ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Algorithms for Radio Networks

Network Coding

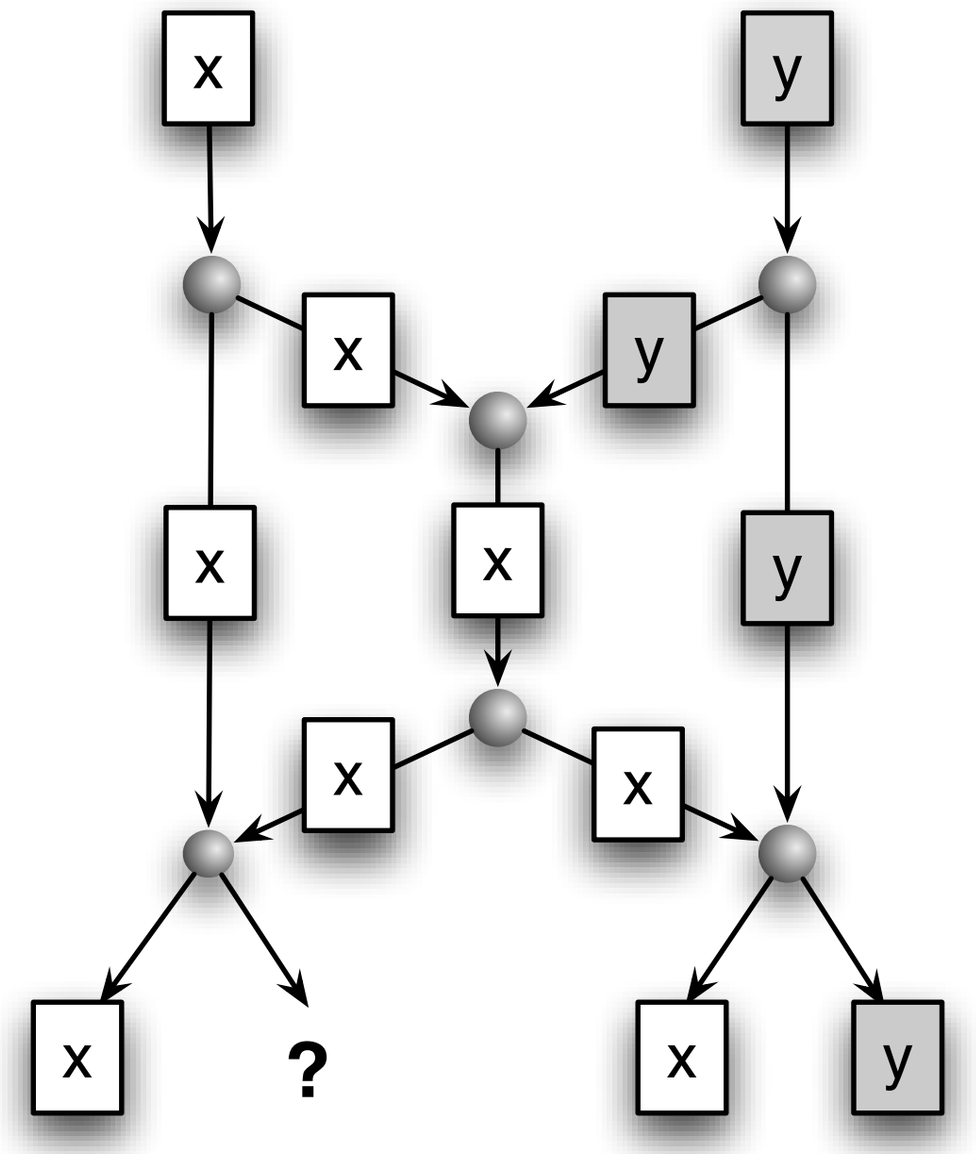
University of Freiburg
Technical Faculty
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

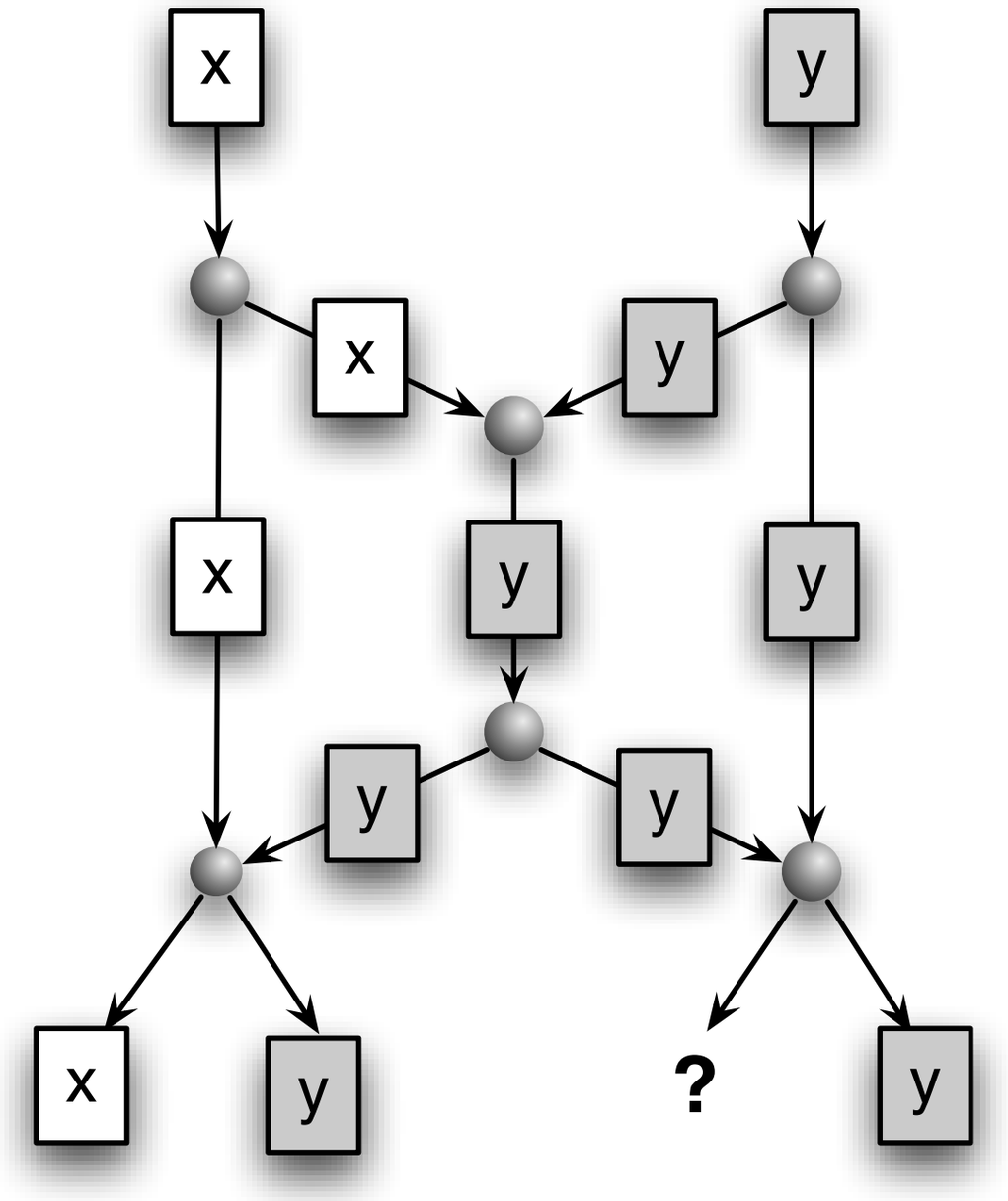
Network Coding

- **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)
- **Example**
  - Bits x and y are to be transfered
  - Each edge carries only a bit
  - If bits are transfered as is
    - then both x and y cannot be received either on the left or right side



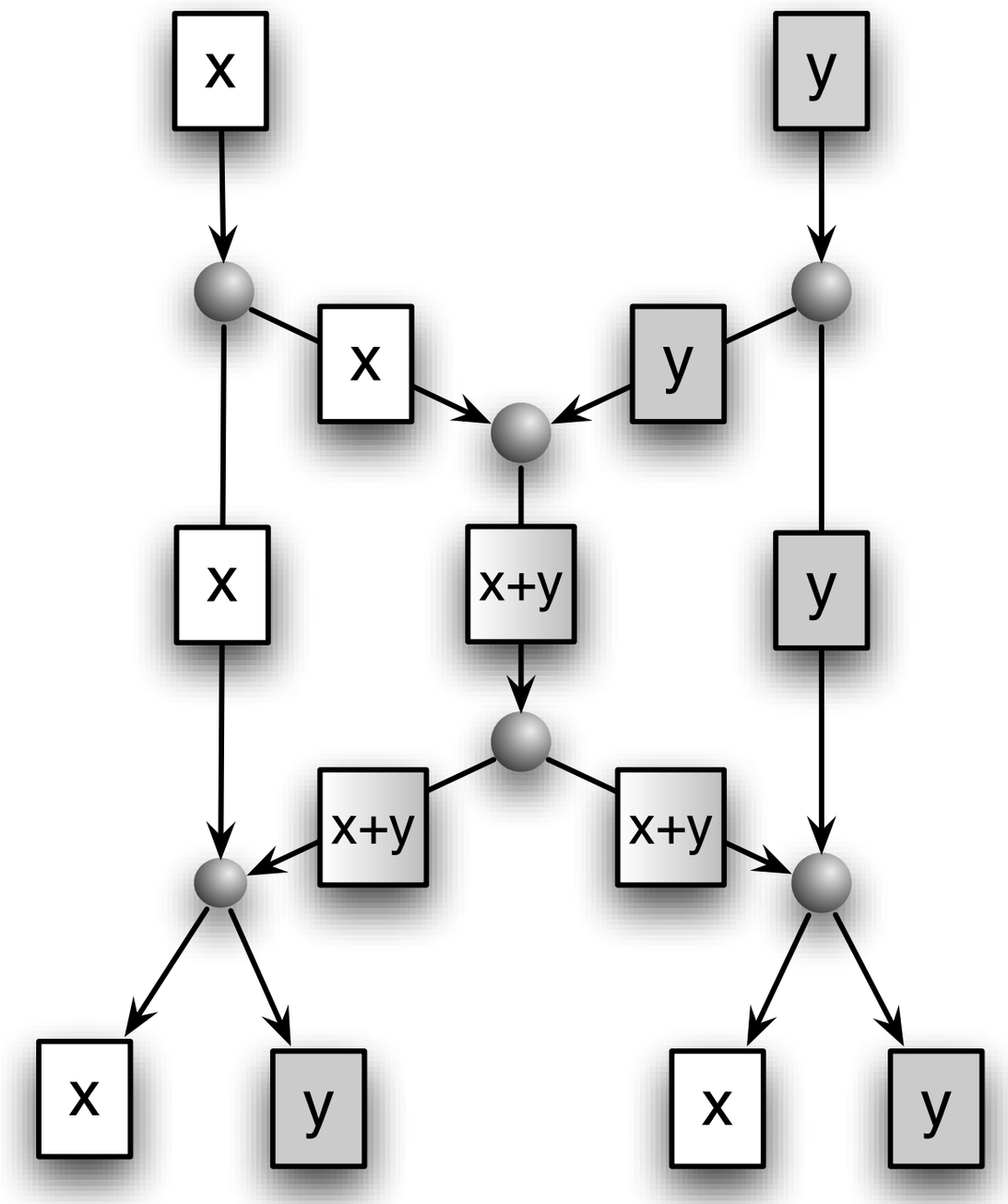
Network Coding

- **Example**
  - Bits x and y are to be transfered
  - Each edge carries only a bit
  - If bits are transfered as is
    - then both x and y cannot be received either on the left or right side



Network Coding

- **Solution**
  - Transfer Xor A+B on the middle edge



Network Coding and Flow

- **Theorem [Ahlswede et al.]**
  - For each graph there exists a network code such that each sink can receive as many bits as the maximum flow allows for each sink.

Linear Codes for Network Coding

- **Koetter, Médard**
  - Beyond Routing: An Algebraic Approach to Network Coding
- **Task**
  - Efficiently compute the network code
- **Solution**
  - Linear codes can always solve network coding
- **Practical Network Coding**
  - With high probability even random linear combinations suffice

Application Areas

- **Satellite Communication**
  - Preliminary work was published there
- **Peer-to-Peer networks**
  - Better information flow better than previous protocols
  - But too inefficient to displace prevalent protocols, e.g. Bittorrent
- **WLAN**
  - Xor in the Air, COPE
    - Simple network code improves flow
- **Ad-Hoc Networks, Wireless Sensor Networks, ...**

Coding and Decoding

- **Original message: $x_1, x_2, \ldots, x_m$**
- **Coding packet: $y_1, y_2, \ldots, y_m$**
- **Random variable r_{ij}**
- **Then:**

$$(r_{i1} r_{i2} \ldots r_{im}) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = y_i$$

$$\begin{pmatrix} r_{11} & \ldots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \ldots & r_{mm} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

- **If the matrix (r_{ij}) is invertable**

$$\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} r_{11} & \ldots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \ldots & r_{mm} \end{pmatrix}^{-1} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

Inverse of a Random Matrix

- **Theorem**
  - If the numbers of an m x m Matrix are chosen randomly from a finite field with b elements, then the matrix is invertable with proability of at least

$$1 - \sum_{i=1}^{m} \frac{1}{b^i}$$

- **Idea: Consder Galois-Field GF[2^k]**
  - Computation is efficient
  - Binary represenation of data straight-forward

Galois Field

- **GF(2^w) = finite field with 2^w elements**
  - elements are binary strings of length w
  - $0 = 0^w$ neutral element of addition
  - $1 = 0^{w-1}1$ neutral element of multiplikation
- **u + v = bit-wise Xor of strings**
  - z.B. 0101 + 1100 = 1001
- **a b = product of polynomials modula a given irreducible polynomial and modulo 2**
  - i.e. $(a_{w-1} \ldots a_1\ a_0)\ (b_{w-1} \ldots b_1\ b_0) =$

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

Example: GF(2^2)

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

Generator 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

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

Example: GF(2^2)

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

*	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

Irreducible Polynomial

- **Irreducible polynomial cannot be factorized**
  - **Irreducible polynomial** $x^2+1 = (x+1)^2 \bmod 2$
- **Irreducible polynomials**
  - 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: $x^{64}+x^4+x^3+x+1$

Fast Multiplication

- **Power law**
  - Consider $\{2^0, 2^1, 2^2,...\}$
  - $= \{x^0, x^1, x^2, x^3, ...$
  - $= \exp(0), \exp(1), ...$
- **$\exp(x+y) = \exp(x) \exp(y)$**
- **Inverse function: $\log(\exp(x)) = x$**
  - $\log(x \cdot y) = \log(x) + \log(y)$
- **$x\ y = \exp(\log(x) + \log(y))$**
  - Caution: in the exponent standard addition
- **Tables store exponential function and logarithm**

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

- $5 \cdot 12 = \exp(\log(5)+\log(12)) = \exp(8+6) = \exp(14) = 9$
- $7 \cdot 9 = \exp(\log(7)+\log(9)) = \exp(10+14) = \exp(24) = \exp(24-15) = \exp(9) = 10$

Special Case GF[2]

- **Network Coding in GF[2]**
  - Boolean Algebra
    - x + y = x XOR y
    - x y = x AND y
- **Example**
  - Xor in the Air
  - Multicasting in Ad-Hoc Networks
- **Disadvantage**
  - Full potential of network coding is unused
- **Advantage**
  - Transparent, intuitiv and very efficient

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Algorithms for Radio Networks

Network Coding

University of Freiburg
Technical Faculty
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG