Peer-to-Peer Networks

11 Network Coding

Arne Vater

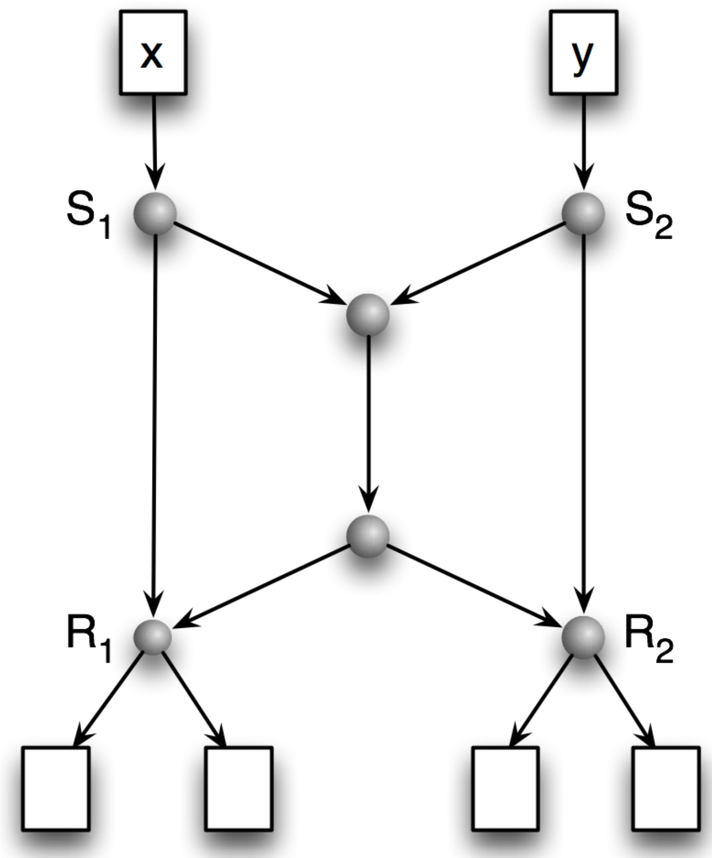
Technical Faculty

Computer Networks and Telematics

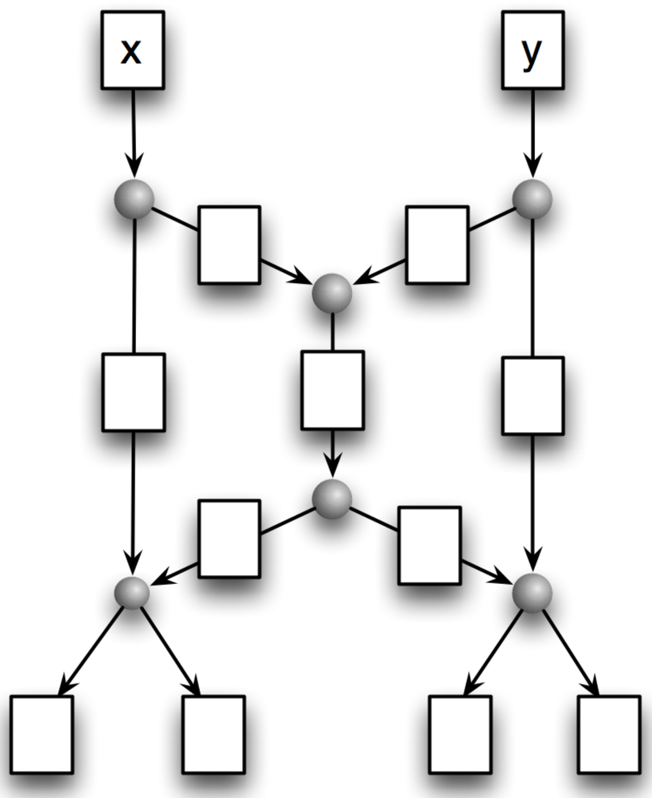
University of Freiburg

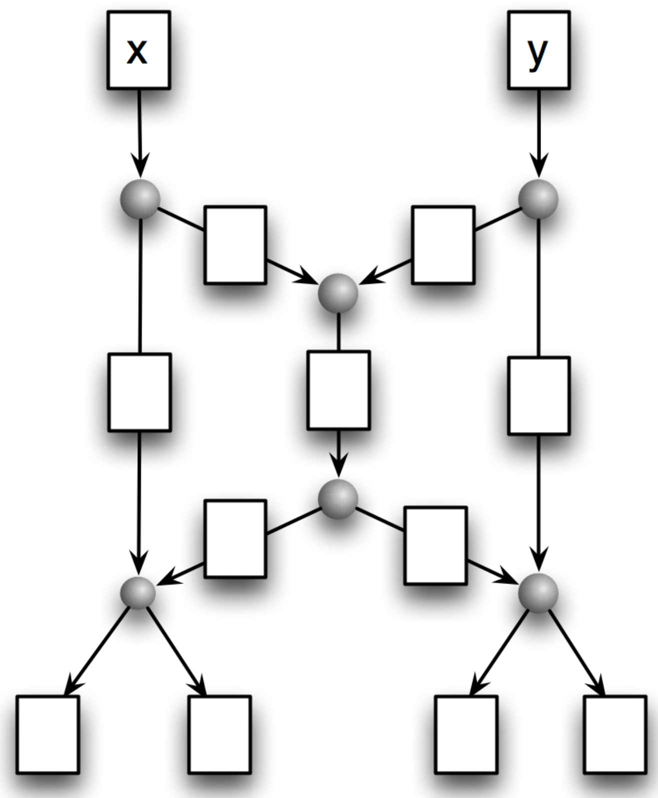
Network Flow

- How can network flow be optimized?
  - two data bits
    - x, y
  - two sender
    - S_1, S_2
  - two receiver
    - R_1, R_2
  - link capacity 1
  - deliver both bits to both receiver



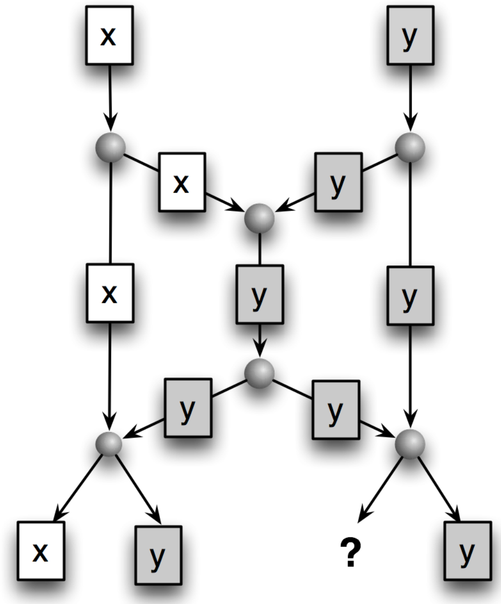
Network Flow





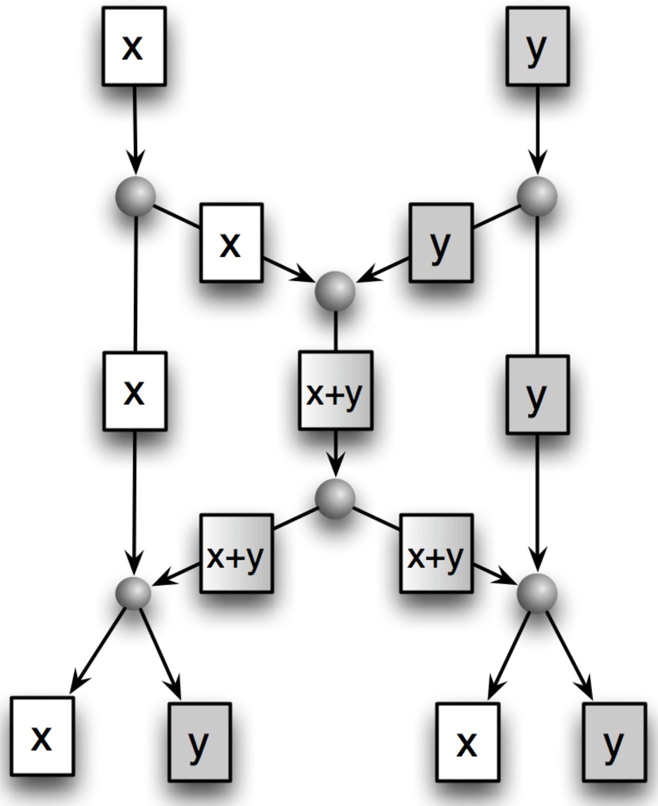
Network Flow

- Simple transmission of bits allows maximal flow 3
  - minimal cut = 3
  - middle edge is bottleneck
- Can we do better?



Network Coding

- R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung
  - *Network Information Flow*, (IEEE Transactions on Information Theory, IT-46, pp. 1204-1216, 2000)
- Solution
  - Send Xor of x and y on the middle edge



Network Coding

- Theorem [Ahlswede et al.]
  - For each graph there exists a network code such that each sink can receive as many information as allowed by the maximum flow to that sink.

Linear Network Codes

- Koetter, Médard
  - Beyond Routing: An Algebraic Approach to Network Coding
- Goal
  - finding those codes for network coding
- Solution
  - linear combinations are sufficient for any network coding
    - even random linear combinations in Practical Network Coding for peer-to-peer networks

Application Areas

- Sattelite communication
  - preliminary work
- WLAN
  - Xor in the Air, COPE
    - simple network code improves network flow
- Ad hoc networks
- Sensor networks
- Peer-to-peer networks

Coding and Decoding

- Original message: $x_1, x_2, ..., x_n$
- Code packets: $b_1, b_2, ..., b_n$
- Random linear coefficient c_{ij}

$$(c_{i1}, \ldots, c_{in}) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = b_i$$

- Thus

$$\begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

If matrix (c_{ij}) is invertible then:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix}^{-1} \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

Inverse of Random Matrix

- Theorem
  - If the values of an n n matrix are randomly chosen from a finite field with s elements, then the matrix is invertible with probability at least

$$1 - \sum_{i=1}^{n} \frac{1}{s^i}$$

- Problem
  - Numbers become larger with each calculation

Galois Fields

- Idea: Use Galois field $GF[2^w]$
  - efficient computation
  - power of two suits binary data representation
- $GF[2^w]$ = finite field with 2^w elements
  - elements are binary strings with length w
  - $0 = 0^w$ identity element for addition
  - $1 = 0^{w-1}1$ identity element for multiplication
- u + v = bit-wise Xor
  - i.e. 0101 + 1100 = 1001
- $a \cdot b$ = polynom product modulo an irreducible polynom and modulo 2
  - i.e.

$$(a_{w-1}\ldots a_1 a_0)(b_{w-1}\ldots b_1 b_0) =$$
$$\left(\left(a_0 + a_1 x + \ldots + a_{w-1}x^{w-1}\right)\left(b_0 + b_1 x + \ldots + b_{w-1}x^{w-1}\right) \bmod q(x)\right) \bmod 2$$

Example: GF[2^2]

$q(x) = x^2 + x + 1$

generating element of GF[4]	polynomial in GF[4]	binary representation in GF[4]	decimal representation
0	**0**	**00**	**0**
x^0	**1**	**01**	**1**
x^1	**x**	**10**	**2**
x^2	**x + 1**	**11**	**3**

$x^2 = x + 1$?

- Why is $x^2 = x + 1$?
  - $q(x) = x^2 + x + 1$

$$x^2 \bmod x^2 + x + 1 =$$

Example: $GF[2^2]$

+	0 = 00	1 = 01	2 = 10	3 = 11
0 = 00	00	01	10	11
1 = 01	01	00	11	10
2 = 10	10	11	00	01
3 = 11	11	10	01	00

bit-wise Xor

Example: GF[2^2]

*	0 = 0	1 = 1	2 = x	3 = x^2
0 = 0	0	0	0	0
1 = 1	0	1	x	x^2
2 = x	0	x	x^2	1
3 = x^2	0	x^2	1	x

$x^3 = 1?$

- Why is $x^3 = 1$?
  - $x^2 = x + 1$
  - $x + x = 0$

$$x^3 =$$

Irreducible Polynoms

- Irreducible polynoms are non-decomposable
  - w = 2: $x^2 + x + 1$
  - w = 4: $x^4 + x + 1$
  - w = 8: $x^8 + x^4 + x^3 + x^2 + 1$
  - w = 16: $x^{16} + x^{12} + x^3 + x + 1$
  - w = 32: $x^{32} + x^{22} + x^2 + x + 1$
  - w = 64: $x^{64} + x^4 + x^3 + x + 1$
- Decomposable polynom: $x^2 + 1 = (x + 1)^2 \bmod 2$

Fast Multiplication

- Power laws
  - $\{2^0, 2^1, 2^2, ...\}$
  - $= \{x^0, x^1, x^2, x^3, ...\}$
  - = exp(0), exp(1), exp(2), ...
- $\exp(x + y) = \exp(x) \cdot \exp(y)$
- Inverse function: $\log(\exp(x)) = x$
  - $\log(x \cdot y) = \log(x) + \log(y)$
- $x \cdot y = \exp(\log(x) + \log(y))$
  - Attention: normal addition in the exponent
- Values for exponential and logarithmic function stored in lookup tables

Example: GF[16]

$q(x) = x^4 + x + 1$

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
exp(x)	1	x	x^2	x^3	$1+x$	$x+x^2$	x^2+x^3	$1+x+x^3$	$1+x^2$	$x+x^3$	$1+x+x^2$	$x+x^2+x^3$	$1+x+x^2+x^3$	$1+x^2+x^3$	$1+x^3$	1
exp(x)	1	2	4	8	3	6	12	11	5	10	7	14	15	13	9	1

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
log(x)	0	1	4	2	8	5	10	3	14	9	7	6	13	11	12

$x \cdot y = \exp(\log(x) + \log(y))$

Special Case: GF[2]

- Boolean algebra
  - $x + y \quad = x \text{ XOR } y$
  - $x \cdot y \quad = x \text{ AND } y$

Peer-to-Peer Networks

11 Network Coding

Arne Vater

Technical Faculty

Computer Networks and Telematics

University of Freiburg