

ネットワーク符号化通信における近隣ノードの未受信シンボル推定に基づく再符号化手法

大塚 祐輔[†] 山本 泰資[†] 廣友 雅徳[‡] 福田 洋治^{††} 毛利 公美^{‡‡} 白石 善明^{†††}
[†]名古屋工業大学 [‡]佐賀大学 ^{††}愛知教育大学 ^{‡‡}岐阜大学 ^{†††}神戸大学

1. はじめに

無線アドホックネットワークを用いたコンテンツ配信が考えられている[1]。無線アドホック通信では、ノードの移動によるリンク切断や、トラフィック増加によるパケット衝突、輻輳が原因でパケットロスが発生してデータが欠落する。データが欠落することによって配信データを復元できるノードの減少につながる。コンテンツ配信でのパケットロス耐性やスループットの向上を目的にネットワーク符号化が用いられている[1][2][3]。ネットワーク符号化では、ノードは所持しているデータと符号化ベクトルを入力とし符号化を行い、符号化データを出力する。ネットワーク符号化の一つであるランダム線形ネットワーク符号化 (Random Linear Network Coding : RLNC) では符号化ベクトルをパケットに含めて送信する[3][4]。パケットを受信したノードは、符号化ベクトルと符号化データを取り出し、復号する。

シンボル x_1, x_2, \dots, x_N を復号するには、 N 個の線形独立な符号化データを必要とする。他の符号化データと線形独立な関係になるものは innovative な符号化データ、他の符号化データと線形従属な関係になり復号処理に利用できないものは non-innovative な符号化データと呼ばれる。少ない通信回数で innovative な符号化データを受信させることができれば、遅延が減りスループットが向上する。

本稿では、RLNC 通信において、受信パケットの符号化ベクトルに注目する。受信パケットの符号化ベクトルから近隣ノードのデータ受信状況を推定し、未受信シンボルが含まれる符号化データを流すことで、近隣ノードに innovative な符号化データを受信させる手法を提案する。復号の計算量が少なく移動体アドホック通信への適用が容易と考えられる LT 符号をベースとした RLNC の拡張である LT Network Codes (LTNC) [5] に提案手法を適用した通信と適用しない通信を比較し、提案手法を適用することで innovative な符号化データを受信させることができることを確認する。

2. ランダム線形ネットワーク符号化 (RLNC)

パケットロス耐性やスループットを向上させるネットワーク符号化がある[3]。ネットワーク符号化の 1 つに符号化ベクトルを各ノードでランダムな値で生成する RLNC がある。無線アドホックネットワークのようにネットワークポロジが変化しやすい通信では、事前に符号化ベクトルを共有できないので RLNC を適用することが考えられている。以下に RLNC を適用した通信の流れを示す。通信モデルは送信処理を行う送信ノード、中継処理を行う中間ノード、受信処理を行う受信ノードから構成される。

【送信処理】

配信データ x を N 個のシンボルに分割する $x = (x_1, x_2, \dots, x_N)$ 。送信ノードは以下の手順を N 回繰り返す。

- Step1. 符号化ベクトル c をランダムに生成する。
- Step2. シンボル x_1, x_2, \dots, x_N と符号化ベクトル c を用いて符号化し符号化データ y を作成する。
- Step3. 符号化データ y と符号化ベクトル c をパケットで送信する。

【中継処理】

中間ノードは一定間隔ごとに受信した M 個の符号化パケットを用いて以下の処理を行う。

Re-encoding Method based on Inferring Unreceived Symbol of Neighbor Nodes for Network Coded Communication

[†] Yusuke OTSUKA and Taisuke YAMAMOTO · Nagoya Institute of Technology

[‡] Masanori HIROTOMO · Saga University

^{††} Youji FUKUTA · Aichi University of Education

^{‡‡} Masami MOHRI · Gifu University

^{†††} Yoshiaki SHIRAISHI · Kobe University

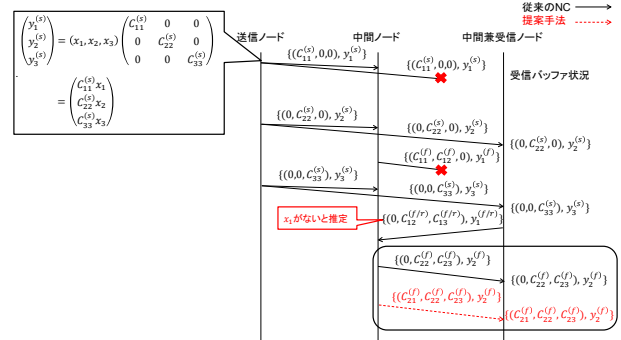


図1 送信先ノードの未受信シンボル推定の例

- Step1. 符号化ベクトル c をランダムに生成する。
- Step2. 受信した符号化データ y と符号化ベクトル c を用いて再符号化し \tilde{y} を作成する。

$$\tilde{y} = \tilde{c}y = \tilde{c}_1 y_1 + \tilde{c}_2 y_2 + \dots + \tilde{c}_M y_M$$

- Step3. シンボル x_1, x_2, \dots, x_N を復号できるように符号化ベクトル \tilde{c} と符号化データ y の符号化ベクトル行列 C を用いて符号化ベクトル \tilde{c} を作成する。

$$\tilde{c} = \tilde{c}C = (\tilde{c}_1, \tilde{c}_2, \dots, \tilde{c}_M) \begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,N} \\ c_{2,1} & c_{2,2} & \dots & c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M,1} & c_{M,2} & \dots & c_{M,N} \end{pmatrix}$$

- Step4. 符号化ベクトル \tilde{c} と再符号化データ \tilde{y} をパケットで送信する。

【受信処理】

受信ノードは N 個以上の符号化パケットを受信したとき、 N 個のパケットを用いて以下の処理を行う。

- Step1. 符号化データと符号化ベクトル行列を用いてシンボル x_1, x_2, \dots, x_N を復号する。

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} c'_{1,1} & c'_{1,2} & \dots & c'_{1,N} \\ c'_{2,1} & c'_{2,2} & \dots & c'_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c'_{N,1} & c'_{N,2} & \dots & c'_{N,N} \end{pmatrix}^{-1} \begin{pmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_N \end{pmatrix}$$

- Step2. 復号したシンボル x_1, x_2, \dots, x_N から配信データ x を復元する。

3. 中間ノードが受信したパケットから送信先の未受信シンボルを推定する手法

ネットワーク符号化通信では、少ない受信回数で復号に必要なパケットを集められることは、少ない通信回数で innovative な符号化データを受信させることを意味する。innovative な符号化データを受信させるためには、non-innovative な符号化データを送信しないというアプローチが考えられる。ノードが近隣ノードにとって innovative な符号化データを作成できるかどうかを知ることができれば、innovative な符号化データを送信できるようになる。

本章では、ノードが近隣ノードにとって innovative な符号化データを作成できるかどうかを知るために、受信したパケットに含まれる符号化ベクトルを用いた送信先ノードの未受信シンボルを推定する手法を提案する。図1は次の手順で示す符号化ベクトルを用いた送信先ノードの未受信シンボル推定の例である。

- Step1. 送信ノードはシンボル (x_1, x_2, x_3) を符号化行列

$$\begin{pmatrix} c_{11}^{(s)} & 0 & 0 \\ 0 & c_{22}^{(s)} & 0 \\ 0 & 0 & c_{33}^{(s)} \end{pmatrix} \text{で符号化する。} \begin{pmatrix} y_1^{(s)} \\ y_2^{(s)} \\ y_3^{(s)} \end{pmatrix} = \begin{pmatrix} c_{11}^{(s)} x_1 \\ c_{22}^{(s)} x_2 \\ c_{33}^{(s)} x_3 \end{pmatrix}$$

Step 2. 送信ノードは $\{(C_{11}^{(s)}, 0, 0), y_1^{(s)}\}$, $\{(0, C_{22}^{(s)}, 0), y_2^{(s)}\}$, $\{(0, 0, C_{33}^{(s)}), y_3^{(s)}\}$ を送信する。

Step 3. 中間ノードは受信した $\{(C_{11}^{(s)}, 0, 0), y_1^{(s)}\}$, $\{(0, C_{22}^{(s)}, 0), y_2^{(s)}\}$ を再符号化し $\{(C_{11}^{(f)}, C_{12}^{(f)}, 0), y_1^{(f)}\}$ を送信する。

Step 4. 中間兼受信ノードは受信した $\{(0, C_{22}^{(s)}, 0), y_2^{(s)}\}$, $\{(0, 0, C_{33}^{(s)}), y_3^{(s)}\}$ を再符号化し $\{(0, C_{12}^{(f/r)}, C_{13}^{(f/r)}), y_1^{(f/r)}\}$ を送信する。

Step 5. 中間ノードは受信した $\{(0, C_{12}^{(f/r)}, C_{13}^{(f/r)}), y_1^{(f/r)}\}$ の符号化ベクトル $(0, C_{12}^{(f/r)}, C_{13}^{(f/r)})$ から、中間兼受信ノードが符号化ベクトルの一番目のシンボルを持っていないと推定し、受信した $\{(C_{11}^{(s)}, 0, 0), y_1^{(s)}\}$, $\{(0, C_{22}^{(s)}, 0), y_2^{(s)}\}$, $\{(0, 0, C_{33}^{(s)}), y_3^{(s)}\}$, $\{(0, C_{12}^{(f/r)}, C_{13}^{(f/r)}), y_1^{(f/r)}\}$ から一番目のシンボルを含むように再符号化し $\{(C_{21}^{(f)}, C_{22}^{(f)}, C_{23}^{(f)}), y_2^{(f)}\}$ を送信する。

Step 6. 中間兼受信ノードは $\{(C_{21}^{(f)}, C_{22}^{(f)}, C_{23}^{(f)}), y_2^{(f)}\}$ を受信する。提案手法を適用しない場合、受信するパケットは $\{(0, C_{22}^{(f)}, C_{23}^{(f)}), y_2^{(f)}\}$ で、すでに受信しているパケット $\{(0, C_{22}^{(s)}, 0), y_2^{(s)}\}$, $\{(0, 0, C_{33}^{(s)}), y_3^{(s)}\}$ に対して、すでに所持しているシンボルしか含まれていないので、non-innovative なパケットである。一方で、提案手法を用いる場合、 $\{(C_{21}^{(f)}, C_{22}^{(f)}, C_{23}^{(f)}), y_2^{(f)}\}$ を受信する。すでに受信しているパケット $\{(0, C_{22}^{(s)}, 0), y_2^{(s)}\}$, $\{(0, 0, C_{33}^{(s)}), y_3^{(s)}\}$ に対して、所持していない一番目のシンボルを含んでいるので innovative なパケットである。このようにして、符号化ベクトルの値から近隣ノードが所持していないシンボルを知ることができる。そのシンボルを含んだ符号化データを所持する他のノードが再符号化し送信することで、ノードは所持していないシンボルを受信できることが期待される。受信ノードにとって、所持していないシンボルを含む符号化データを受信するということは、innovative な符号化データを受信することと同じであると言える。つまり、受信パケットの符号化ベクトルの値から、近隣ノードの所持していないシンボルを推定し、推定したシンボルを含む符号化データを用いて再符号化し、符号化データを送信することで近隣ノードは innovative な符号化データを受信できると考えられる。

4. 評価実験

4.1. 実験内容

LTNC[5]の通信モデルで、提案手法を適用した場合と適用していない場合の受信パケット数のオーバーヘッドとデータ普及率を評価する。オーバーヘッドは、以下の式で算出する。(α: オーバーヘッド, n: 配信データ復元時の平均受信パケット数, k: シンボル数)

$$n = (1 + \alpha)k.$$

配信データの普及率では、すべてのシンボルを復号したノードの数を計測する。受信パケット数のオーバーヘッドとデータ普及率を評価して、提案手法を適用することで innovative な符号化データを受信させられているかを確認する。ここでは移動体アドホック通信の一例として交差点での路車間通信と車車間通信を組み合わせた一対多の路車車間通信を行うモデルとして、路側機は送信ノード、車両は中間/受信ノードの役割を担うものとする。

4.2. 実験方法

ネットワークシミュレータ OMNeT++4.0[6]と無線移動体通信用フレームワーク MiXiM1.1[7]を用いて実験を行う。シミュレーションフィールドは $400[m] \times 400[m]$ の2次元平面上に、東西南北に片道3車線の道路を $200[m]$ 間隔にそれぞれ3本、計6本を配置した格子状道路を利用する。右左折車両は車速 $20[km/h]$ で車間距離 $20[m]$ 、直進車両は車速 $60[km/h]$ 、車間距離 $40[m]$ で走行しているものとする。交差点の中心点から半径 $200[m]$ 以内を通信有効エリアとし、エリア内に存在するノードを中間兼受信ノードとする。通信有効エリア内に200台の車両が存在し、車両の進行方向は右左折、直進の3種類とする。配信データのサイズは $50[Kbyte]$ で $1[Kbyte]$ のシンボルに分割するものとする。送信ノード、中間ノードは $100[ms]$ 間隔毎にパケットを送信するものとする。パケットの有効期限は $10000[ms]$ として、有効期限が切れたパケットは破棄する。MAC層プロトコルは IEEE802.11b、通信帯域幅は $11[Mbps]$ とし、送信パケットは $10[mW]$ の送信電力で送信するものとした。提案手法を適用した通信は、シンボル数の集計間隔を $100, 400[ms]$ の2パターンで実験を行う。

表 1 受信パケット数のオーバーヘッド

	受信パケット数のオーバーヘッド (%)
LTNC	83.3
LTNC+提案手法(集計間隔 = 100ms)	52.3
LTNC+提案手法(集計間隔 = 400ms)	58.5

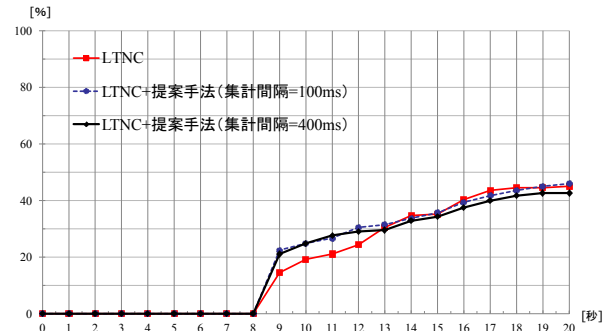


図 2 配信データの普及率

4.3. 実験結果

表 1 は受信パケット数のオーバーヘッドを示している。提案手法を適用した通信の中間ノードは近隣ノードが所持しているシンボルを推定し、それを基に再符号化するシンボルを決める。提案手法を適用していない通信と比較して、復号に利用できる innovative なパケットを短時間で受信することができると予想される。確かに、表 1 が示すように提案手法を適用した 2 種類の通信は、提案手法を適用していない通信よりも受信パケット数のオーバーヘッドが小さくなっている。提案手法を適用した 2 種類の通信において、集計間隔 $400[ms]$ のときよりも $100[ms]$ としたときの方が、オーバーヘッドが大きかったのは、集計間隔が短いと各集計時に利用する符号化ベクトルの数が減少し、近隣ノードが所持しているシンボル数を正確に推定できなかったからである。図 2 は、経過時間に対する配信データの普及率を示しており、3 種類の通信はそれぞれ同程度の普及率であること。以上ことから、提案手法を適用することで受信パケット数のオーバーヘッドを減らせることを確認した。

図 2 は差が出ていないが、これは配信データの普及率をすべてのシンボルを復号したノードの数を集計したためである。復号したシンボル数を集計することで、提案手法により innovative な符号化データを受信させられていることを確認できると考えられるが、今後の課題とする。

5. おわりに

中間ノードが送信先のノードの受信シンボルの状況を知るために、受信パケットの符号化ベクトルを利用し、近隣ノードの未受信シンボルを推定する手法を提案した。シミュレーション評価では受信パケット数のオーバーヘッドと配信データの普及率について、LTNCに提案手法を適用した通信と適用しない通信と比較した。提案手法は配信データの普及率を向上させることはできなかったが、受信パケット数のオーバーヘッドは提案手法により約 30% 少なくできることを示した。

参考文献

- [1] Chen, Y. and Schwan, K.: Opportunistic Overlays: Efficient Content Delivery in Mobile Ad Hoc Networks, Proc. the ACM/IFIP/USENIX 2005 International Conference on Middleware, p.354-374 (2005)
- [2] Dong, Q., Wu, J., Hu, et al.: Practical network coding in wireless networks, Proc. 13th Annu. ACM Int. Conf. Mobile Comput. Netw., pp.306-309 (2007).
- [3] Chou, A.P., Wu, Y. and Jain, K.: Practical network coding, Proc. 41st Annu. Allerton Conf. Communication, Control, and Computing, Monticello, IL, (2003).
- [4] Ho, T., Médard, M., Koetter, R., et al.: A Random Linear Network Coding Approach to Multicast, IEEE Transactions on Information Theory, vol.52, no.10, pp.4413-4430, (2006).
- [5] Champel, M.L., Huguenin, K, Kermarrec, M.A., et al.: LT network codes, Proc. IEEE 30th International Conference on Distributed Computing Systems (ICDCS 2010), pp.536-546 (2010).
- [6] OMNeT++, available from <http://www.omnetpp.org/>(accessed 2014-01-13).
- [7] MiXiM, available from <http://mixim.sourceforge.net/>(accessed 2014-01-13).