

## UDP ブロードキャストを用いた 高信頼性データ通信プロトコルの実装と評価

田中裕大<sup>†1</sup> 荒川 豊<sup>†2</sup>  
田頭茂明<sup>†2</sup> 福田 晃<sup>†2</sup>

我々はこれまで、近距離環境において複数人で利用可能なファイル共有手法であるローカルクラウドストレージと、ローカルクラウドストレージ上で動作するファイル転送プロトコルであるダイレクトネイバーキャストを提案している。ダイレクトネイバーキャストは近距離環境にいる複数の相手との通信時に、UDP ブロードキャストとアプリケーションレイヤでの再送を組み合わせることにより、高速かつ信頼な 1 対多通信を実現している。本稿では、ダイレクトネイバーキャストを実装し、その信頼性と通信速度について様々な環境で評価を行った結果を報告する。

### Implementation and Estimation of UDP Broadcast based Reliable Data Transfer Protocol

YUTA TANAKA,<sup>†1</sup> YUTAKA ARAKAWA,<sup>†2</sup>  
TAGASHIRA SHIGEAKI<sup>†2</sup> and AKIRA FUKUDA<sup>†2</sup>

We have proposed “Local Cloud Storage” which is extemporarily constructed through short-range wireless devices, and “Direct Neighbor Cast” which realizes a high-speed and high-reliability data transmission by using UDP broadcast and application-layer retransmission. In this paper, we design a file sharing protocol for such short-range wireless communication and evaluate its reliability and communication speed.

<sup>†1</sup> 九州大学システム情報科学府

Kyushu University Graduate School of Information Science and Electrical Engineering

<sup>†2</sup> 九州大学システム情報科学研究院

Kyushu University Faculty of Information Science and Electrical Engineering

### 1. はじめに

近年、ネットブックやタブレット端末の台頭により、コンピュータを持ち運ぶ機会が増え、屋外や喫茶店など様々な場所でコンピュータ端末を操作することが一般的になってきた。これに伴い、ユーザ同士が近距離にいる状況で、複数のユーザがファイルの受け渡しやファイルの共有を行う機会が増えている。また、CO<sub>2</sub> 排出量削減のため、今まで紙に印刷して配布していた資料をデジタルデータのまま配布する需要も増えてきていると考えられる。

このような状況では、メールにファイルを添付して送信する方法や、USB メモリを使用してファイルを渡す方法を用いることが多い。しかし、メールを用いた方法は、全ての端末がインターネットに繋がっている状況でないと使用できない。また、USB メモリを使用した方法は、人数が多い場合や頻りにファイルがアップデートされる場合に USB メモリの受け渡しが多くなり煩雑である。

我々は、このような状況でのファイル共有手法として、ローカルクラウドストレージ (LCS:Local Cloud Storage) を提案している<sup>1)</sup>。この手法では、ローカルクラウドクライアントと呼ばれるデバイスと共有したい複数の端末にそれぞれ接続する。全てのデバイスはあらかじめペアリングされており、内蔵された無線モジュールを用いて無線ネットワーク上でアドホックに繋がっている<sup>2)</sup>。また、ローカルクラウドクライアントには記憶領域が内蔵されており、端末に接続すると OS 上では 1 つのドライブとして認識されると共に、ネットワークで繋がった全てのデバイスで共有される。そのため、ユーザは OS にマウントされた記憶領域に書き込むだけで、ファイル共有が可能となる。

ローカルクラウドストレージでは、複数個の分散した記憶領域を高速に同期する必要がある。通常、ファイル転送のような信頼性が重視される通信には TCP によるユニキャスト通信が一般的である。しかしながら、1 対 1 で順次ファイルをコピーしていく TCP は、共有相手が増加すると同期時間が比例して増加するという問題がある。ローカルクラウドストレージで行われる通信は、1 台の端末が編集したファイルを他の全ての端末に転送するため、1 対多の通信が主となる。一方、UDP ブロードキャストは、同一セグメント上にいる複数の相手に同時にパケットを送信することができるため、今回の想定環境で利用できると考えられる。ただし、UDP 通信であることから信頼性が低く、そのままファイル転送に用いることはできない。

そこで、我々は、UDP ブロードキャストにアプリケーションレイヤで信頼性を確保した、ダイレクトネイバーキャスト (DNC:Direct Neighbor Cast) を提案している。アプリケー

ションレイヤにおいて信頼性を確保する方法として、TCP でも用いられている ACK を用いることが考えられる。ACK とは ACKnowledgement の略であり、受信端末がパケットを正常に受信できた際に送信端末にパケットを送り知らせる手法である<sup>3)</sup>。しかし、UDP ブロードキャスト上で ACK を用いると、ACK パケットでネットワークがあふれてしまう可能性が高い。そこで、ダイレクトネイバーキャストでは NACK を用いて信頼性を確保する。NACK とは Negative ACKnowledgement の略であり、受信端末がパケットを正常に受信出来なかったことを検知した際に送信端末にパケットを送り知らせ、受け取った送信端末が該当パケットを再送する手法である。

文献<sup>1)</sup> では、ダイレクトネイバーキャストを提案し、概要を説明したが、その詳細なプロトコルおよび性能に関する評価は行っていなかった。そこで、本稿では、ダイレクトネイバーキャストの詳細な設計と実装を行い、さまざまな環境におけるダイレクトネイバーキャストの性能を測定した。その結果、ダイレクトネイバーキャストはネットワーク負荷が高い状況でも 100% のパケットを送信することが出来ることを確認した。また、受信端末が 4 台の場合でも受信端末が 1 台の場合と比較して転送時間は 121% 程度に抑えられており、TCP ユニキャストと比較すると半分以下の時間で転送出来ることを確認できた。

本稿の構成を以下に示す。第 2 章では、想定環境である近距離無線通信の定義およびローカルクラウドストレージについて説明する。第 3 章では、ダイレクトネイバーキャストの概要と実装について述べる。第 4 章では、ダイレクトネイバーキャストの信頼性および通信速度について評価する。第 5 章では、本手法と関連のある研究として、Reliable Multicast について説明する。最後に第 6 章では、まとめと今後の課題について述べる。

## 2. ローカルクラウドストレージ

我々は、複数の PC 間で、インフラを用いずに即時的なファイル共有を実現する仕組みとして、近距離通信を用いたローカルクラウドストレージを提案している<sup>1)</sup>。本章では、近距離無線通信の定義とローカルクラウドストレージについて説明する。

### 2.1 想定環境

近距離通信とは、様々な定義があり、NFC(Near Field Communication) などによる数センチ程度の範囲による通信も含まれるが、今回は、十数メートル以内にいる数人から十人程度のユーザが同じテーブルのまわりや同じ部屋の中などで行う通信と定義する。例として、取引先の企業を訪れた営業者が、相手企業でプレゼンテーションを行う際、プレゼンテーションで使用する資料を参加者に配布したい状況が考えられる。この場合、現在は資料

を紙で印刷し、後日電子データをメールで送信する機会が多い。他の例として、街中にある喫茶店で友人同士がデジタルカメラで撮影した画像を交換するという状況が考えられる。この例でも、ユーザはインターネットに接続できない場合があり、既存の通信インフラに頼らずに通信を行う手段が必要になってくる。また、上に挙げた 2 つの例を含め、このような通信はその場限りの一時的なものが多く、事前に相手の端末の識別子などを交換したり、前回通信した時の情報を使用することは困難な場合が多い。

### 2.2 ローカルクラウドストレージ

これまで我々は、想定環境のような近距離にいる相手と手軽にファイルを交換する仕組みとして、近距離無線通信を用いたローカルクラウドストレージを提案している<sup>1)</sup>。ローカルクラウドストレージの概要を図 1 に示す。この手法では、ローカルクラウドクライアントと呼ばれるデバイスを複数個使用する。ローカルクラウドクライアントには無線モジュールと記憶領域を内蔵している。使用するローカルクラウドクライアント同士はあらかじめペアリングされており、ローカルクラウドクライアントが無線ネットワーク上でインフラなしに接続されている<sup>2)</sup>。端末の OS からはローカルクラウドクライアントは USB メモリのような記憶領域に見えており、ユーザは特にネットワークを意識することなく記憶領域に書き込むだけでファイル共有が可能となる。

文献<sup>1)</sup> では、ローカルクラウドストレージの概念およびその環境に適した通信方式として、UDP ブロードキャストを用いた高信頼プロトコルであるダイレクトネイバーキャストの概念の提案を行った。本稿では、ダイレクトネイバーキャストの詳細と性能評価について報告する。

## 3. ダイレクトネイバーキャスト

ダイレクトネイバーキャストとは、ローカルクラウドストレージのような 1 対多のファイル共有に適したプロトコルであり、同時接続端末数に依存せず高速かつ高信頼なデータ転送が可能であるという特徴を持つ。

### • 信頼性の確保

下記の送信シーケンスと受信シーケンスにおいて用いられる、信頼性を確保する手法について説明する。

### • 送信シーケンス

送信端末はユーザ指定のファイルをダイレクトネイバーキャストを用いて受信端末への送信を行う。

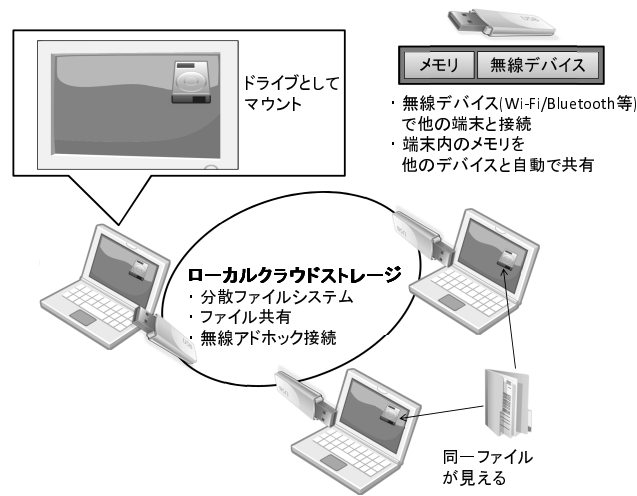


図1 ローカルクラウドストレージ 概要

### ● 受信シーケンス

受信端末では、送信端末からダイレクトネイバーキャストを用いて送信されたファイルを受信し、ファイルとして書き出す。

#### 3.1 信頼性の確保

UDP は、TCP とは異なり信頼性の保証を行っていない。信頼性のない UDP ブロードキャスト通信をファイル共有で用いるために、ダイレクトネイバーキャストでは NACK を用いてアプリケーション層で信頼性を確保している。NACK とは Negative Acknowledgment の略で、受信側がパケット受信に失敗した場合に送信側に再送要求を送り、送信側が再送を行う手法である。NACK は、ACK を用いる場合と比較して、エラーが比較的小さい場合に高速に動作するという特徴がある。ダイレクトネイバーキャストでファイルを送信する場合、各パケットに一意で連続したシーケンス番号を割り振る。受信端末は受信したパケットのシーケンス番号を見ることで、受信できていないパケットを知ることができる。受信端末は再送要求として受信できなかったパケットのリストを送信する。(図 2)

また、NACK パケット自体が届かなかった場合や、再送パケットの受信に再び失敗してしまう可能性もあるため、送受信のシーケンスではどのパケットがロスしても最終的にファ

イルのすべてのパーツが相手に届くように実装しなければならない。

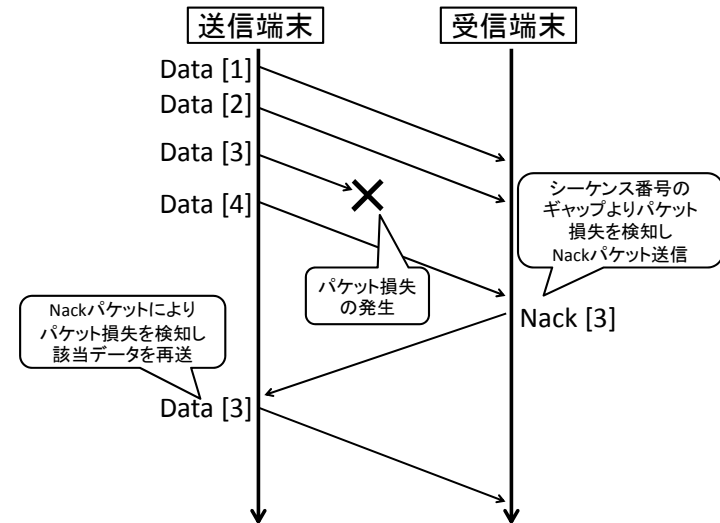


図2 NACKによるパケット損失の検知

#### 3.2 送信シーケンス

送信の流れを図3に記す。送信端末では、まず該当ファイルを一定の大きさに分割し、それぞれにシーケンス番号を付与する。これは、通常UDPではシーケンス番号を付与しないため、アプリケーションでシーケンス番号を確保する必要があるためである。1つのシーケンス番号は1パケットに対応する。

その後全てのパケットを送信リストに追加し、パケット送信処理に入る。送信リストにパケットがある場合、そのうちの一つを送信し、送信リストから削除する。すべてのパケットを送信し終えた場合、一定時間おきに FINISH パケットを送信する。

また、受信端末からの NACK パケットを受け取った場合、送信リストに該当パケットを加える。送信していないパケットがなくなってから一定時間経過しても NACK パケットを受信しなかった場合、全ての受信端末が受信を終了したと判断し、送信シーケンスを終了

する。

### 3.3 受信シーケンス

受信の流れを図4に記す。受信シーケンスは開始されるとパケット受信待ち状態に入る。パケットを受信するとそのパケットの種類を確認する。この後の動作は受信したパケットの種類に応じて以下の3種類に分かれる。

- DATA パケット  
受信したパケットがDATAパケットの場合、該当パケットを過去に受け取ったかを確認する。過去に受け取ったことがあるパケットの場合、何もせずにパケット受信待ち状態に戻る。始めて受信したパケットの場合、そのパケットを記憶しパケット受信待ち状態に戻る。
- FINISH パケット  
受信したパケットがFINISHパケットの場合、シーケンス番号を基にすべてのパケットを受信したか確認を行う。受信できていないパケットが存在した場合、該当パケットが記されたNACKパケットを送信する。すべてのパケットを受信している場合、パケットからファイルを作成し、シーケンスをを終了する。
- NACK パケット  
受信したパケットがNACKパケットの場合、何もせずにパケット受信待ち状態に戻る。以上をすべてのシーケンス番号のファイルを受信するまで続ける。全てのシーケンス番号のファイルの受信に成功するとそれらのパケットからファイルを作成し終了する。

## 4. 評価

本章では、ダイレクトネイバーキャストの有効性を明らかにするために、信頼性および転送速度の観点から評価を行う。信頼性に関する評価では、様々な環境においてダイレクトネイバーキャストを動作させ、全ての環境において受信端末にファイルが転送されたことを確認する。転送速度に関する評価では、TCPユニキャストと比較した場合の転送にかかる時間について評価する。

### 4.1 信頼性に関する評価

#### 4.1.1 評価実験環境

実験は以下の3つの環境で行った。

- 低負荷環境

ダイレクトネイバーキャスト以外の通信がほとんど発生していない環境を低負荷環境と

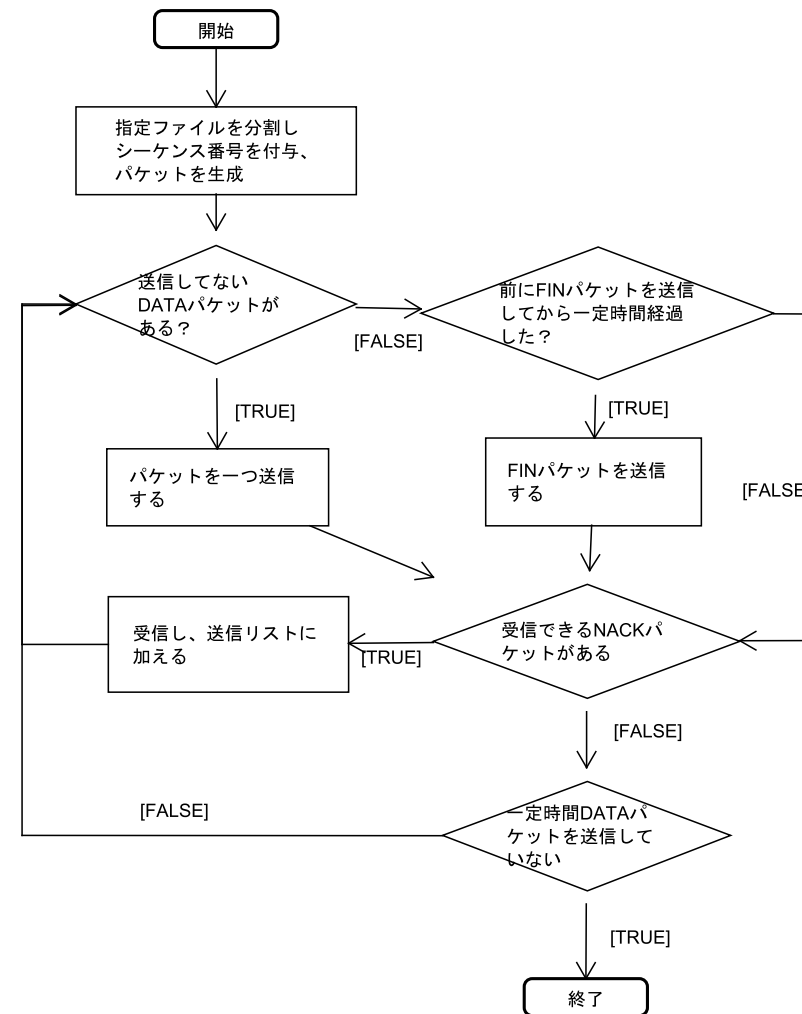


図3 送信シーケンス

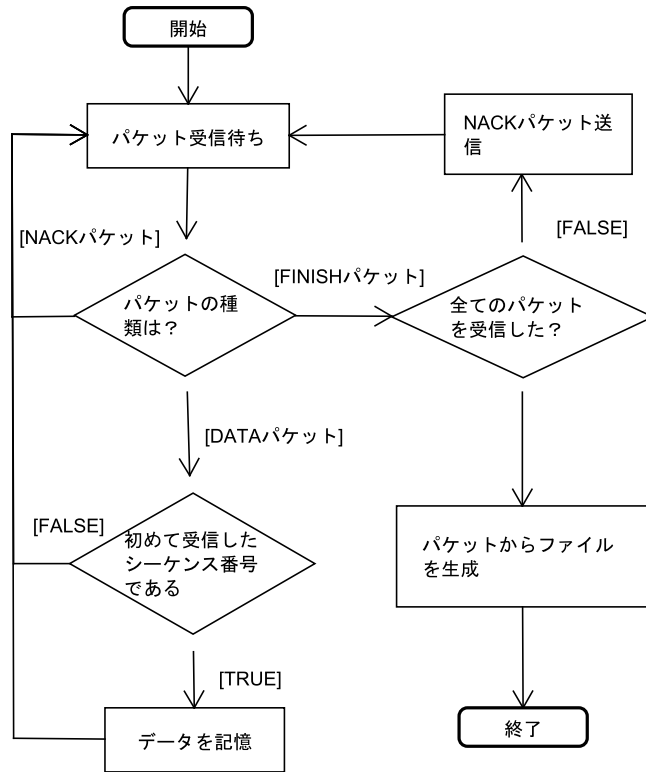


図 4 受信シーケンス

定義する．今回の測定では，使用チャンネルおよび使用チャンネルと干渉する可能性のあるチャンネルにはアクセスポイントが見えていない環境を使用した．

- 中負荷環境  
 会社のオフィスのようにダイレクトネイバーキャスト以外の通信がある程度の量発生している環境を中負荷環境と定義する．今回の測定では，測定に使用したチャンネルと同じチャンネルに 2 つのアクセスポイントが見えている環境を使用した．
- 高負荷環境  
 意図的にダイレクトネイバーキャスト以外の通信を発生させた環境を高負荷環境と定義する．今回は，2 台の端末が妨害パケットとして，1000times/sec・1000Bytes/packet の ping を発生させている環境とする．  
 また，各環境に共通する項目を表 1 に記す．

表 1 評価実験 ネットワーク環境

通信方式	IEEE802.1g
チャンネル	1
暗号化方式	なし

#### 4.1.2 評価実験方法

各環境において，1 台の送信端末が 1 台の受信端末に 1MiB のファイルを 1000 バイトごとに分割しシーケンス番号を付与しダイレクトネイバーキャストで送信する．その時，各パケットが受信されるまでの平均送信回数と，その最大送信回数を記録した．以上の実験を 10 回行い，各平均送信回数の平均値と各最大送信回数の最大値を結果とした．

#### 4.1.3 評価実験結果

各負荷毎の送信回数の割合を表 2 に示す．負荷が高くなることで，平均送信回数・最大送信回数共に増加していることが分かる．

次に，表 2 に，100%の転送に要する送信回数の分布を示す．低負荷環境および中負荷環境の場合は 9 割以上のパケットを 1 回の送信で受信できている．また，高負荷環境においても 2 回の送信で 9 割以上のデータ転送に成功していることが分かる．そして，いずれの場合も最終的に 100%のパケットの転送に成功している．

このことから，ダイレクトネイバーキャストは 100%のパケットを受信端末に届ける信頼性を持っているといえる．

表 2 負荷毎の平均送信回数と最大送信回数

	低負荷環境	中負荷環境	高負荷環境
平均送信回数 (回)	1.097	1.109	1.365
最大送信回数 (回)	6	9	17

表 3 負荷毎の各パケットの送信回数の割合

	低負荷環境 (%)	中負荷環境 (%)	高負荷環境 (%)
1 回	92.3	93.4	76.4
2 回	6.4	4.1	19.2
3 回	1.1	0.6	2.1
4 回以上	0.2	0.9	2.3
合計	100	100	100

## 4.2 転送速度に関する評価

本節では、TCP ユニキャストおよびダイレクトネイバーキャストの転送速度の比較評価を行う。

### 4.2.1 評価実験環境

実験のネットワーク環境を、表 4 に記す。ユニキャストおよびマルチキャストの通信レートは 1Mbps に固定しているため、TCP ユニキャストとダイレクトネイバーキャストは両方とも 1Mbps での通信となる。

表 4 転送速度に関する評価実験 ネットワーク環境

通信方式	IEEE802.1g
チャンネル	1
暗号化方式	なし
通信レート	1Mbps

### 4.2.2 評価実験方法

今回の実験では 1 台の送信端末と 1 台から 4 台の受信端末を用いて行った。送信端末は、TCP ユニキャストまたはダイレクトネイバーキャストを用いて、各受信端末に一定の大きさのファイルを 1000 バイト毎に分割して送信する。今回の計測には、1MiB と 10MiB のファイルを使用した。全ての受信端末にファイルが届くまでの時間を計測した。以上の実験

を 10 回行い、その平均値を結果とした。

### 4.2.3 評価実験結果

転送ファイルが 1MiB と 10MiB の場合について、それぞれの受信ノード数に対する平均転送時間の実験の結果を図 5 と図 6 に示す。

いずれの場合も受信端末数が 1 台のときは、TCP ユニキャストのほうが早く転送が完了していることが分かる。これは、TCP ユニキャストは再送をデータリンク層やトランスポート層で行うのに対し、ダイレクトネイバーキャストでは再送をアプリケーション層で行っているため、オーバーヘッドが大きいことや、現在はフロー制御などの実装を行っていないなどの実装上の問題が大きいと考えられる。

しかし、TCP ユニキャストでは、端末数に比例して転送時間が増加していることが分かる。これはユニキャストでは複数の端末への同時送信ができないため、1 台ずつ順次転送されるためである。一方、ダイレクトネイバーキャストはブロードキャストをベースとしたプロトコルであるため、1 パケットを複数端末に送信することが可能であり、端末数が増加しても転送時間の増加が少ないことが分かる。

結果として、受信端末が 4 台の場合、ダイレクトネイバーキャストは 1MiB の場合約 44%、10MiB の場合約 37%の時間でファイルの転送が完了している。さらに台数が増加した場合、この差はさらに顕著になるとと思われる。

## 5. おわりに

本稿では、ダイレクトネイバーキャストの実装として 1 ファイルを複数人へ同時送信するプログラムを作成した。また、評価実験として、ダイレクトネイバーキャストの信頼性に関する評価と、TCP ユニキャストと比較した場合の通信速度に関する評価を行った。

その結果、信頼性を評価する実験では、想定した 3 つの状況全てにおいて 100%のパケットの送受信に成功し、ダイレクトネイバーキャストが信頼性を持つことが確認できた。次に、通信速度の評価実験では、端末台数を増加するに従って、TCP ユニキャストによる転送ではほぼ端末台数に比例した時間がかかっていることに対し、ダイレクトネイバーキャストでは端末台数を 1 台から 4 台に増加させても最大で 21%程度しか増加していないことが分かった。これにより、想定している状況においてダイレクトネイバーキャストは従来手法より優れていることが確認できた。

今後の課題としては以下が考えられる。

- 前方誤り訂正による再送の抑制

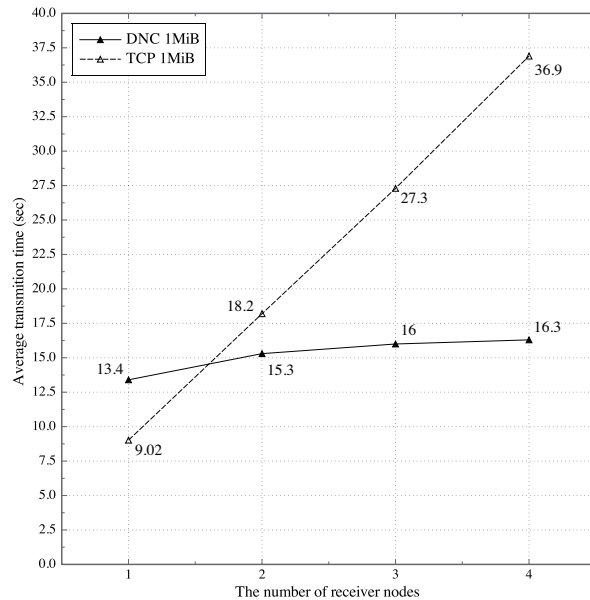


図 5 TCP ユニキャストとダイレクトネイバーキャストの転送時間の比較図 (ファイルサイズ 1MiB)

現在のシステムは、パケットが取得できなかった全てのパケットの再送を行う。パケットを送信する前に前方誤り訂正による符号化を行うことで全体のパケット送信量を減らせる可能性がある。ただし、この手法は一定の処理能力が必要なため、処理能力が低い場合時間がかかると考えられる。また、前方誤り訂正以外にも、受信端末が、他の端末が送信した NACK パケットを解析し、同じ NACK パケットを送信しないなどの処理を行うことで、全体のパケットの量を削減できると考えられる。

● システムの実装

現在の実装は指定したファイルを送信するだけである。想定しているようなローカルクラウドストレージ上での共有に使用するには、リアルタイムのファイル更新検知やファイルの衝突検知などの仕組みが必要になる。これらの問題を解決し、実際にデバイス上にローカルクラウドストレージを実装する必要がある。

謝辞 本研究は、(財)電気通信普及財団研究調査助成(平成 22 年度)の研究助成を受けて行われた。ここに記して謝意を示す。

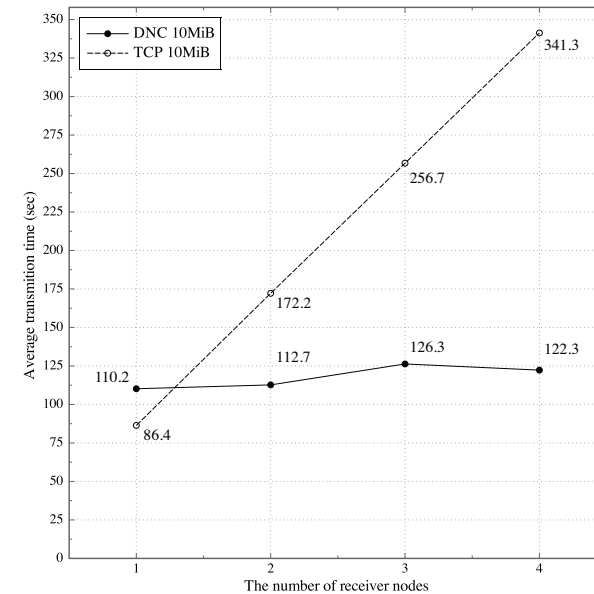


図 6 TCP ユニキャストとダイレクトネイバーキャストの転送時間の比較図 (ファイルサイズ 10MiB)

参 考 文 献

- 1) 田中裕大, 荒川 豊, 田頭茂明, 福田 晃: 近距離無線環境のためのローカルクラウドストレージの提案, 信学技報 CS2010-43, Vol.J85-B, No.11, pp.35-40 (2010).
- 2) 田中裕大, 野田厚志, 田頭茂明, 北須賀輝明, 北口貴史, 津村直樹, 中西恒夫, 福田 晃: シンボルを用いた近距離無線ペアデバイス通信に関する研究, DICOMO2009 マルチメディア, 分散, 協調とモバイルシンポジウム論文集, pp.871-878 (2009).
- 3) 松江英明, 守倉正博: 802.11 高速無線 LAN 教科書, IDG ジャパン (2003).