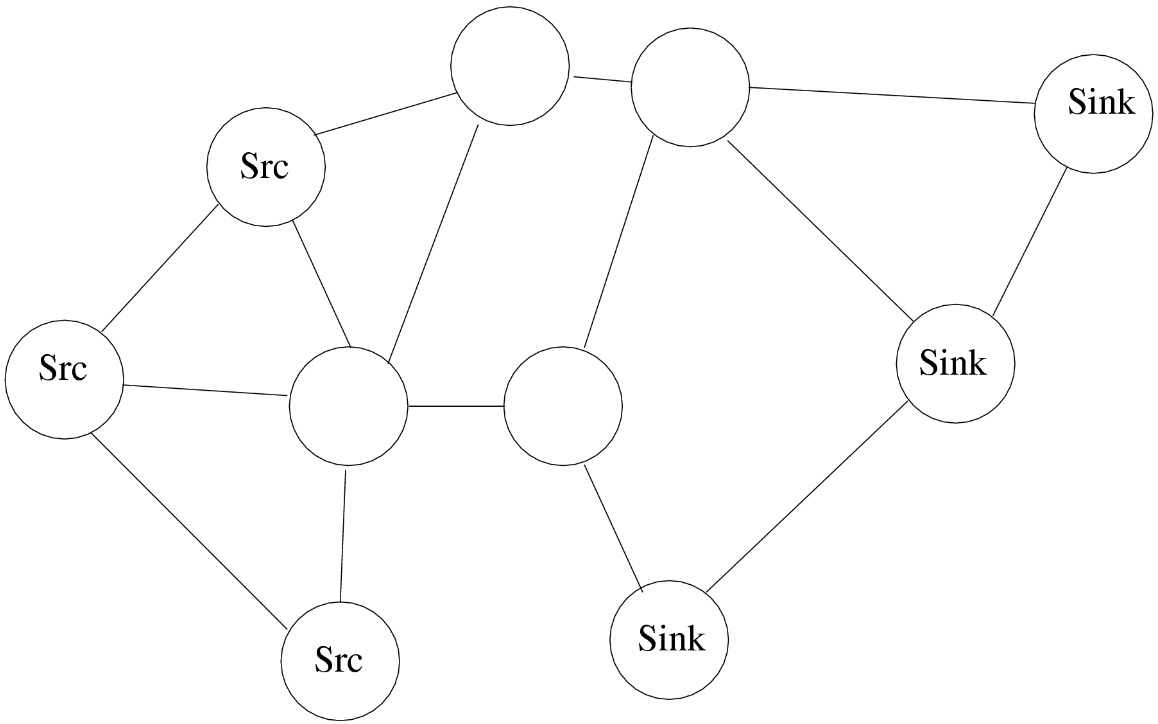


The sinks decode the source packets (x's) by solving a linear system in the received coded packets (y's).

$$\begin{bmatrix} y(e_1) \\ y(e_2) \\ y(e_3) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & g_2(e_1) & g_3(e_1) \\ g_1(e_2) & g_2(e_2) & g_3(e_2) \\ g_1(e_3) & g_2(e_3) & g_3(e_3) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= G_t \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Medard, Ho et. al. [4] present an algorithm for performing random, linear NC in the general multi-source, multi-sink case.



- This technique is at the heart of Microsoft's plan to de-throne BitTorrent.

- It is shown that as code length increases, the probability of achieving capacity w/ random coefficients $\rightarrow$ 1.

- Random network coding is demonstrated as a more efficient algorithm than naïve flooding, when a source wishes to communicate with a sink in a dense network with no explicit routing information.

How can we guarantee invertibility of the global encoding matrix at each sink?

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \dots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \dots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}$$

The encoding vectors used at each node must be chosen so that the global encoding coefficients (g's) at the sinks form an **invertible matrix.**

It has been proven that if the size of the finite field from which the encoding vector coefficients is large compared to the number of edges In the network, the matrix of g's may be made invertible with high probability if the **encoding vectors at the relays are chosen randomly.**

Example:

$$|E| = 2^8 \qquad |F| = 2^{16} \qquad 1 - |E|/|F| \approx 0.996$$

Number of edges | Size of finite field | Probability of invertible G

Out-of-order arrival, buffering, and disseminating the encoding coefficients are practical Implementation issues in network coding.

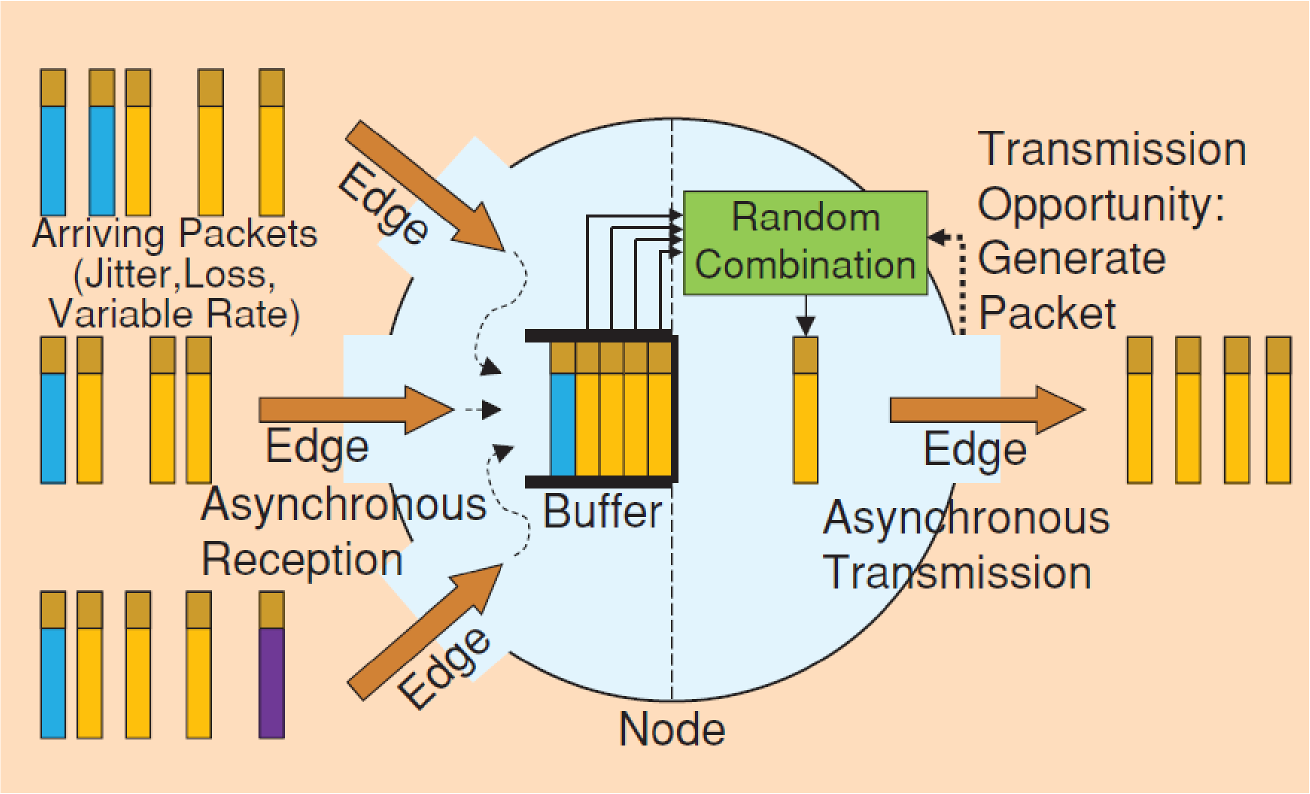


Image courtesy of Chou et. al. [1]

Since the encoding coefficients are generated at random at the relays, they are **concatenated with the data packet**, leading to an average **overhead** ~ 3-5%

Asynchronous arrival of packets necessitates buffering at the sink nodes.

Decoding proceeds when enough packets are received to invert the coefficient matrix.

Avalanche [5] is research project at Microsoft to create a peer-to-peer file-sharing service, similar to BitTorrent, which utilizes network coding.

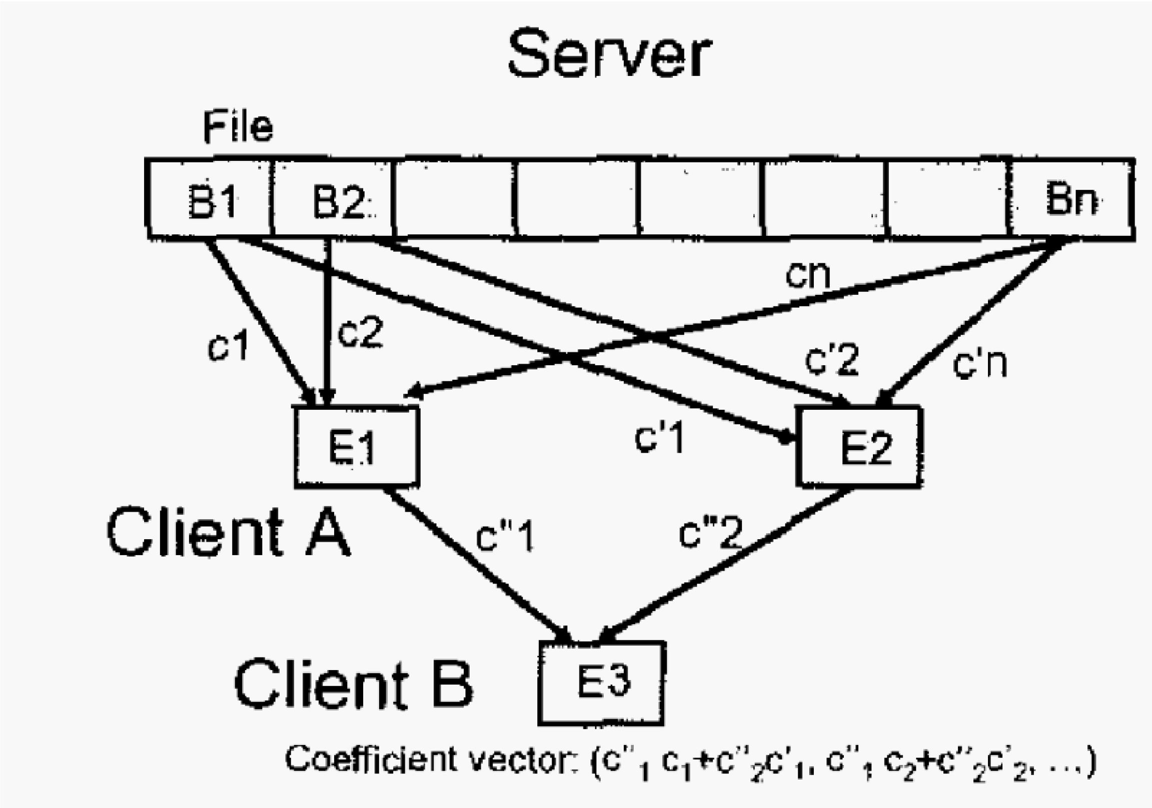


Image courtesy of Gkantsidis and Rodriguez [4]

- File servers partition files into **blocks**.

- Combinations of blocks are encoded with **random coefficients** and distributed to clients.

- Clients share blocks amongst themselves until enough are received such that each client can recover the file by **inverting a matrix of encoding coefficients.**

- Simulation results show throughput improvements of **2-3x** over a system which uses no coding.

Simulation results for a network of approximately 100 nodes representing the SprintLink ISP network.

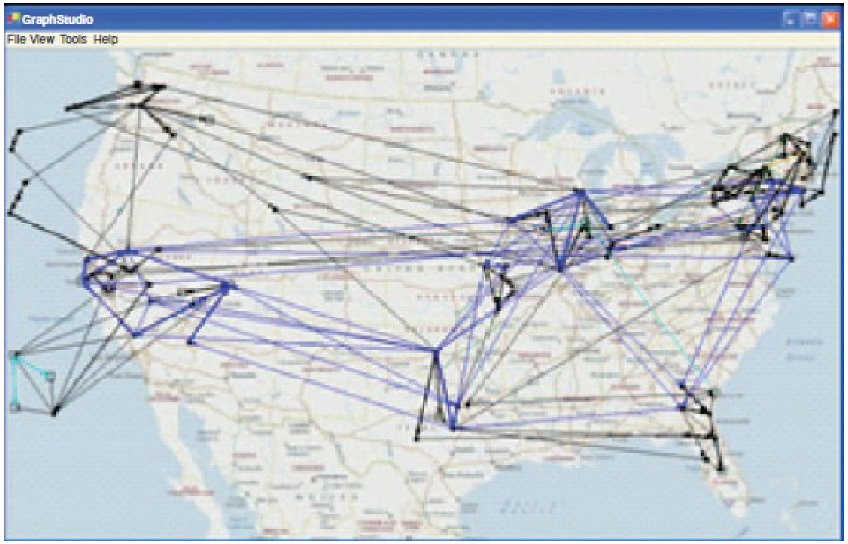


Image courtesy of Chou et. al. [1]

Simulation parameters:

-89 *nodes*
-972 *edges*
-one *source*, 20 *sinks*
-sender throughput:
450 Mbps-833 Mbps

Multicast throughput very close to capacity is attained.

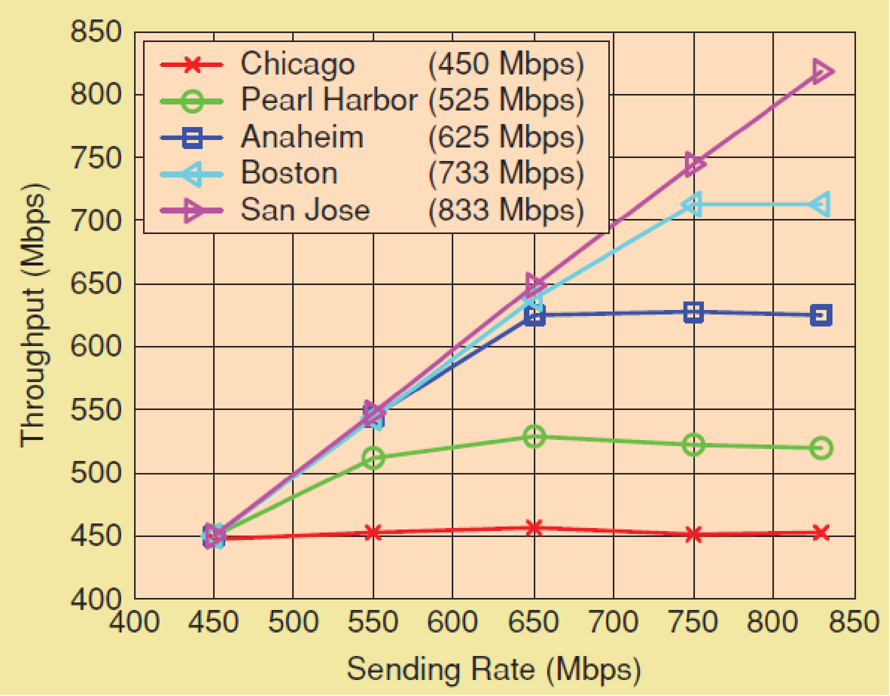
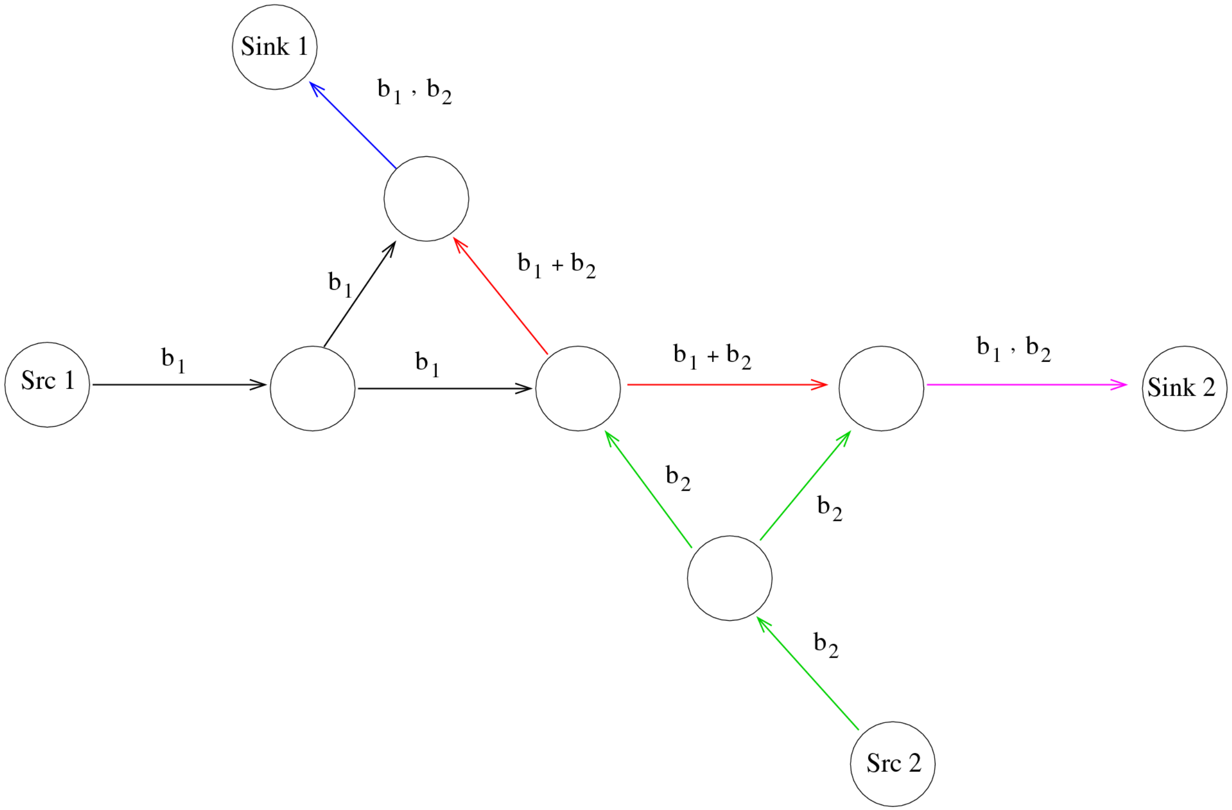


Image courtesy of Chou et. al. [1]

Wireless mesh networks provide unique opportunities for network coding both multicast and unicast flows, due to the broadcast nature of the medium.



Wireless mesh network utilizing opportunistic listening

Densely packed mesh networks provide opportunities for nodes to **overhear** transmissions from adjacent flows.

A particular network coding protocol for mesh networks known as **COPE** [3] implements this concept in the Linux kernel.

Experimental results show throughput increases of 5-300%, depending on **traffic patterns, network topology**, and **network layer protocol**.

Conclusion: Network coding is an effective means to achieve single-source multicast capacity in wired networks, and provides unique opportunities in wireless networks.

-The maximum achievable unicast capacity is described by the *max-flow min-cut theorem.*

- Network coding is necessary to achieve multicast capacity, with linear network coding sufficient to achieve capacity.

- Random generation of relay encoding coefficients eliminates the need for knowledge of global network topology at decoding nodes.

- Simulation and experimental results show throughput improvements ranging from 5-300% in practical topologies.

NEXT TIME

- Physical-layer network coding

- Simulation techniques for network coding

Conclusion: Network coding is an effective means to achieve single-source multicast capacity in wired networks, and provides unique opportunities in wireless networks.

References

[1] P. Chou and Y. Wu, "Network coding for the internet and wireless networks," *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 77–85, 2007.

[2] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, July 2000.

[3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Trans. Networking*, pp. 497–510, June 2008.

[4] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, 2006.

[5] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, 2005, pp. 2235–2245 vol. 4.