

Androidプラットフォーム上での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

林 直秀^{†1} 野口 拓^{†1} 川合 誠^{†1}

概要: アドホックネットワークでは、消費電力の低減や通信帯域の利用効率を高めるためにネットワーク内での不要なパケットの転送を削減することが重要である。本稿では、ネットワークコーディングを用いて複数のパケットを単一パケットに合成することにより、ネットワーク内に流れているパケットの数を削減する方式について実験による結果を評価する。本稿では、ネットワークコーディングを用いることでトラフィックを抑え、受信時間の減少や送信回数を低下させることができるか、Android 端末を使い実験を行って調査した結果、送信パケット数が増加した際、ネットワークコーディングを利用することによりフラディングよりもネットワーク内のトラフィックを低減できることを確認した。

Ad Hoc Broadcasting Using Network Coding on Android Platform

Abstract: Reducing unnecessary packet transmission is important to enhance the utilization efficiency of power and communication bandwidth in ad hoc networks. This paper evaluates the experimental results of the system reducing the number of packets flowing in the network by using network coding, which can encode multiple packets into a single packet. We have confirmed that network coding reduces more traffic in the network than flooding from the experimental results of transmission time and transmitted data amount in network coding using Android tablets.

1. はじめに

近年急速にスマートフォンやタブレットといった小型移動端末が普及し始め、これらの端末同士で無線アドホックネットワークの構築が可能になった。端末が自律分散的に動作するアドホックネットワークでは、電力や通信帯域の浪費を防ぐためにネットワークのトラフィックを削減することが重要である。

アドホックネットワークにおいて、情報をネットワーク上のすべての端末に送るブロードキャストを実現する方法として、フラディングがある。これは、各端末が受信したパケットをすべての隣接端末に転送する方法である。しかし、フラディングを用いた場合、ネットワーク内に冗長なパケットが氾濫し、パケット衝突の増加や電力の浪費といった Broadcast Storm 問題 [1] が発生する。このフラディングにおける冗長なパケットを削減する技術に、ネットワークコーディング [2][3] がある。これは、ネットワーク

内の中継端末でパケット同士の線形演算を行い単一パケットに合成する技術である。これにより、ネットワーク内の転送パケット数を削減することができ、パケット衝突を減らし、電力削減にもなる。また、送信データ量が大きくなればなるほど、冗長トラフィックの問題は深刻となる。そのため、アドホックネットワークにおけるネットワークコーディングを適用した様々な研究が行われている [4][5][6]。しかし、これらの研究の多くはシミュレーションによる評価にとどまっており、実機環境下での性能は明らかとなっていない。そこで本稿では、Android 端末を使って実応用を想定した実装実験を行い、ネットワークコーディングを用いたブロードキャスト通信の品質評価を行う。

2. ネットワークコーディング

ネットワークコーディングは、ネットワーク内での符号化技術である。図 1 に示すように、中継ノードで受信された複数のパケットを符号化し単一パケットに変換して転送することにより、パケット転送効率を高めている。ここでは、中継ノードが受信した n 個の異なるパケット p_1, \dots, p_n

^{†1} 現在、立命館大学大学院 情報理工学研究科
Presently with Graduate School of Science and Engineering, Ritsumeikan University

の符号化を行い、生成されたパケット p_{out} を送信している。

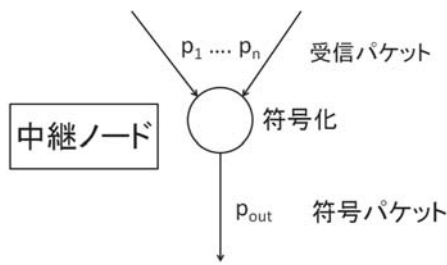


図 1 ネットワークコーディングの符号化

2.1 パケットの演算方法

本稿では、ネットワークコーディングの符号化方式の中でも、線形符号を用いる線形ネットワークコーディング [7] を用いる。送信ノードで生成・送信されるパケットを情報パケット、中継ノードにおいて符号化によって生成されるパケットを符号パケットとする。線形ネットワークコーディングでは、パケットをガロア体の要素とみなし、線形演算により符号化が行われる。ブロードキャストを用いる場合は、図 2-(a) のようにノード S が送信ノード、ノード N_1, \dots, N_5 が受信ノードである場合、各受信ノードにおいて情報パケット p_1, p_2 を符号化せずにブロードキャストをするが、ネットワークコーディングを用いる場合は、図 2-(b) のようにノード N_1, N_2, N_3 において符号化が行われる。例えば、 N_3 で生成される符号パケット p_{out3} は符号化係数 e_5, e_6 を用いて、 $p_{out3} = e_5 p_1 + e_6 p_2$ のように表せる。ただし、 e_5, e_6 はガロア体の要素である。 N_4, N_5 は、受信した符号パケット ($p_{out3}, p_{out1}/p_{out2}$) から p_1, p_2 についての連立方程式を構成し、それを解くことにより復号し、 p_1, p_2 を得ることができる。図 2 の例では、 p_1, p_2 をすべてのノードへ送信する場合、ブロードキャストでは総転送回数は 16 となるが、ネットワークコーディングでは 10 となり、パケット転送効率が高いことが分かる。

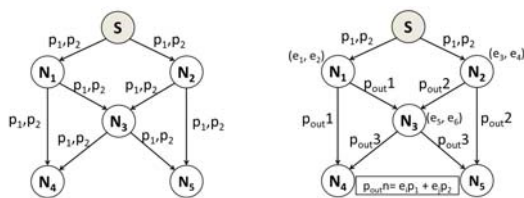


図 2 ブロードキャストとネットワークコーディングの比較

一般に、復号済みの受信パケット (情報パケット) を $p_i (i = 1, 2, \dots, N)$, 符号化ベクトルを $e = (e_1, e_2, \dots, e_N)$, N を符号化数, 符号パケットを c とすると, 符号パケットの生成は式 (1) のような演算で行われる。

$$c = \begin{pmatrix} e_1 & e_2 & \cdots & e_N \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{pmatrix} \quad (1)$$

ここで、符号化係数 $e_i (i = 1, 2, \dots, N)$ はガロア体の要素であり、演算はガロア体上で行われる。式 (1) は式 (2) でも表すことができる。

$$c = \sum_{i=1}^N e_i p_i \quad (2)$$

受信ノードで復号を行うには、符号化に用いられた符号化数 N 以上の符号パケットが必要になる。また、 N 個の符号パケットの符号化ベクトルは互いに線形独立である必要がある。

復号された N 個のパケットを $P = (p_1, p_2, \dots, p_N)$, 符号パケット $c_i (i = 1, 2, \dots, N)$ の符号化に使用された符号化係数を $e_{i,j} (j = 1, 2, \dots, N)$ とすると、復号は式 (3) で表すことができる。

$$P = \begin{pmatrix} e_{1,1} & \cdots & e_{1,N} \\ \vdots & \ddots & \vdots \\ e_{N,1} & \cdots & e_{N,N} \end{pmatrix}^{-1} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} \quad (3)$$

ここで、符号化ベクトルを $e_i = (e_{i,1}, \dots, e_{i,N})$, $E = (e_1, e_2, \dots, e_N)$ とすると、復号は式 (4) でも表すことができる。

$$P = E^{-1}C \quad (4)$$

式 (4) より、符号パケットが復号可能となる条件は、 E が逆行列を持つ場合である。

2.2 ランダムネットワークコーディング

ランダムネットワークコーディング [8] とは、各ノードで符号化ベクトルの要素となる符号化係数 $e_i (i = 1, 2, \dots, N)$ をガロア体 $GF(2^m)$ よりランダムに選択して、符号化を行う方法である。ただし、すべての e_i が 0 となる符号化ベクトルは用いない。符号化数が N の場合、符号パケット c は符号化係数 e_i を用いて式 (2) で表される。受信ノードにおいて、すべての符号パケットを復号するには、少なくとも N 個の符号パケットが必要となる。復号に成功すれば図 3-(a) のように元のパケットを復号し、再び符号化して符号パケットを生成する。図 3-(b) のように符号パケット数が符号化数に満たない場合や、図 3-(c) のように受信した N 個の符号パケットの幾つかが線形従属となる場合は式 (4) における E の逆行列の生成が不可能となるため復号

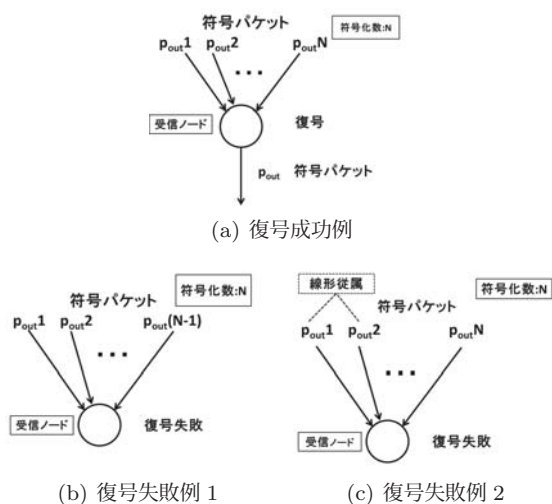


図 3 受信ノードにおける復号例

することができない。

一般的なネットワークコーディングでは、各受信ノードで復号可能となるように事前に符号化ベクトルを割り当てておく。この場合、トポロジが変化すれば、それに従って符号化ベクトルの割り当てをやり直す必要がある。

それに対しランダムネットワークコーディングの場合、トポロジの変化に応じた符号化ベクトルの再割り当てが必要ない。復号する際に、符号化ベクトルが線形従属となり復号に失敗する可能性は存在するが、ガロア体のサイズを大きくすることで、符号化ベクトルが線形従属になる確率を小さくすることができる。しかし、ガロア体のサイズが大きくなれば、符号パケット中の符号化ベクトルを格納するフィールドが大きくなってしまいう問題があり、ネットワークの大きさに応じて適切なガロア体サイズを設定することが重要となる。

3. 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

ネットワークコーディングは本来マルチキャストネットワークにおいて、最大フローのスループットで情報伝送を可能とする技術である。しかし、本稿では、スループットを向上させる効果よりも複数のパケットを集約して単一パケットとして送信する結果、ネットワークコーディングがネットワーク内部で転送されるパケット数を削減するという性質に主眼を置いて無線アドホックネットワークでの適用を考える。

3.1 配送アルゴリズム

3.1.1 概要

本稿では、アドホックネットワークにおけるブロードキャスト配信方式として、ネットワークコーディングを用いたマルチキャストプロトコルである CodeCast[9]をブロードキャスト化したものを用いる。CodeCast は、符号

化方式にランダムネットワークコーディングを使用し、さらに、再送機能も備えていることから、実機実装に適したプロトコルである。CodeCast は、ネットワーク層でのプロトコルであり、CodeCast Agent によって符号化・復号を行っている。CodeCast では、ネットワーク内に存在するノードは、送信ノード、中継ノード、受信ノードの3種類に分けられる。送信ノードはマルチキャストの送信元であり、受信ノードはそのマルチキャストメンバである。中継ノードはマルチキャストメンバではないが、パケットの中継を行うノードである。本稿では、CodeCast をブロードキャスト化するため、送信ノード以外のすべてのノードを受信ノードとし、受信ノードは中継ノードの役割も兼ねることとする。また、符号化に使用した符号化ベクトルは、パケットのペイロードに格納して送信する。

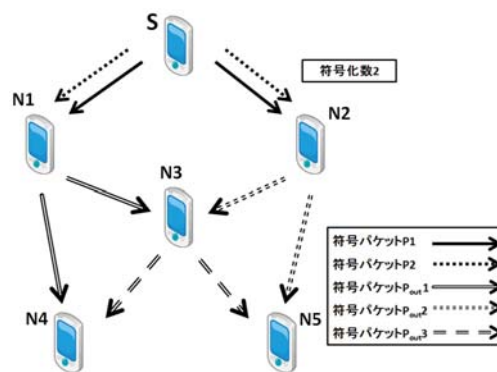


図 4 提案方式の構成例

本稿で用いるブロードキャスト配信方式（以下、提案方式）の構成例を図 4 に示す。ノード S は送信ノードであり、2 種類の符号パケット送信する（符号化数 $N = 2$ ）。受信ノード群（図 4 におけるノード N1~N5）は、中継ノードも兼ねているため、2 種類の符号パケットを受信後、復号し隣接ノードに再び符号パケットを転送する。例えば、図 4 において、ノード N3 はノード N1 および N2 からそれぞれ符号パケット P_{out1}, P_{out2} を受信後、復号および再符号化を行い符号パケット P_{out3} を転送する。3.1.2, 3.1.3 節にて、本方式における各ノードの振る舞いを示す。3.2 節にて、パケットの仕様について示す。

3.1.2 送信ノード

送信ノードは、送信するデータであるアプリケーションフレームをブロック単位に分割する。符号化数を N とすると、線形演算を用いて N 個のブロックから符号パケットを生成する。符号パケットに付加されるラベル ($blockid$) は、符号パケットがどのブロックに属しているかを表すとともに、シーケンス番号の役割も兼ねている。ラベルは、ブロック毎に符号化数分増加する。符号パケットを $f_{(blockid, N)}$ 、符号化係数を $e_i (i = 1, 2, \dots, N)$ 、アプリケーションブロックを $p_n (n: アプリケーションブロック番号)$ とすると、符号

化は次の式で表すことができる。

$$f_{(blockid,N)} = \sum_{i=1}^N e_i p_{(i-1+blockid)} \quad (5)$$

生成された符号パケットのヘッダには、符号化に使用された符号化ベクトル $e = (e_1, e_2, \dots, e_N)$ が格納される。送信ノードは、 N 個の符号パケットを生成し、隣接ノードに送信する。

ここで、符号化数が2のときの送信ノードにおける処理の例を図5に示す。アプリケーションフレームを p_1, \dots, p_N の4つのブロック（情報パケット）に分割したとする。符号化数は2であるので、 p_1, p_2 から2つの符号パケット $f_{(1,2)}, f'_{(1,2)}$ を生成する。2つの符号パケットを生成する際に使用した符号化ベクトル e は、互いに線形独立であれば、それぞれ値は異なる。符号パケット $f_{(1,2)}, f'_{(1,2)}$ のヘッダにそれぞれ使用した符号化ベクトルを格納し、送信する。 p_3, p_4 においても同様の処理をし、 $f_{(3,2)}, f'_{(3,2)}$ を生成、送信する。

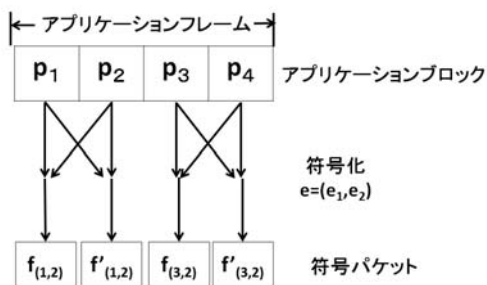


図5 送信ノードの処理の例

3.1.3 受信（中継）ノード

符号パケットを受信した中継ノードは、復号を行うために同じラベルを持った N 個の線形独立な符号パケットが揃うまで受信パケットをバッファに格納する。もし、バッファ内のパケットと線形従属な符号パケットを受信した場合は、その符号パケットを破棄する。この時、新しいラベルを持った符号パケットを受信した場合、そのラベルに対しタイマーがセットされる。このタイマーは、同一ラベルを持つ線形独立な符号パケットを受信する度にリセットされ、タイムアウトになった時点でバッファ内に N 個以上の符号パケットが存在した場合は復号を行い、 N 個のアプリケーションブロック（情報パケット）を上位層に引き渡すとともに、再び符号化し符号パケットを生成し、転送する。以後、既に送信した符号パケットと同じラベルを持つパケットは受信しても破棄する。また、タイムアウトになった時点で、バッファ内の符号パケットが N 個未満だった場合、復号未完了であるラベルを格納した再送要求を全隣接ノードに向けてブロードキャストする。再送要求を受信した隣接ノードは、自身のバッファに要求ラベルのパケットが存在すれば、要求元のノードに再送パケットと

して符号パケットを送信する。

3.2 パケット仕様

提案方式では、UDP/IP を用いて図6に示す形式の3種類のパケットを使用する。

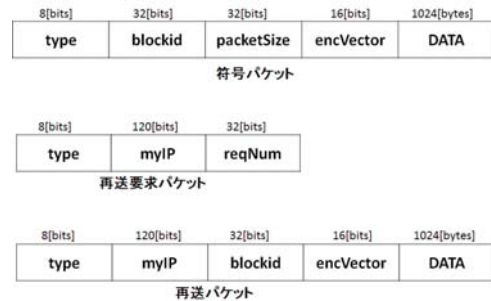


図6 パケット仕様

再送要求パケットは、受信ノードにおいて復号に必要な符号パケットが集まらなかった時に隣接ノードにブロードキャストされる。再送パケットは、再送要求パケット受信後、要求元への返送として使用される。各フィールドの概要を表1に示す。

表1 パケットフィールド仕様

フィールド名	説明
type	パケット識別子
blockid	ラベル
packetSize	パケット数
encVector	符号化ベクトル
DATA	送信するデータ
myIP	送信元 IP アドレス
reqNum	再送要求番号

4. 性能評価

4.1 実験環境

本性能評価には同機種 of Android 端末を使用した。Android 端末の仕様は表2の通りである。

表2 端末仕様

製品名/製品型番	ICONIA TAB A500-10S16
OS	Android 3.2
CPU	Dual Core ARM Cortex A9 CPU(1GHz)
Memory	1GB DDR2
通信モジュール	IEEE802.11b/g/n

実験では、フラッディングと提案方式において、以下の2つのデータが変化した場合の総受信データ量、遅延を評価した。

- 送信する画像データの容量
- 再送要求の TIME OUT

総受信データ量は、各ノードが受信したパケットのデータ総量であり、遅延は各ノードが、最初のパケットを受信してからすべての符号パケットを復号し、欠けているパケットが存在する場合は再送要求を行い、オリジナルデータ(画像)を復元できるまでの時間をノード数で割ったものである。

ノードは6台、送信ノードは1台で、送信ノード以外のすべてのノードは中継ノードであり受信ノードである。トポロジは図7の通りである。実験では、実験スペースの制約上、各ノードを物理的に距離を離して配置することが困難であったため、隣接ノード以外からの受信パケットはフィルタリングすることで擬似的にこのトポロジを構成している。

また、線形ネットワークコーディングにおいては本来、線形演算をガロア体上で行うが、本性能評価では実装を容易にするため実数体上での線形演算を行っている。実数体を用いる場合、パケットのデータ容量が増加し、線形演算時にガロア体を用いた場合に比べ負荷は増加する。このため、本性能評価は、最悪条件での評価となっている。

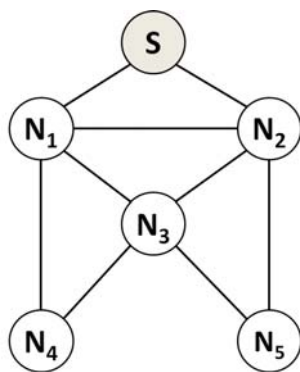


図7 トポロジ

4.2 実験結果

実験では、フラッディングと符号化数2の提案方式を比較している。それぞれ、グラフ上では Network Coding, Flooding と表記している。送信データ容量変化の実験では、再送要求の TIME OUT 値を 1500 ミリ秒で統一している。

送信する画像のデータ容量を変化させた場合の送信ノードの送信時間特性を図8に示す。送信時間は、送信ノードがデータを送信する送信開始時刻から送信終了時刻までの時間のことである。

データ容量が 100KBytes の時、ネットワークコーディングおよびフラッディングの送信時間の差は小さい。再送要求の TIME OUT 値変化の実験では、送信時間の差による影響を抑えるためデータ容量を 100KBytes で統一する。

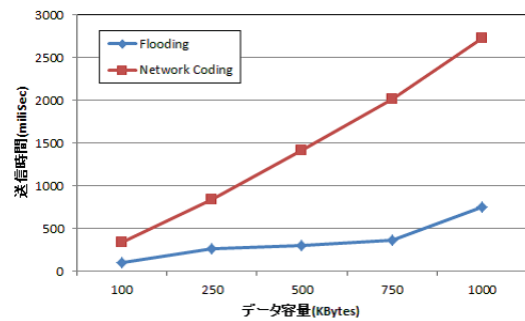


図8 送信ノードの送信時間

4.2.1 総受信データ量

図9はデータ容量を変化させた場合の、図10は再送要求 TIME OUT 値を変化させた場合の総受信データ量特性を示している。

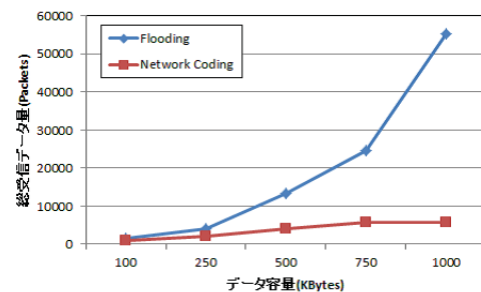


図9 送信データ容量変化による総受信データ量

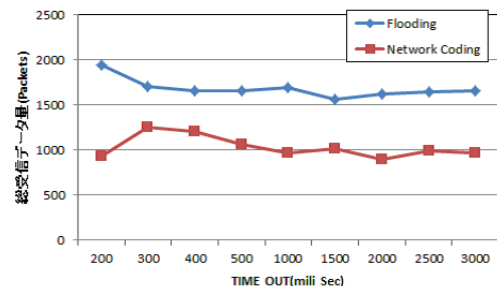


図10 TIME OUT 変化による総受信データ量

データ容量が変化する場合、再送要求 TIME OUT 値が変化する場合いづれにおいても、ネットワークコーディングの総受信データ量がフラッディングと比較して減少している。これは、ネットワーク内の中継ノードにおいて、符号化数分のアプリケーションブロックをそのまま送信する代わりに、ネットワークコーディングによって単一の符号パケットを生成し送信することで、各ノードの転送回数が削減されたためである。

図9において、データ容量の増加に伴い、フラッディングでは総受信データ量が大きく増加した。これは、フラッディングでは、各ノードが受信したパケットを全隣接ノードにブロードキャストし、大量の冗長パケットがネットワーク内に氾濫するためである。隣接ノードにブロードキャストする点はネットワークコーディングも同一である

が、ネットワークコーディングでは単一の符号パケットを転送するため、転送パケット数がフラッディングよりも減り、総受信データ量が抑えられる。

再送要求の TIME OUT 値の増加による総受信データの影響は、フラッディング、ネットワークコーディングともに少なかった。フラッディングとネットワークコーディングの総受信データ量の差は、おもに転送パケット数の違いからくるものである。したがって、ネットワークコーディングは、データ容量、つまり送信パケット数が大きいほどトラヒック低減効果が顕著に表れる。

4.2.2 遅延

図 11 はデータ容量を変化させた場合の、図 12 は再送要求 TIME OUT 値を変化させた場合の遅延特性を示している。

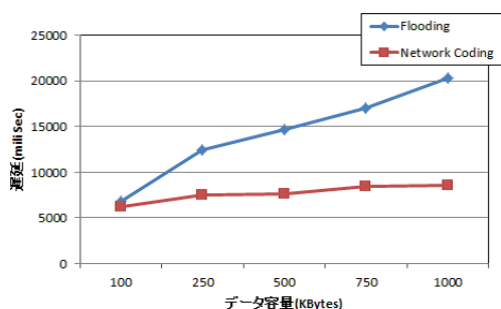


図 11 送信データ容量変化による遅延

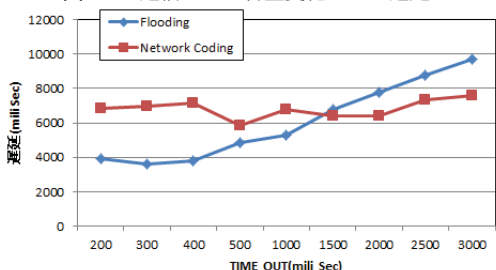


図 12 TIME OUT 変化による遅延

データ容量が変化する場合、フラッディング、ネットワークコーディングともにデータ容量の増加に比例して遅延が増加する。フラッディングに比べ、ネットワークコーディングの遅延の増加率が小さいのは、転送パケット回数、再送要求数がフラッディングより少ないためである。

再送要求 TIME OUT 値が変化する場合、ネットワークコーディングでは、遅延が 6000 から 8000 ミリ秒の間に収まり、あまり変化が見られない。一方、フラッディングでは、TIME OUT 値の増加に伴い遅延も増加する。本評価では、送信データ容量を 100KBytes としたため、ネットワーク内の負荷が小さくパケット到達率は非常に高い状況にある。このような状況下では、各ノードにおいて受信処理は完了しているにもかかわらず、TIME OUT を待つという無駄な空き時間が生じている。このため、TIME OUT 値が増加すれば、この空き時間も増加し、結果として遅延

も悪化する。フラッディングの場合のみに遅延の増加が見られる理由は、TIME OUT 値の増加による再送要求や再送パケットの減少よりも空き時間の増加の方が大きくなったためである。

5. まとめ

本稿では、Android 端末で構成されたアドホックネットワークにおいて、ネットワークコーディングを用いたブロードキャスト配信方式を実装し、性能評価を行った。実装実験より、ネットワークコーディングを用いることでトラヒックを削減し、また遅延の減少が可能なることを明らかにした。本実験では、ネットワークサイズ、トポロジも限定されたものであったため、今後多様なネットワーク条件下で実験を行い、実用面での性能を明らかにする必要がある。また、トラヒックのさらなる削減を図るため、符号化アルゴリズムおよび再送制御の改良を行う予定である。

参考文献

- [1] S. -Y. Ni, Y. -C. Tseng, Y. -S. Chen, and J. -P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," ACM Mobilcom 99, pp. 151-162, Aug. 1999.
- [2] R. Ahlswede, N. Cai, S. -Y. Li, and R. Yeung, "Network Information Flow," IEEE Transactions on Information Theory, Vol. 46, No. 4, pp. 1204-1216, July 2000.
- [3] P. Vingelmann, F. H. P. Fitzek, M. V. Pedersen, and J. Heide, "Synchronized Multimedia Streaming on the iPhone Platform with Network Coding," IEEE Communications Magazine, pp. 126-132, June 2011.
- [4] Takahiro Matsuda, Taku Noguchi, and Tetsuya Takine, "Broadcasting with Randomized Network Coding in Dense Wireless Ad Hoc Networks," IEICE Transactions on Communications, vol. E91-B, No. 10, pp. 3216-3225, October 2008.
- [5] 星野 豊, 野口 拓, 川合 誠, "適応型ネットワークコーディングを用いたアドホックマルチキャストネットワーク," 第 72 回情報処理学会全国大会, Mar. 2010
- [6] Y. E. Sagduyu, and A. Ephremides, "On Joint MAC and Network Coding in Wireless Ad Hoc Networks," IEEE Transactions on Information Theory, Vol. 53, No. 10, pp. 3697-3713, Oct. 2007.
- [7] S. -Y. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," IEEE Transactions on Information Theory, Vol. 29, No.2, pp. 371-378, Feb. 2003.
- [8] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On Randomized Network Coding," The 41st Annual Allerton Conference on Communication, Control, and Computing, Oct. 2003.
- [9] J. -S. Park, M. Gerla, D. S. Lun, Y. Yi, and M. Medard, "CodeCast: A Network-Coding-Based Ad Hoc Multicast Protocol," IEEE Wireless Communications, pp. 76-81, Oct. 2006.