Iterative Coding for Network Coding

Andrea Montanari*, Vish Rathi† and Rüdiger Urbanke†

*Stanford, †EPFL

March 26, 2008
Outline

1. Information is not oranges
2. Practical network coding
3. Noisy network coding
4. Construction and decoding algorithm
5. Achieving capacity
Outline

1. Information is not oranges
2. Practical network coding
3. Noisy network coding
4. Construction and decoding algorithm
5. Achieving capacity
Outline

1. Information is not oranges
2. Practical network coding
3. Noisy network coding
4. Construction and decoding algorithm
5. Achieving capacity
Outline

1. Information is not oranges
2. Practical network coding
3. Noisy network coding
4. Construction and decoding algorithm
5. Achieving capacity
Outline

1. Information is not oranges
2. Practical network coding
3. Noisy network coding
4. Construction and decoding algorithm
5. Achieving capacity
Information is not oranges
Current approach to communication networks
Current approach to communication networks

![Diagram of a network with source and destination nodes connected by edges.](image-url)
Current approach to communication networks
Current approach to communication networks
What are we losing? The butterfly example

[Ahlswe, Cai, Li, Yeung, 2000]
The butterfly example: Routing
The butterfly example: Routing
The butterfly example: Routing
The butterfly example: Routing
The butterfly example: Routing

1.5 bits per cycle
The butterfly example: Network Coding
The butterfly example: Network Coding
The butterfly example: Network Coding
The butterfly example: Network Coding
The butterfly example: Network Coding
The butterfly example: Network Coding

2 bits per cycle
Practical network coding
Problem

No one knows the network structure.

Source/destination do not control intermediate nodes.
Problem

No one knows the network structure.

Source/destination do not control intermediate nodes.
Idea

Forward random combinations

\[ \alpha x_2 + \beta x_2 \]

[Chou, Wu, Jain/ Ho, Kötter, Medard, Karger, Effros 2003]
How can that possibly work?

\[ \alpha_1 \gamma_1 + \beta_1 \]

\[ \alpha_2 \gamma_2 + \beta_2 \]
Wait a minute... 

How am I supposed to figure out $\alpha$, $\beta$, $\gamma$?
Wait a minute... 

How am I supposed to figure out $\alpha$, $\beta$, $\gamma$?
Idea: Header

Input:

\[
\begin{align*}
\text{Red} &= [1 \ 0 \ | \ \cdots \ x_1 \ \cdots ] \\
\text{Yellow} &= [0 \ 1 \ | \ \cdots \ x_2 \ \cdots ]
\end{align*}
\]
Idea: Header

Output:

\[ \begin{align*}
? \quad \text{apple} & \quad ? \quad \text{orange} = [\alpha \ | \ \cdot \cdot \cdot \ | \ \alpha x_1 + \gamma x_2 \cdot \cdot \cdot ] \\
? \quad \text{apple} & \quad ? \quad \text{orange} = [\delta \ | \ \cdot \cdot \cdot \ | \ \delta x_1 + \beta x_2 \cdot \cdot \cdot ]
\end{align*} \]
Noisy network coding
Rank Metric Channel (Symmetric Network Channel) (Gabidulin, 1985)

\[ y = x \oplus z, \]

where \( z \) is uniformly random with \( \text{rank}(z) = \ell \omega \).
Asymptotics, Rate, Capacity

\[ \ell = N\lambda, \ m = N(1 - \lambda), \ N \to \infty \]

\[ R = \frac{\log_2 |\text{Code}|}{m\ell}, \]

\[ C(\lambda, \omega) = \frac{1 - \lambda - \omega + \lambda\omega^2}{1 - \lambda}. \]
\( \ell = N\lambda, \ m = N(1 - \lambda), \ N \to \infty \)

\[ R = \frac{\log_2 |\text{Code}|}{ml}, \]

\[ C(\lambda, \omega) = \frac{1 - \lambda - \omega + \lambda\omega^2}{1 - \lambda}. \]
Asymptotics, Rate, Capacity

\[ \ell = N\lambda, \ m = N(1 - \lambda), \ N \to \infty \]

\[ R = \frac{\log_2 |\text{Code}|}{ml}, \]

\[ C(\lambda, \omega) = \frac{1 - \lambda - \omega + \lambda\omega^2}{1 - \lambda}. \]
Matrix channels

\[ x = \{ x_{ij} \} \in \mathbb{F}_2^{m \times \ell} \]
Network coding???

(Kötter, Kschischang, 2007)

\[ \begin{bmatrix}
1 \\
1 \\
1 \\
1
\end{bmatrix} \equiv \begin{bmatrix}
1 \\
\vdots \\
\vdots \\
1
\end{bmatrix} \begin{bmatrix} x \\ \vdots \end{bmatrix} \]

(Chou, Wu, Jain, 2003)
Reliable network

\[
\begin{bmatrix}
1 & x
\end{bmatrix} \xrightarrow{\text{NET}} \begin{bmatrix}
G & Gx
\end{bmatrix} \xrightarrow{G^{-1}} \begin{bmatrix}
1 & x
\end{bmatrix}
\]
Faulty network

\[
\begin{bmatrix}
1 & x \\
\end{bmatrix}
\xrightarrow{\text{NET}}
\begin{bmatrix}
G & Gx + z' \\
\end{bmatrix}
\xrightarrow{G^{-1}}
\begin{bmatrix}
1 & x + G^{-1}z' \\
\end{bmatrix}
\]
Binary Erasure Channel

(Elias, 1954)

\[ y_{ij} = \begin{cases} 
  x_{ij} & \text{with prob. } \epsilon, \\
  * & \text{with prob. } 1 - \epsilon.
\end{cases} \text{ independently} \]
\ell = N\lambda, \ m = N(1 - \lambda), \ N \to \infty

\begin{align*}
R & = \frac{\log_2 |\text{Code}|}{m\ell}, \\
C(\epsilon) & = 1 - \epsilon.
\end{align*}
\( \ell = N\lambda, \ m = N(1 - \lambda), \ N \to \infty \)

\[
R = \frac{\log_2 |\text{Code}|}{m\ell},
\]

\[
C(\epsilon) = 1 - \epsilon.
\]
Asymptotics, Rate, Capacity

\[ \ell = N\lambda, \ m = N(1 - \lambda), \ N \to \infty \]

\[ R = \frac{\log_2 |\text{Code}|}{m\ell}, \]

\[ C(\epsilon) = 1 - \epsilon. \]
Construction and decoding algorithm
Equivalent description of the channel

\[ x(m) = x^{(1)} x^{(2)} x^{(3)} \]
Equivalent description of the channel

\[ Z \equiv \text{uniformly random subspace \( \subseteq F_2^\ell \)} \]
\[ z^{(1)}, \ldots, z^{(m)} \in Z \]
Information is not oranges
Practical network coding
Noisy network coding
Construction and decoding algorithm
Achieving capacity

Code

$m - \ell \omega$ ‘symbols’ $\rightarrow$ LDPC

$\ell \omega$ ‘symbols’ $\rightarrow$ learn the error space $Z$
Information is not oranges
Practical network coding
Noisy network coding
Construction and decoding algorithm
Achieving capacity

**LDPC: Check nodes**

\[ h_1x^{(1)} + h_2x^{(2)} + h_3x^{(3)} = 0 \]

'Edge labels' \( h_i \in \mathbb{F}_2^{\ell \times \ell} : \ell \times \ell \) matrices

Montanari, Rathi, Urbanke
Iterative Coding for Network Coding
LDPC: Check nodes

\[ h_1x^{(1)} + h_2x^{(2)} + h_3x^{(3)} = 0 \]

'Edge labels' \( h_i \in \mathbb{F}_2^{\ell \times \ell} : \ell \times \ell \) matrices
LDPC: Check nodes

\[ h_1 x^{(1)} + h_2 x^{(2)} + h_3 x^{(3)} = 0 \]

‘Edge labels’ \( h_i \in \mathbb{F}_2^{\ell \times \ell} : \ell \times \ell \) matrices
Information is not oranges
Practical network coding
Noisy network coding
Construction and decoding algorithm
Achieving capacity

LDPC: Variable nodes

\[ x^{(i)} \in \{ z^{(i)} + Z \} \]

Degree = 2!
Message passing decoder

Messages → Affine subspaces $V_{i \rightarrow a} \subseteq \mathbb{F}_2^\ell$

Operations → Subspace intersections/sums
Message passing decoder: Check nodes

$h_1 x^{(1)} + h_2 x^{(2)} + h_3 x^{(3)} = 0$

$V_3 = h_3^{-1}(h_1 V_1 + h_2 V_2)$
Message passing decoder: Check nodes

\[ h_1 x^{(1)} + h_2 x^{(2)} + h_3 x^{(3)} = 0 \]

\[ V_3 = h_3^{-1}(h_1 V_1 + h_2 V_2) \]
Message passing decoder: Check nodes

\[ h_1 x^{(1)} + h_2 x^{(2)} + h_3 x^{(3)} = 0 \]

\[ V_3 = h_3^{-1} (h_1 V_1 + h_2 V_2) \]
Message passing decoder: Variable nodes

\[ x^{(i)} \in \{z^{(i)} + Z\} \]

\[ V_2 = V_1 \cap \{z^{(i)} + X\} \]
Message passing decoder: Variable nodes

\[ x^{(i)} \in \{ z^{(i)} + Z \} \]

\[ V_2 = V_1 \cap \{ z^{(i)} + X \} \]
Message passing decoder: Variable nodes

\[ x^{(i)} \in \{z^{(i)} + Z\} \]

\[ V_2 = V_1 \cap \{z^{(i)} + X\} \]
Achieving capacity
Capacity achieving ensemble

Variable degree:

\[ P_i^{(k)} = \frac{2k(k-1)}{i(i-1)(i-2)}. \]

Check degree:

Not the soliton!
Variable degree:

\[ v_i = 2 \]

Check degree:

\[ P^{(k)}_i = \frac{2k(k - 1)}{i(i - 1)(i - 2)} \]

Not the soliton!
Capacity achieving ensemble

Variable degree:

\[ 2 \]

Check degree:

\[ P_i^{(k)} = \frac{2k(k - 1)}{i(i - 1)(i - 2)}. \]
Capacity achieving ensemble

Variable degree:

\[ V_i = 2 \]

Check degree:

\[ P_i^{(k)} = \frac{2k(k - 1)}{i(i - 1)(i - 2)}. \]

Not the soliton!
Theorem

If \( \omega = 1/k \), then the ensemble \((2, P^{(k)})\) has rate equal to the capacity of the rank metric channel and achieves vanishing error probability under message passing decoding.

Further, it achieves vanishing error probability over the erasure channel.
Capacity achieving ensemble

**Theorem**

If $\omega = 1/k$, then the ensemble $(2, P^{(k)})$ has rate equal to the capacity of the rank metric channel and achieves vanishing error probability under message passing decoding. Further, it achieves vanishing error probability over the erasure channel.
Encoding complexity \( O(m\ell^2) \)

Decoding complexity \( O(m\ell^2) \)

Error probability \( \exp[-\Omega(\gamma^{\text{iter}})], \exp(-\Omega(m, \ell)) \)

Realistic example (from Chou et al.):

\( \ell \approx 50, \quad m \approx 1700 \) (over \( \mathbb{F}_{2^8} \))
Information is not oranges
Practical network coding
Noisy network coding
Construction and decoding algorithm
Achieving capacity

Encoding complexity \( O(m\ell^2) \)
Decoding complexity \( O(m\ell^2) \)
Error probability \( \exp[-\Omega(\gamma^{\text{iter}})], \exp(-\Omega(m, \ell)) \)

Realistic example (from Chou et al.):
\[ \ell \approx 50, \quad m \approx 1700 \quad (\text{over } \mathbb{F}_{2^8}) \]
The magic

1. Fix number of iterations.

2. For large $m$, message $V_i$ uniformly random conditional on dimension.

3. For large $\ell$, output dimension determined by input dimension.

4. For $\omega = 1/k$, only dimension $\ell \omega$ or 0 is possible.
The magic

1. Fix number of iterations.

2. For large $m$, message $V_i$ uniformly random conditional on dimension.

3. For large $\ell$, output dimension determined by input dimension.

4. For $\omega = 1/k$, only dimension $\ell \omega$ or 0 is possible.
The magic

1. Fix number of iterations.

2. For large $m$, message $V_i$ uniformly random conditional on dimension.

3. For large $\ell$, output dimension determined by input dimension.

4. For $\omega = 1/k$, only dimension $\ell \omega$ or 0 is possible.
1. Fix number of iterations.

2. For large $m$, message $V_i$ uniformly random conditional on dimension.

3. For large $\ell$, output dimension determined by input dimension.

4. For $\omega = 1/k$, only dimension $\ell \omega$ or 0 is possible.
“For large $\ell$, output dimension determined by input dimension”

$d_{1 \cap 2} \approx \max(d_1 + d_2 - d_{amb}, 0)$. 
“For large $\ell$, output dimension determined by input dimension”

\[ d_{1\cap2} \approx \max(d_1 + d_2 - d_{amb}, 0). \]
Simulations at small packet sizes

\( m = 42, \ \ell = 136 \quad (LP \text{ optimized degree sequence}) \)
Conclusion 1

\(q\)-ary LDPC’s (Davey, MacKay/ Burshtein, Miller/ Rathi, Urbanke)

Mixed outcomes

Exact density evolution for large \(q\)

Capacity achieving ensembles
Conclusion 1

\(q\)-ary LDPC’s (Davey, MacKay/ Burshtein, Miller/ Rathi, Urbanke)
Mixed outcomes

Exact density evolution for large \(q\)

Capacity achieving ensembles
Conclusion 2

LDPC codes achieve capacity (under message passing decoding) over the erasure channel (Luby et al. 97) ...

... and the rank metric channel
LDPC codes achieve capacity (under message passing decoding) over the erasure channel (Luby et al. 97) . . .

. . . and the rank metric channel
LDPC codes achieve capacity (under message passing decoding) over the erasure channel (Luby et al. 97) …

… and the rank metric channel