

# DTN における複製削減のためのライフタイム委譲を用いた複製管理手法の提案

波多野 敏明<sup>‡</sup>植原 啓介<sup>§</sup>

## 概要

Delay- and Disruption- Tolerant Networking(DTN)は遅延や分断の避けられない環境でのネットワーク構築手法として、モバイルアドホックネットワーク、水中通信網、センサーネットワーク、深宇宙探査ネットワークなど幅広い分野への応用が期待されている。DTNではメッセージをノードからノードへの転送の都度複製し、メッセージのコピーを宛先へ届けるルーティングが用いられるが、複製の管理手法が十分でないため複製の保持・転送のために多くのストレージや帯域を消費して効率的でないという問題があった。

本研究では、DTNのストレージや帯域の使用効率改善のための複製管理手法 Lifetime token Delegation(LtD)を提案する。LtDはメッセージのライフタイムを分割した区間をトークンとし複製に割り当て、複製の作成を抑制することで、ネットワーク資源の効率的な使用を促す複製管理手法である。

提案手法をDTNシミュレータ上で動作させ、メッセージの配送率、配送時間、転送量、ストレージ使用量について既存手法との比較評価を行った。

## 1. はじめに

ネットワークの遅延や分断の問題を克服する通信アーキテクチャとして、Delay- and Disruption- Tolerant Networking(DTN) [1, 2, 3] が提案されている。DTNは当初、惑星間通信に伴う伝搬遅延に耐えるネットワークのアーキテクチャとして提案されていたが、分断の問題にも同様のアーキテクチャが有効であることが発見され、交通網上へのネットワーク構築に応用できるのではないかと期待されている。交通网上的ネットワークはノードの移動によりネットワークの接続関係が刻々と変化し、ネットワークは分断される。

接続関係が刻々と変化し、分断のある環境ではある時点のネットワークトポロジを正確に把握することが困難である。既存のDTNルーティングに関する研究では、中継のたびにメッセージの複製を作成し、ネットワーク中に次々と作成される複製のいずれか一つでも宛先に到達すればよしとする手法がしばしば用いられる。しかし、メッセージの複製はネットワーク帯域やストレージ容量を消

費するため、ルーティング効率の向上のためには過剰なメッセージの複製を抑制する必要がある。

本研究では、DTNルーティングの複製管理手法を改善することで、DTNルーティングの効率改善を目指す。分断のあるネットワーク環境を想定することからノードの協調動作による複製管理手法は用いることができない。DTNではメッセージの生存期限を実時間で与えることに着目し、現在時刻から生存期限までの時間をライフタイムとして、時間に基づく複製管理手法である Lifetime token Delegation(LtD)を提案する。キーとして時刻を用いることで各ノードで自律分散的な複製管理を実現する。

## 2. 関連研究

本研究はDTNルーティングの特に複製管理に関する研究である。本節ではDTNルーティングの既存研究でも、特にメッセージの複製を伴う手法について述べる。

Epidemic[4]はメッセージを中継ノードで次々に複製し、宛先まで到達させるという手法の中でも最も簡単なものである。ノードとノードが通信可能な状態すなわちコンタクト時にはすべてのメッセージを複製し、相手のノードへ中継しようと試みる。複製の管理はFirst-in-First-out(FIFO)に基づいており、ノードが中継のために持つストレージの容量が一杯になったとき、ノードの保持している最も受信時刻の古いメッセージを破棄する。中継のために消費される帯域やストレージ容量は大きい、それらに制約がない場合のメッセージの配送時間は最小である。

中継先ノードの選択を取り入れた手法としてPRoPHET[5]やPEAR[6]がある。これらの手法では過去のノードのコンタクトに基づき計算される値から、あるノードがメッセージの中継先として相応しいか計算し、相応しくないノードへの中継を抑制することで帯域やストレージの消費抑制を狙う。

Spray and Wait[7]はメッセージ毎に複製回数の上限を定めることで帯域やストレージ消費抑制を狙う手法である。Spray and Waitでは、メッセージの複製回数が上限に達するまでメッセージを複製してノードに配置し、ノードの接続関係の変化によりメッセージを保持するノードがメッセージの宛先とコンタクトすることを期待する。複製回数の上限により帯域やストレージの消費は少ないが、一方でネットワークの状態を把握することが困難な分断のあるDTN環境でどのように複製回数の上限を与えるのかは問題である。

<sup>‡</sup> 慶應義塾大学 政策・メディア研究科

<sup>§</sup> 慶應義塾大学 環境情報学部

### 3. Lifetime token Delegation

本節では Lifetime token Delegation(LtD) の設計について述べる。

メッセージの複製を抑制するアプローチの一つは、転送先の選択機能と転送メッセージの選択機能で、より宛先への到達に寄与するメッセージのみ複製するよう、転送の絞り込みを的確に行うことである。しかし、ノードの移動性の高い環境では宛先への到達率を正しく判定することは原理的に困難であるため、この手法は用いることはできない。また、Ack もメッセージの複製を抑制するアプローチの一つであるが、原理的に宛先へ届いた後の複製のみ抑制され、宛先に届くまでの複製回数を抑制することができない。そこで、転送先や転送メッセージの選択による抑制や Ack による抑制とも異なる、新たな複製管理機構を提案する必要がある。

交通網上の移動体によるネットワークを想定した場合、積極的に転送を行わなくとも時間の経過によりノードの移動性によってメッセージが宛先に到達する可能性がある。つまり、メッセージをノードに蓄積し、一定の時間転送を行わずに待機させることは、ノードの移動性に基づいてメッセージの複製を伴わない転送と同様の効果が期待できる。他方で、将来の宛先とのコンタクト確率の高いノードにメッセージを蓄積するほどノードの移動性は活用できるが、遅延や分断のある環境では適切なノードの選択がそもそも困難であり、やはり複製による確率向上が必要である。

ここで、どれだけ複製を作成するのが効率的かという課題について明らかにする必要がある。まず、複製回数の制限は単純であるがネットワークの規模やメッセージのライフタイムの長さに対して柔軟性に欠けるため、この手法は適切でない。ノードの移動性とメッセージの蓄積に基づく配送は、どれだけ時間メッセージがノードに蓄積されたかにより転送率が高まると考えられる。つまり、長い時間ノードに蓄積されるメッセージは、ノードの移動により宛先のノードとコンタクトが発生し、転送される機会が多いため複製の個数が少なくとも宛先への到達が見込め、逆にノードに短い時間しか蓄積されないメッセージはそのような偶発性と時間経過に頼った宛先への到達の見込みが低いので複製の個数を増やさねばならない。

提案手法である Lifetime token Delegation では、メッセージのライフタイムに着目して複製のトークンとすることで、ライフタイムの残りが少ないメッセージの複製を積極的にを行い、ライフタイムの長いメッセージは消極的に複製を行う。

#### 3.1 Lifetime token Delegation の動作概要

Lifetime token Delegation ではルーターはメッセージと関連づけられたトークンを持つ。トークンは有効期間を持ち、メッセージのライフタイム (メッセージが作成された時刻から TTL が終了するまでの期間) の一部を指す。現在時刻が有効期間となるトークンを持つノードで

は、そのトークンと対応するメッセージの複製を行うことができ、現在時刻が有効期間でないトークンを持つノードはメッセージの複製が抑制される。

Lifetime token Delegation ではメッセージのライフタイム中の任意の時間を有効期間として指し示すトークンをネットワーク中にただ一つだけ存在するよう制御する。これによりネットワーク中で新たに複製を行えるメッセージの複製はただ一つとなり、無秩序に複製が行われなくなる。

#### 3.2 メッセージ生成時の動作

メッセージ生成時、メッセージの発信元ノードではメッセージに対応する最初のトークンが生成される。最初のトークンはメッセージの生成時刻からライフタイムが終了する時刻までを有効期間として持つ。メッセージ生成時の動作を図 1 に擬似コードにより示す。

```
// メッセージ生成時、Lifetime と等しい
// 有効期間のTokenを与える
public void createNewMessage(Message m)
{
    // getTime() => 現在時刻
    // Message.getTtl() => メッセージのTTL
    tokens.put(
        m.getId(),
        new Token(
            getTime(),
            getTime() + m.getTtl()));
}
```

図 1 LtD のメッセージ生成時の動作

#### 3.3 コンタクト時の挙動

コンタクトしたノードがメッセージの宛先でない場合、メッセージのトークンが確認される。トークンが現在時刻を含むときメッセージは複製転送を行い、そうでなければメッセージは転送されない。

複製の可否の判定を擬似コードで図 2 に示す。

```
// このノードの持つtokenが現在時刻を
// 含むときメッセージは複製可能
public boolean canReplicate(Message m)
{
    Token token = tokens.get(m.getId());
    if( token.timeBegin <= getTime()
        && getTime() < token.timeEnd ) {
        return true;
    }
    return false;
}
```

図 2 LtD のメッセージの複製可能判定

コンタクト時のノードの動作を擬似コードで図 3 に示す。

```

// 転送可能メッセージの有無を
// 定期的なチェック
public void update()
{
    // 転送可能なメッセージのリスト
    Queue<Message> relayTransferQueue
        = new Queue<Message>();

    // ノードの持つメッセージについて
    for( Message m : messages ) {
        if( canReplicate(m) ) {
            // メッセージが複製可能な場合
            // 転送キューに追加
            relayTransferQueue.add(m);
        }
    }
}

```

図3 LtD のコンタクト時の動作

メッセージの複製が行われる場合、複製元のノードが持つトークンの有効期間を  $n$  分割し、分割した有効期間の現在時刻を含む部分を複製先のノードにトークンとして渡す。今回の実装では  $n = 3$  とおいた。複製先に渡された有効期間は複製元のトークンから取り除かれる。ノード間で有効期間を受け渡すことで、新たに複製をすることのできるメッセージはどの時刻をとっても常に1つとなる。メッセージの複製転送に伴うライフタイムトークンの委譲動作を擬似コードで図4に示す。

### 3.4 Lifetime token Delegation 全体の動作

まず、図5に Lifetime token Delegation のアプローチであるノードの移動性の活用について示す。生成元ノードから複製転送されるメッセージは、ライフタイムトークンの委譲に従って、緩やかに複製の数を増やしながらか宛先への到達を目指す。複製時期の早いメッセージほど長い待ち時間を持つため、よりネットワーク中に拡散することが期待される。ネットワーク中に拡散した複製を持つノードのどれか一つでも宛先とコンタクトすればメッセージは到達する。

Lifetime token Delegation 全体の動作概要を図6に示す。

ノード  $S$  がメッセージの生成元ノード、ノード  $D$  がメッセージの宛先ノード、ノード  $N_1, N_2, N_3$  はそれぞれ宛先でも生成元でもないノードを表す。

時刻  $T_0$  はメッセージの生成時を表す。メッセージの生成時、現在時刻からメッセージの TTL 終了までの期間を持つトークンがノード  $S$  に生成される。

時刻  $T_1, T_2, T_3$  がそれぞれメッセージの宛先でないノードとのコンタクトを表す。コンタクト時にライフタイムトークンの現在時刻を含む部分から  $1/3$  切り出し、転送先ノードに委譲する。

時刻  $T_3$  では時刻  $T_1$  にトークンの有効期限を委譲して、

```

// 送信時、token の前方から 1/3を
// メッセージに付加して送信する
// 自身は token の後方 2/3を持つ
public void messageSend(Message out)
{
    Token tokenOrig;
    Token tokenSend, tokenRecv;
    double now, split, end;

    tokenOrig
        = tokens.get(out.getId());

    now = getTime()
    end = tokenOrig.getTimeEnd();
    split = ( end - now ) / 3 + now;

    tokenSend = new Token(split, end);
    tokens.put(out.getId(),
        tokenSend);

    tokenRecv = new Token(now, split);
    out.addProperty('LtD.token',
        tokenRecv);
}

// 受信メッセージの持つ token を
// このノード token として保持する
public void messageReceive(Message inc)
{
    Token tokenRecv;
    tokenRecv
        = inc.getProperty('LtD.token');
    tokens.put(inc.getId(), tokenRecv);
}

```

図4 LtD のメッセージの転送に伴う Lifetime token の委譲

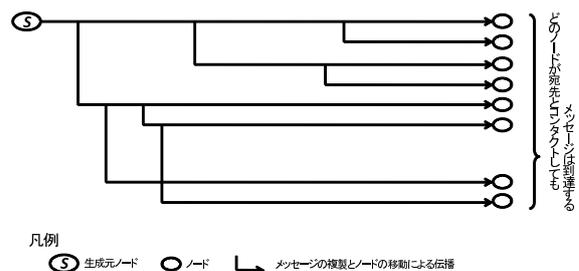


図5 LtD を用いたメッセージ複製転送の模式図

メッセージを複製できない状態であったノード  $S$  が、時刻の経過により再びメッセージの複製が可能となる状況を表す。時刻  $T_1$  から  $T_3$  の間にノード  $S$  と他のノードにコンタクトがあったとしても、ノード  $S$  がトークンを持たない期間には複製は行われない。

時刻  $T_4$  ではライフタイムトークンを持たないノード  $N_2$  とメッセージの宛先  $D$  にコンタクトがあった場合を表す。トークンの如何に関わらず、宛先へはメッセージが送信され、メッセージの転送が完了する。

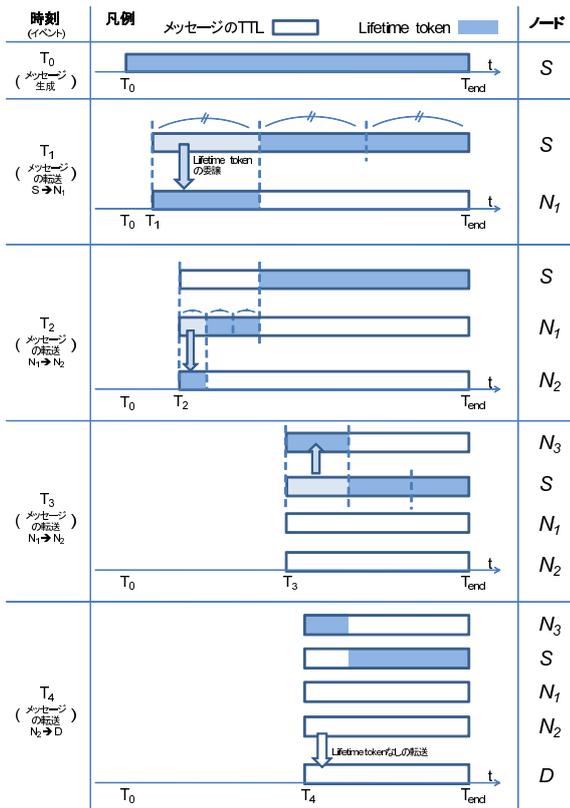


図6 メッセージの転送と Lifetime token の委譲の模式図

#### 4. Lifetime token Delegation + PRoPHET

Lifetime token Delegation(LtD) はライフタイムを活用することで複製の削減を狙う、複製管理の手法である。複製管理手法のみでは DTN ルーティングとして動作し得ないため、提案手法である LtD と、転送先選択機能と転送メッセージの選択機能としての PRoPHET を組み合わせた新たな DTN ルーティング LtD + PRoPHET を提案する。PRoPHET は転送先選択と転送メッセージ選択についてよく知られた手法であることからこれを選んだ。

#### 5. 評価

交通網上に構築された DTN を模したシミュレーションにより、提案手法である Lifetime token Delegation の評価を行った。

交通システムの通信手段としての DTN 活用のためには DTN ルーティングにおいてメッセージの複製を如何に抑制するかが課題である。

評価軸としては以下の四点を用いる。

- メッセージの到達率  
生成されたメッセージのうち宛先へ到達したメッセージの割合である。

- メッセージの到達所要時間 (平均)  
宛先へ到達したメッセージのみについて、生成されてから到達するまでに要した時間の平均である。
- メッセージの転送回数 (平均)  
メッセージがノードからノードへ転送された回数の平均である。
- メッセージのストレージ使用量 (平均)  
メッセージがストレージに蓄積されていた時間の平均である。

メッセージ到達率と到達所要時間から本手法によるメッセージの到達に及ぼす影響を、メッセージの転送回数とストレージ使用量から本手法がネットワーク資源の使用量に及ぼす影響を評価する。

##### 5.1 評価環境の設定

提案手法の評価のためネットワークシミュレーターを用いた評価環境を構築し、既存手法との比較を行う。

シミュレーションの想定シナリオは、都市部の交通網上へのネットワーク構築である。ノードとして歩行者と車両、路面電車の3種類のモビリティモデルを持つノードをシミュレートする。ノード上では DTN ルーティング手法が動作しており、すべてのノードが一定量のストレージと無線によるリンクを備え、ノード同士が無線の通信範囲に入ることによって一定の帯域でノード間でメッセージの送信が行えるものとする。

##### 5.2 シミュレータ

都市部の交通シミュレーションとその上で動作する DTN のシミュレーションに実績があることから、シミュレータとしては The Opportunistic Network Simulator (The ONE) [8] を用いた。

The ONE は DTN 環境におけるルーティング手法の評価のために開発されたシミュレーターであり、Java で記述され GPLv3[9] ライセンスの元で公開されているオープンソースソフトウェアである。

##### 5.3 シミュレーションシナリオ

シミュレーションするモビリティシナリオには The ONE の提供するマップベースのシナリオ *Helsinki City Scenario(HCS)* を用いた。HCS ではマップはヘルシンキ近郊およそ 4km 四方の道路地図を擬えており、ノードは歩行者と車両と路面電車のいずれかのプロファイルに属し、プロファイルに従い行動する。

歩行者と車両のプロファイルのモビリティモデルは Shortest Path mobility model [8] である。Shortest Path mobility model は、ノードが道路地図上の任意の交差点を目的地として最短移動距離となる経路を一定速度で移動し、目的地で一定時間停止した後、新たな目的地を選び再び移動するモビリティモデルである。歩行者のプロファイルはすべての道路地図上のすべての道路を移動できる、車両のプロファイルは道路地図上の幹線道のみ移動可能である。路面電車のプロファイルのモビリティモデルは、ノードはあらかじめ道路地図上に設定された路線に沿って一定速度で移動し、あらかじめ決められた停

表1 ノードに関するパラメータ

パラメータ	値域		
	歩行者	車両	路面電車
移動速度	1.8-5.4 (km/h)	10-50 (km/h)	25-36 (km/h)
停止時間	10-120 (sec)	10-120 (sec)	10-30 (sec)
通信可能距離	10 (m)		
通信帯域	16 (Mbps)		
ストレージ容量	10 (MB)	100 (MB)	

表2 トラフィックに関するパラメータ

パラメータ	値域
メッセージ TTL	120 (min)
メッセージサイズ	500-1000 (kByte)
メッセージ生成間隔	30 (message/sec)

止場所で一定時間停止するモビリティモデルである。移動速度と停止時間は各プロファイルに値域が定められており、停止毎に正規分布に従い新たな値が選ばれる。

各ノードは無線のインターフェースを備え、2ノードが一定の距離内にあるときこれを通信可能とみなす。また、各ノードはメッセージの中継のためネットワークに一定量のストレージを提供するものとする。

ノードの持つパラメータをプロファイル毎に表1に示す。

#### 5.4 トラフィックシナリオ

シミュレーションではトラフィックのシナリオとして、移動体間の接近通知や交通情報システムやユーザーなどによるメッセージング一般を想定し、DTN 網上のノードからノードへとメッセージが送信される環境を想定する。

トラフィックに関するシミュレーションのパラメータを表2に示す。メッセージは一定の間隔でランダムなノードで生成され、宛先もランダムなノードが選ばれるものとする。

### 6. 実験結果

#### 6.1 ライフタイムをシミュレーションパラメータとした比較

メッセージ生成時のライフタイムをパラメータとした際の各手法の実験の結果を図7に示す。メッセージの宛先への到達率を図7(a)に、宛先に到達したメッセージについて生成されてから宛先に到達するまでの所要時間の平均を図7(b)に、メッセージが複製や転送によりノード間を移動した回数の平均を図7(c)に、メッセージがノードのストレージに蓄積された時間の平均を図7(d)に示した。

#### 6.2 ストレージ容量をシミュレーションパラメータとした比較

各ノードの持つストレージの容量をパラメータとした際の各手法の実験の結果を図8に示す。メッセージの宛先への到達率を図8(a)に、宛先に到達したメッセージについて生成されてから宛先に到達するまでの所要時間の平均を図8(b)に、メッセージが複製や転送によりノード間を移動した回数の平均を図8(c)に、メッセージがノードのストレージに蓄積された時間の平均を図8(d)に示した。

#### 6.3 ノードの数をシミュレーションパラメータとした比較

ノード数をパラメータとした際の各手法の実験の結果を図9に示す。メッセージの宛先への到達率を図9(a)に、宛先に到達したメッセージについて生成されてから宛先に到達するまでの所要時間の平均を図9(b)に、メッセージが複製や転送によりノード間を移動した回数の平均を図9(c)に、メッセージがノードのストレージに蓄積された時間の平均を図9(d)に示した。

#### 6.4 メッセージ発生間隔をシミュレーションパラメータとした比較

ネットワーク全体におけるメッセージの発生間隔をパラメータとした際の各手法の実験の結果を図10に示す。メッセージの宛先への到達率を図10(a)に、宛先に到達したメッセージについて生成されてから宛先に到達するまでの所要時間の平均を図10(b)に、メッセージが複製や転送によりノード間を移動した回数の平均を図10(c)に、メッセージがノードのストレージに蓄積された時間の平均を図10(d)に示した。

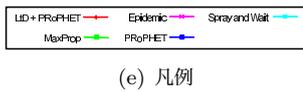
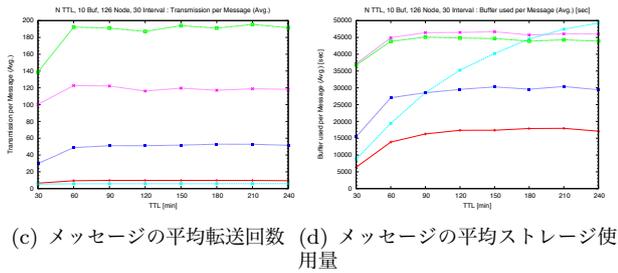
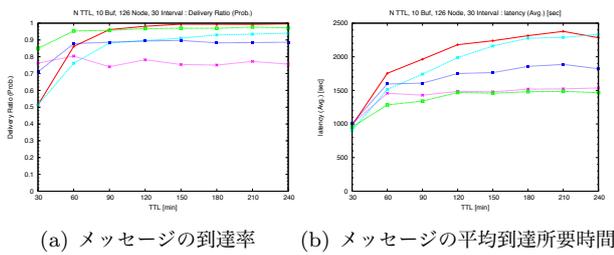


図7 ライフタイムの変化に伴う影響の比較

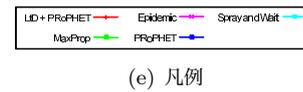
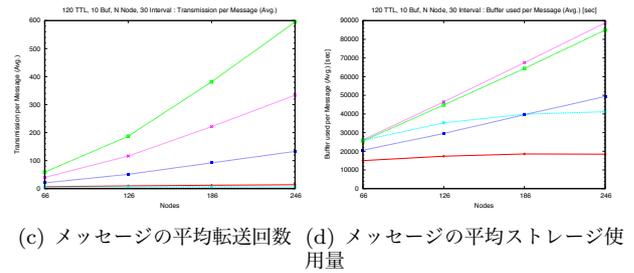
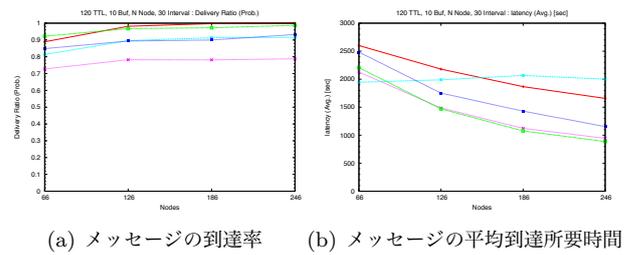


図9 ノード数の変化に伴う影響の比較

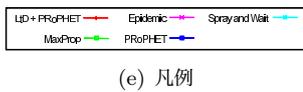
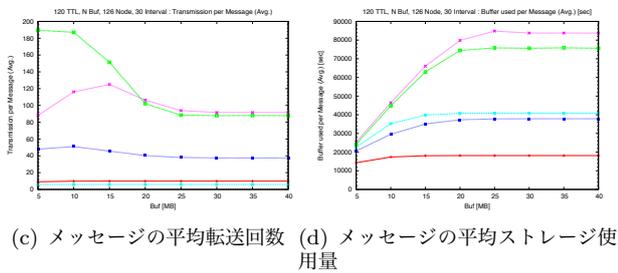
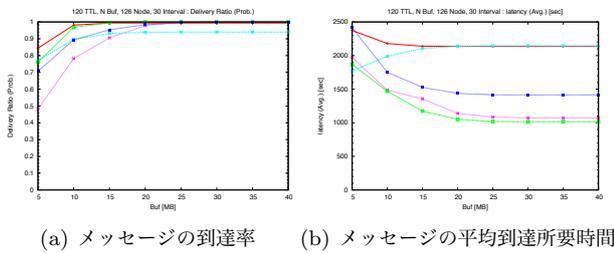


図8 ストレージ容量の変化に伴う影響の比較

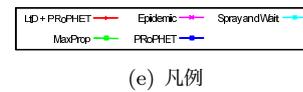
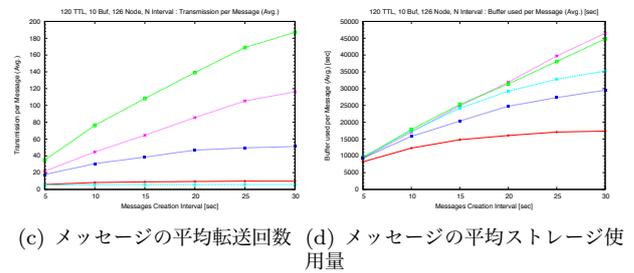
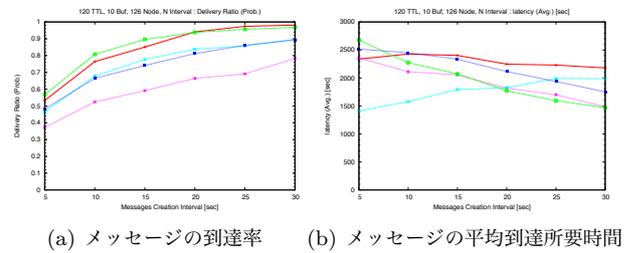


図10 メッセージの発生間隔の変化に伴う影響の比較

## 6.5 実験結果の考察

以上のシミュレーション結果より評価軸について考察を行う。

- メッセージの到達率について  
図 7(a)、図 8(a)、図 9(a) から見て取れるように  $TTL < 120min$  の場合を除いて、LtD + PRoPHET は MaxProp と同程度のメッセージ到達率を達成している。図 7(a) の  $TTL < 120min$  の場合 LtD + PRoPHET はメッセージの到達率が MaxProp と比べて大きく下がる、これは図 8(b)、図 9(b) などが示すように LtD + PRoPHET はメッセージの宛先への到達所要時間が MaxProp と比べて大きいため宛先に到達する以前にメッセージのライフタイムが切れるためと思われる。
- メッセージの到達所要時間について  
図 7(b)、図 8(b)、図 9(b) から見て取れるように LtD ではメッセージの到達に MaxProp のおおよそ 2 倍程度時間が必要である。しかし Epidemic、PRoPHET、Spray and Wait と比較すれば同程度の所要時間である。
- メッセージの転送回数について  
図 7(c)、図 8(c)、図 9(c) から LtD + PRoPHET ではメッセージの転送回数が低く抑えられていることがわかる。特に、メッセージの到達率について本手法と同程度の MaxProp と比較した場合、転送回数を 1/10 から 1/20 程度まで抑制することができた。
- メッセージのストレージ使用量  
図 7(d)、図 8(d)、図 9(d) から LtD ではストレージの使用量が比較したどの手法よりも低く抑えられていることがわかる。今回行ったシミュレーションの多くの領域で、MaxProp の 40% 程度までストレージ使用量を抑制することができた。

図 7(a)、図 8(a)、図 9(a) から見て取れるように  $TTL < 120min$  の場合 LtD + PRoPHET はメッセージの到達率で MaxProp と比べて大きく劣るものの、それ以外の範囲では MaxProp と同程度の到達率を達成している。

図 7(c)、図 8(c)、図 9(c) が示すように LtD + PRoPHET ではメッセージの転送回数が低く抑えられている。図 7(d)、図 8(d)、図 9(d) からはストレージの使用量についても LtD + PRoPHET は低く抑えられていることがわかる。

また、一連の図 10 よりメッセージの発生間隔が変化し、ネットワーク全体を流れる時間あたりのメッセージ量が増減した場合も、LtD + PRoPHET ではメッセージの転送回数やストレージの使用量が低く抑えられていることがわかる。

## 7. まとめ

本研究では DTN におけるメッセージであるメッセージの持つライフタイムに着目し、ライフタイムを複製管理のトークンとして用いることで、過剰な複製を抑制す

る複製管理手法 Lifetime token Delegation(LtD) を提案した。LtD は、メッセージの複製転送だけでなく移動性も活用してメッセージを到達させる手法であり、ノードの移動性を活用して、メッセージをストレージに蓄積し、転送を行わずに一定時間待機させることで、複製回数やストレージ使用量の削減と到達率の両立を達成する手法である。

また、提案手法を、交通情報システムを模した DTN 評価用のシミュレーション上で評価した。既存手法との比較の中でも、宛先へのメッセージ到達率に優れることが知られた手法である MaxProp と比較して、到達率について提案手法が MaxProp とほぼ同じに保った上で、複製回数を 1/20 まで削減、ストレージの使用量を 40% まで削減するなど、既存手法に対してと大きくネットワーク資源の使用量の面で優れることを確認した。

## 参考文献

- [1] Kevin Fall. A delay-tolerant network architecture for challenged internets, 2003. 1.
- [2] Vinton G. Cerf, Scott C. Burleigh, Robert C. Durst, Kevin Fall, Adrian J. Hooke, Keith L. Scott, Leigh Torgerson, and Howard S. Weiss. Delay-tolerant networking architecture (rfc 4838), April 2007. 1.
- [3] K. Scott and S. Burleigh. Bundle protocol specification (rfc 5050), November 2007. 1.
- [4] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. *Technical report, Duke University*, 2000. 2.
- [5] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *Lecture Notes in Computer Science*, pages 239–254, 2004. 2.
- [6] Hideya Ochiai and Hiroshi Esaki. Mobility entropy and message routing in community-structured delay tolerant networks, 2008. 2.
- [7] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, page 259. ACM, 2005. 2.
- [8] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMU-Tools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST. 5.2, 5.3
- [9] Free Software Foundation. Gnu general public license, Junu 2007. 5.2