



# ネットワークコーディング の技術動向

松田崇弘、滝根哲哉

NICT九州リサーチセンター

大阪大学大学院工学研究科  
極限コミュニケーションラボラトリ



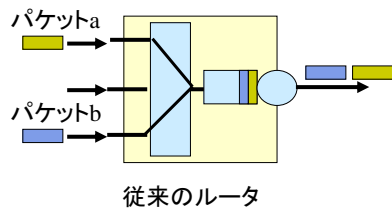
## ネットワークコーディングに関する研究

- ネットワークコーディングに関する最初の論文
  - [1] R. Ahlswede, et al., "Network Information Flow," IEEE Transactions on Information Theory, Vol. 46, No. 4, pp-1204-1216, Jul. 2000.
- 国際会議等
  - ISIT 2007 (International Symposium on Information Theory): ネットワークコーディングに関する7つのセッション
  - NetCod (Workshop on Network Coding, Theory, and Applications)
- 実用化
  - Microsoft p2p型ファイル交換システム"Avalanche"
    - [2] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," Proc. INFOCOM 2005, No.4, pp.2235-2245, Mar. 2005.

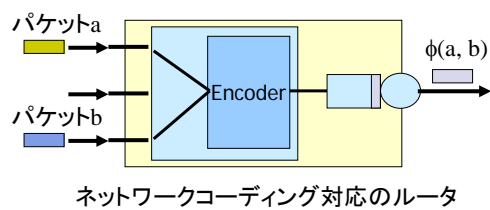


# ネットワークコーディングとは

- 従来のルータ(中継ノード)の役割
  - ルーティング、フォワーディング
- ネットワークコーディングの機能を備えたルータ
  - 中継ノードにパケット同士の演算(符号化)をさせる機能を付加する



従来のルータ

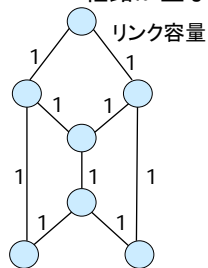


ネットワークコーディング対応のルータ

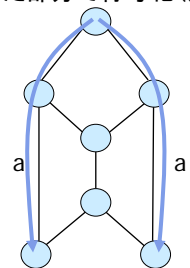


# ネットワークコーディングとは

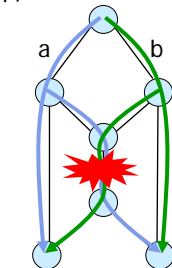
- マルチキャストにおいて複数の受信ノードへの最大フロー(ネットワークの伝送容量の最大値)を実現
- 各送受信ノード対に対し、複数経路を使用
- 経路が重なった部分で符号化(演算)



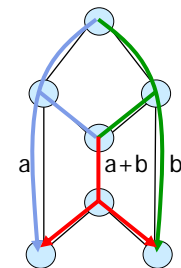
ネットワークポロジ



通常のマルチキャスト



同時に最大フローを実現することはできない



$a, b$   $a, b$   
 $a+b$  は論理演算  
ネットワークコーディング



# ネットワークコーディング定理



- [1] R. Ahlswede, et al., "Network Information Flow," IEEE Transactions on Information Theory, Vol. 46, No. 4, pp-1204-1216, Jul. 2000.
- 任意のネットワークにおいて、最大フローを実現するための符号が存在する。
  - 十分大きな符号空間
  - 送信ノード  $s$
  - 受信ノード  $t_i (i = 1, 2, \dots, L)$   $L$ は受信ノード数
  - $s$ から  $t_i$ までの最大フロー  $f_i^{\max}$
  - ネットワークコーディングにより

$$f_{\max} = \min_{1 \leq i \leq L} \{f_i^{\max}\}$$

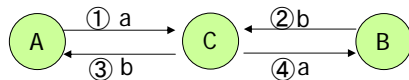
のスループットを実現



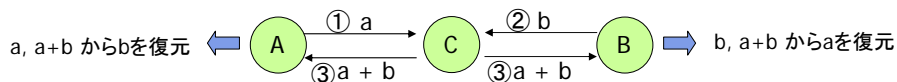
# 応用例 ～無線ネットワーク～



- $A \rightarrow B, B \rightarrow A$ への通信
- $A \leftrightarrow C, B \leftrightarrow C$ が無線リンク
- 符号化により、パケットの送信回数を削減
  - 帯域, 消費電力の節約



ネットワークコーディングを使用しない場合



ネットワークコーディングを使用した場合



# ガロア体

- パケット同士を演算するには？
- パケットを有限体(ガロア体)とみなして演算を行う。
- 体
  - 要素間で四則演算が定義された集合
    - 実数体、複素数体
- ガロア体 $GF(q)$ : 要素数が有限である体
  - $q$ : 位数 元の個数
- 最も単純なガロア体  $GF(2)=\{0, 1\}$ 
  - 加算  $0+0=0, 0+1=1, 1+0=1, 1+1=0$
  - 減算  $-x = x$
  - 乗算  $0 \cdot 0=0, 0 \cdot 1=0, 1 \cdot 0=0, 1 \cdot 1=1$
  - 除算  $1^{-1}=1, 0^{-1}=\text{不定}$
- $GF(2^m)$ :  $2^m$ 個の要素数
  - **mビットのパケットとみなす。**
  - **パケット同士の演算を定義することが可能**

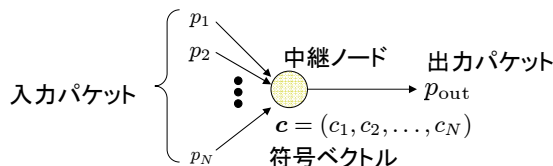
GF(2<sup>4</sup>)

番号	要素	ビット表現
1	0	0000
2	1	0001
3	$\alpha$	0010
4	$\alpha^2$	0100
5	$\alpha^3$	1000
6	$\alpha^4$	0011
7	$\alpha^5$	0110
8	$\alpha^6$	1100
9	$\alpha^7$	1011
10	$\alpha^8$	0101
11	$\alpha^9$	1010
12	$\alpha^{10}$	0111
13	$\alpha^{11}$	1110
14	$\alpha^{12}$	1111
15	$\alpha^{13}$	1101
16	$\alpha^{14}$	1001

# 線形ネットワークコーディング(1/3)

[3] S. Li and R. W. N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371-381, 2003.

- 符号ベクトルが各中継ノードに割当てられている
- 複数の入力パケットを線形演算後、合成して出力



$$p_{out} = c_1 p_1 + c_2 p_2 + \dots + c_N p_N$$

$$c_i, p_i \in GF(2^m)$$

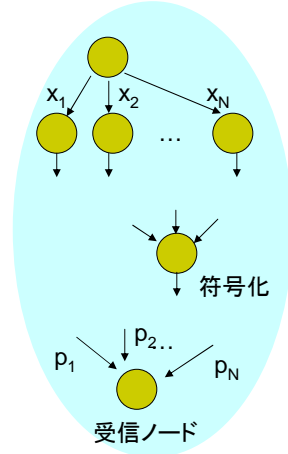
積・和は有限体(ガロア体)  $GF(2^m)$  上で定義された演算

## 線形ネットワークコーディング(2/3)

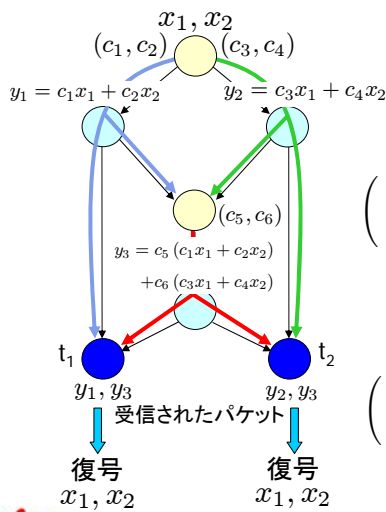
- 中継ノードで多段にわたって符号化を行う。
- 受信したN個の packets  $p = (p_1, p_2, \dots, p_N)^T$  から元の packets  $x = (x_1, x_2, \dots, x_N)^T$  を復号
- 受信 packets は元の packets の線形演算  
⇒ 連立方程式を解くことによって復号

$$p = Cx = \begin{pmatrix} \tilde{c}_1 \\ \tilde{c}_2 \\ \vdots \\ \tilde{c}_N \end{pmatrix} x = \begin{pmatrix} \tilde{c}_{1,1} & \tilde{c}_{1,2} & \dots & \tilde{c}_{1,N} \\ \tilde{c}_{2,1} & \tilde{c}_{2,2} & \dots & \tilde{c}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{c}_{N,1} & \tilde{c}_{N,2} & \dots & \tilde{c}_{N,N} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

$x = C^{-1}p$  より packets を復元  
 $C^{-1}$  が存在すれば復号可能



## 線形ネットワークコーディング(3/3)



受信ノード $t_1$ の受信 packets

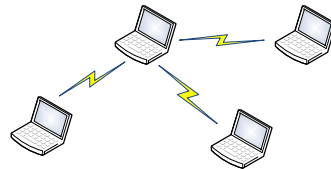
$$\begin{pmatrix} y_1 \\ y_3 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ c_1c_5 + c_3c_6 & c_2c_5 + c_3c_6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

受信ノード $t_2$ の受信 packets

$$\begin{pmatrix} y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} c_3 & c_4 \\ c_1c_5 + c_3c_6 & c_2c_5 + c_3c_6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

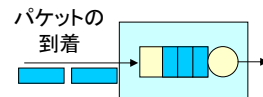
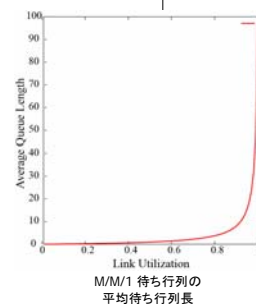
## ネットワークコーディングの応用技術

- 物理ネットワーク、論理ネットワーク？
  - ルータ: 高速な符号化処理が要求される。  
⇒p2p、アドホックネットワークへの適用が中心
- 有線、無線？
  - 有線: 複数の経路が重なるような経路制御を考える必要がある。
  - 無線: そもそも経路が冗長に作られやすい
    - 隣接ノード全てにパケットが送られる。
 ⇒ネットワークコーディングを適用しやすい。
- ネットワークコーディングの応用技術
  - p2pへの応用
    - アプリケーションレベルマルチキャスト
    - ファイル配信
  - 無線アドホックネットワークへの応用
    - 疎なネットワークにおける感染型ルーティング
    - 密なネットワークにおけるブロードキャスト



## ネットワークコーディングの効果

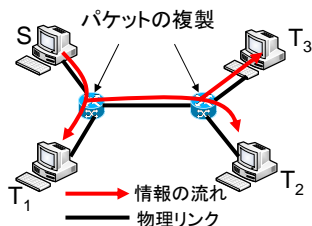
- 伝送効率の向上
  - マルチキャスト通信における最大スループット (最大フロー)の実現
    - あまり現実的でない。
  - 使用帯域の削減
    - 同じスループットでもネットワークコーディングを用いた方がネットワーク内を流れるパケット数が少ない。
- ロバスト性の強化
  - 一つの符号化されたパケットは元のパケットの情報を複数個含んでいる。
- セキュリティ[4]
  - 符号化されたパケットを必要な数だけ受け取れないと復号できない。
  - 符号化の方法(符号ベクトル等)がわからないと復号できない。



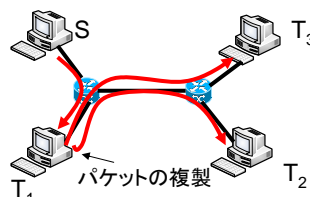
# アプリケーションレベルマルチキャストへの応用(1/2)



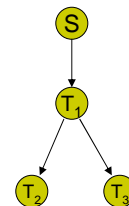
- 1対多通信の実現
  - IPマルチキャスト
  - アプリケーションレベルマルチキャスト
- アプリケーションレベルマルチキャスト(p2pマルチキャスト)
  - 論理ネットワーク上でマルチキャスト通信を行う。
  - ルータにマルチキャストの機能を実装する必要がない。
- 技術課題
  - パケットが冗長な経路を流れる
 ⇒物理ネットワーク上に流れるトラフィックが増加



IPマルチキャスト  
(物理ネットワーク上でのマルチキャスト)  
独立行政法人  
情報通信研究機構



アプリケーションレベルマルチキャスト  
(論理ネットワーク上でのマルチキャスト)



アプリケーションレベルマルチキャスト  
の論理ネットワーク  
大阪大学大学院工学研究科  
情報コミュニケーションラボラトリ

# アプリケーションレベルマルチキャストへの応用(2/2)

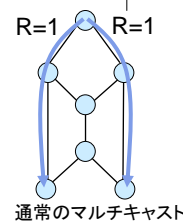


[5] Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 107-120, Jan. 2004.

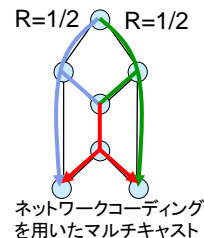
[6] 中井隆幸, 野口 拓, 松田崇弘, 滝根哲哉 "ネットワークコーディングを用いたアプリケーションレベルマルチキャストのための経路構築法", 電子情報通信学会技術報告IN, 2008年3月(発表予定)

- N本の経路を用い、送信レートを $1/N$ に下げ、並列伝送。
- 経路が重なるリンクでネットワークコーディングを用いる。
  - スループットを維持しながら、物理ネットワークに流れるトラフィック量を削減

送信レート  $R$   
 $N=2$ の場合



$R=1/2$   $R=1/2$

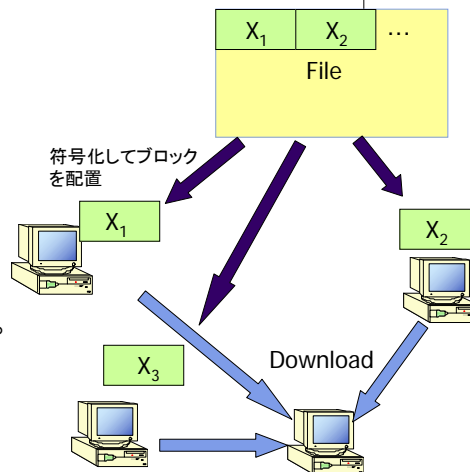


NICT 独立行政法人  
情報通信研究機構

大阪大学大学院工学研究科  
情報コミュニケーションラボラトリ

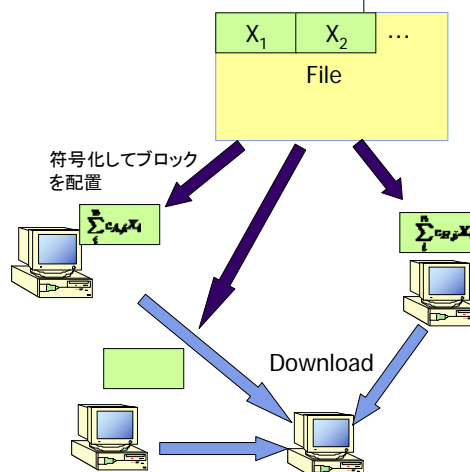
## ファイル配信技術(1/2)

- ファイルを複数のブロックに分割して, ユーザ(Peer)に分散配置.
- 複数のPeerから平行してブロックをダウンロードする.
  - 元もファイルを構成するブロックを全て受信する必要がある.



## ファイル配信技術(2/2)

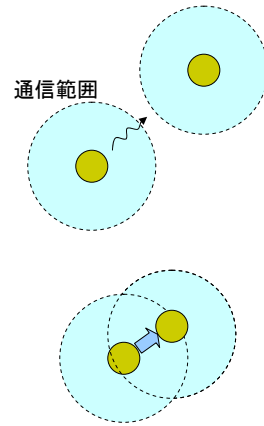
- [2] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," Proc. INFOCOM 2005, No.4, pp.2235-2245, Mar. 2005.
- ブロックをネットワークコーディングにより符号化して配置
  - 元ファイルを構成する全ての情報が各ブロックに含まれる
  - 線形独立なブロックをもつPeerが存在すればファイルを復元可能
- 複数のPeerから平行してブロックをダウンロードする.
- どのPeerがどのブロックをもっているかを気にする必要はなく, 符号ベクトルが線形独立であればよい.





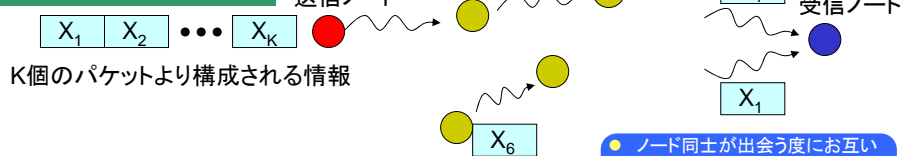
## 感染型ルーチング への応用(1/2)

- [7] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "On the Benefits of Random Linear Coding for Unicast Applications in Disruption Tolerant Networks," *IEEE Network Coding Workshop*, 2006.
- 疎密度モバイルアドホックネットワーク
  - 各移動ノードの通信範囲内には、大部分の時間、他のノードが存在しない。
- 感染型ルーチング (Epidemic Routing)
  - DTN (Delay Tolerant Network) における代表的なルーチング
  - ノードが送信もしくは中継したいパケットを保持している場合
    - 通信範囲内に他の端末がいる場合⇒パケットを複製して渡す(感染)
    - 通信範囲内に他の端末がいない場合⇒保持したまま移動



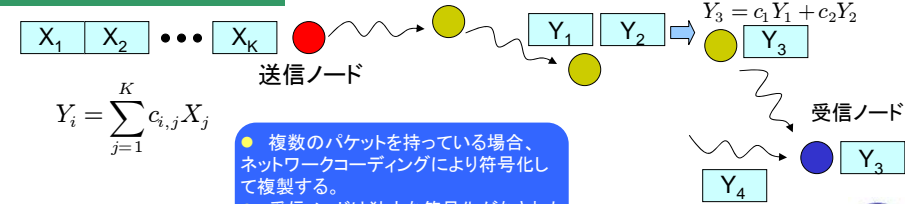
## 感染型ルーチング への応用(2/2)

### 通常の感染型ルーチング



- ノード同士が出会う度にお互い持っていないパケットを複製する。
- 受信ノードは同じパケットを受信する可能性がある。

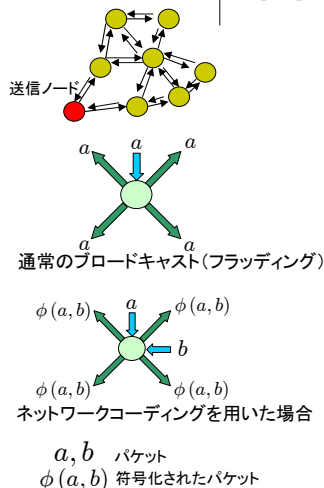
### ネットワークコーディングを用いた感染型ルーチング



- 複数のパケットを持っている場合、ネットワークコーディングにより符号化して複製する。
- 受信ノードは独立な符号化がなされたパケットをK個受け取れば復号可能

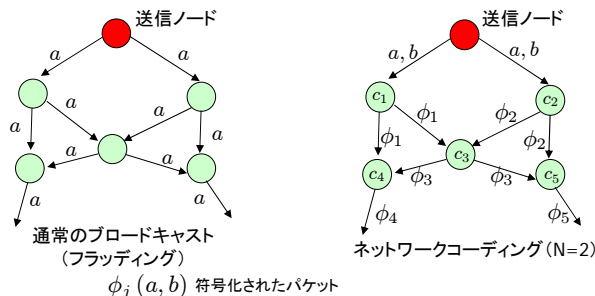
## 無線アドホックネットワークにおけるブロードキャスト通信(1/4)

- [8] 松田崇弘, 野口拓, 滝根哲哉, “ランダムネットワークコーディングを用いたアドホックブロードキャスト,” 電子情報通信学会技術研究報告. IN2006-74, pp. 145-150, 2006年.
- 無線アドホックネットワークにおけるブロードキャスト
  - ブロードキャストストーム問題 (broadcast storm problem)
  - 無線リンクの同報性から、パケットがネットワーク中に氾濫  $\Rightarrow$  衝突の増加
- 符号化により複数のパケットを集約
  - 送信パケット数に対する滞留パケット数を削減
    - 低消費電力化
    - パケットの衝突を抑える
- ノード密度の高いネットワークでは、各ノードは複数個の符号化されたパケットを受信  $\Rightarrow$  復号可能



## 無線アドホックネットワークにおけるブロードキャスト通信(2/4)

- 提案システムの概要
- 送信ノードの動作
  - N個のパケットを1つのブロックとする。
  - 1ブロック当たりN個のパケットを隣接ノードに送出
- 中継ノード(受信ノードの動作)
  - N個の異なるパケットを受信すると、符号化して1つのパケットを送出

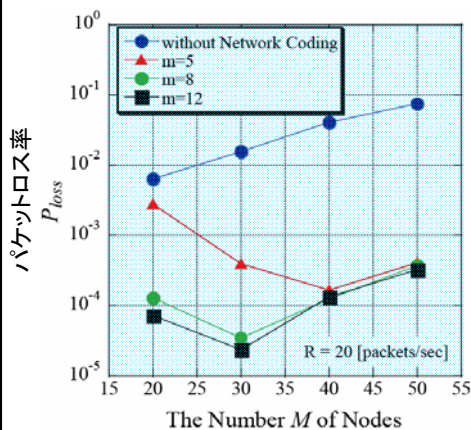


## 無線アドホックネットワークにおけるブロードキャスト通信(3/4)



- ノード数  $M=20, 30, 40, 50$
- ノードの配置:  $100[m] \times 100[m]$ の平面にランダムに配置
- 送信ノード数 1
- パケットの送信レート  $R=20$  [packets/sec]
- 無線リンク速度 5Mbps
- パケット長1000 [bytes]
- 各ノードの通信半径 50m
- ガロア体 $GF(2^m)$ :  $m = 5 \sim 12$
- 符号ベクトルの大きさ  $N=2$
- MACプロトコル non-persistent CSMA
  - ビジー検知時には(0, 50msec)の範囲でランダムバックオフ

## 無線アドホックネットワークにおけるブロードキャスト通信(4/4)



パケットロス率

$$= 1 - \frac{\text{復号に成功したパケット数}}{\text{送信したパケット数}}$$

- ネットワークコーディングによりパケットロス率が大幅に改善
- $m$ が小さい場合には、復号に失敗する確率が増加するため、改善度が小さい。
- ノード数が少ない $\Rightarrow$ 復号に必要なパケット数が得られない場合が増加
- ノード数が多い $\Rightarrow$ MAC層での衝突が増加

## まとめ

- ネットワークコーディングについて
  - ネットワークコーディングの概要
    - 線形ネットワークコーディング
  - 応用例
    - p2pへの応用
    - 無線アドホックネットワーク
- 今後の課題
  - 符号化方法
  - 経路制御
  - ネットワークコーディングの応用技術
    - ネットワーク内部で符号化することに意味のあるアプリケーション



## 参考文献

- [1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204-1216, 2000.
- [2] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," Proc. INFOCOM 2005, No.4, pp.2235-2245, Mar. 2005.
- [3] S. Li and R. W. N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371-381, 2003.
- [4] N. Cai and R. W. Yeung, "Secure network coding," *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, 2002.
- [5] Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 107-120, Jan. 2004.
- [6] 中井隆幸, 野口 拓, 松田崇弘, 滝根哲哉 "ネットワークコーディングを用いたアプリケーションレベルマルチキャストのための経路構築法", 電子情報通信学会技術報告IN, 2008年3月(発表予定)
- [7] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "On the Benefits of Random Linear Coding for Unicast Applications in Disruption Tolerant Networks," *IEEE Network Coding Workshop*, 2006.
- [8] 松田崇弘, 野口拓, 滝根哲哉, "ランダムネットワークコーディングを用いたアドホックブロードキャスト," 電子情報通信学会技術研究報告. IN2006-74, pp. 145-150, 2006年.



## 付録



- ネットワークコーディングに必要な手続き
  - 符号ベクトルの決定・配置
    - 符号化を行うノードに配置する符号ベクトル
    - 各ノードで自律分散的に符号ベクトルを決める方法
      - ランダムネットワークコーディング
  - 符号ベクトルの配信
    - 受信ノードは、符号化されたパケットがどのような符号ベクトルで符号化されているのかを知る必要がある。
      - パケットに符号ベクトル情報を埋め込んでおく。



## ランダムネットワークコーディング

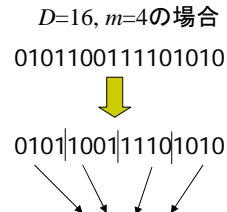


- T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," *Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.
- 各ノードにおける符号ベクトル  $c = (c_1, c_2, \dots, c_N)$
- $c_i$  をガロア体  $GF(2^m)$  からランダムに選択
  - 各ノードで自律分散的に符号ベクトルを生成可能
- 受信ノードで復号不可能となる場合がある。
  - 逆行列  $C^{-1}$  が存在しない。
- 無線ネットワーク、p2pなどパケットロスやトポロジー変動が頻繁に発生する場合に有利
- ランダムネットワークコーディングの性能
  - $m$ 、 $N$ に依存
  - $m$ 、 $N$ とネットワーク性能とのトレードオフ



## 符号ベクトルのパケットへの埋め込み(1/4)

- P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Allerton Conference on Communication, Control, and Computing*, 2003.
- パケット内部に符号化に使用されている符号ベクトルを埋め込む
- データ長  $D=2^r$  と仮定
- $D$  を  $m$  ビット毎の  $B=D/m$  のブロックに分割
- 各ブロックを  $GF(2^m)$  の要素とみなす
- $\mathbf{c} = (c_1, c_2, \dots, c_N)$  で符号化されている場合  
元データ  $(a_{k,1}, a_{k,2}, \dots, a_{k,B})$  ( $k = 1, 2, \dots, N$ ) とすると、



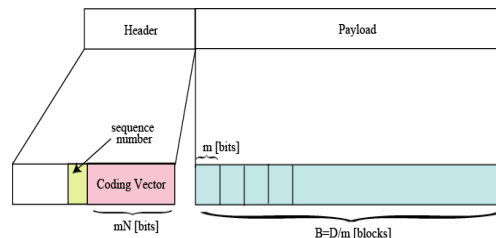
符号化されたパケット

$$(d_1 \ d_2 \ \dots \ d_B) = (c_1 \ c_2 \ \dots \ c_N) \begin{pmatrix} a_{1,1} & a_{2,1} & \dots & a_{1,B} \\ a_{2,1} & a_{2,2} & \dots & a_{2,B} \\ \vdots & \vdots & & \vdots \\ a_{N,1} & a_{N,2} & \dots & a_{N,B} \end{pmatrix}$$

ガロア体の要素と考える

## 符号ベクトルのパケットへの埋め込み(2/4)

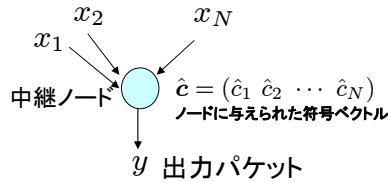
- 符号化方法
  - **ペイロード部分のブロック化**
    - ガロア体  $GF(2^m)$
    - 各ブロックの大きさ  $m$  [bits]
  - 各ブロックをガロア体の要素とみなす
  - **符号ベクトルをパケット内部に埋め込む**
    - 例  $m=8, N=4$  の場合、32ビットの冗長性付加



## 符号ベクトルのパケットへの埋め込み(3/4)



中継ノードでの符合化



到着パケット

(符号ベクトルを含んだ形で表記)

$$\mathbf{x}_k = (\mathbf{c}_k \mathbf{d}_k)$$

$$\mathbf{c}_k = (c_{k,1} \ c_{k,2} \ \dots \ c_{k,N})$$

$$\mathbf{d}_k = (d_{k,1} \ d_{k,2} \ \dots \ d_{k,B})$$

$$\mathbf{y} = \hat{\mathbf{c}} (\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N)^T$$

$$= (\hat{c}_1 \ \hat{c}_2 \ \dots \ \hat{c}_N) \begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,N} & d_{1,1} & d_{1,2} & \dots & d_{1,B} \\ c_{2,1} & c_{2,2} & \dots & c_{2,N} & d_{2,1} & d_{2,2} & \dots & d_{2,B} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ c_{N,1} & c_{N,2} & \dots & c_{N,N} & d_{N,1} & d_{N,2} & \dots & d_{N,B} \end{pmatrix}$$

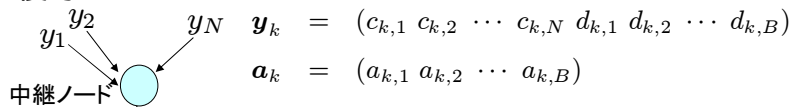
$$= (\mathbf{c}_{\text{out}} \ \mathbf{d}_{\text{out}})$$



## 符号ベクトルのパケットへの埋め込み(4/4)



復号化



$a_1, a_2, \dots, a_N$  元パケットの復元

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,N} \\ c_{2,1} & c_{2,2} & \dots & c_{2,N} \\ \vdots & \vdots & & \vdots \\ c_{N,1} & c_{N,2} & \dots & c_{N,N} \end{pmatrix}^{-1} \begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,N} & d_{1,1} & d_{1,2} & \dots & d_{1,B} \\ c_{2,1} & c_{2,2} & \dots & c_{2,N} & d_{2,1} & d_{2,2} & \dots & d_{2,B} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ c_{N,1} & c_{N,2} & \dots & c_{N,N} & d_{N,1} & d_{N,2} & \dots & d_{N,B} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & \dots & 0 & a_{1,1} & a_{1,2} & \dots & a_{1,B} \\ 0 & 1 & \dots & 0 & a_{2,1} & a_{2,2} & \dots & a_{2,B} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & & 1 & a_{N,1} & a_{N,2} & \dots & a_{N,B} \end{pmatrix}$$

元のデータ

