

平成27年度 卒業論文

和文題目

**NTMobile** を用いた  
**E2E** チャットアプリケーションの提案

英文題目

**Proposal of Chat Application of  
End To End Communication using NTMobile**

情報工学科 渡邊研究室  
(学籍番号: 120430073)

中村 隼大

提出日: 平成28年2月10日

名城大学理工学部



## 概要

モバイルネットワークの普及に伴い、チャットが重要なコミュニケーションツールとなっている。チャットはクライアントサーバシステムで実現するのが一般的である。しかし、サーバから情報漏洩する懸念や、サーバの障害・二重化等に対する管理負荷が大きいという課題がある。我々は、端末がどのようなネットワーク環境にいてもエンドツーエンドで通信を行うことができるNTMobile(Network Traversal with Mobility)を提案している。本論文ではNTMobileを用いたエンドツーエンド通信によるチャット通信方式を提案する。Linux上で提案方式を実装し、仮想環境上で動作検証を行った。また、提案方式の性能評価を行い、NTMobileを用いたチャット通信では通信パケット数が減少することを確認した。



# 目次

|       |                   |    |
|-------|-------------------|----|
| 第1章   | はじめに              | 1  |
| 第2章   | 既存のチャットアプリケーション   | 3  |
| 2.1   | クライアントサーバ型チャットの概要 | 3  |
| 2.2   | クライアントサーバ型チャットの構成 | 4  |
| 2.3   | クライアントサーバ型チャットの課題 | 5  |
| 第3章   | NTMobile          | 6  |
| 3.1   | NTMobileの概要       | 6  |
| 3.2   | NTMobileの動作       | 6  |
| 3.2.1 | アドレス情報の登録処理       | 7  |
| 3.2.2 | トンネル構築処理          | 7  |
| 3.3   | NTMobileを用いる利点    | 8  |
| 第4章   | 提案方式              | 9  |
| 4.1   | 提案方式の概要           | 9  |
| 4.2   | 通信方式              | 10 |
| 第5章   | 実装と性能評価           | 11 |
| 5.1   | 実装                | 11 |
| 5.2   | 動作検証              | 12 |
| 5.3   | 性能評価              | 13 |
| 第6章   | 従来方式との比較          | 14 |
| 第7章   | まとめ               | 15 |
|       | 謝辞                | 17 |
|       | 参考文献              | 19 |
|       | 研究業績              | 21 |



# 第1章 はじめに

スマートフォンやタブレット端末等の携帯端末の普及やモバイルネットワークの発展により、インターネット利用の需要が急激に増加している。それに伴い、コミュニケーションツールとしてチャットを利用したいという要求が高まっている。

現在のネットワークでは、グローバルアドレスの枯渇を短期的に解決するため、通信経路上に NAT(Network Address Translation) が存在していることが多い。NAT が存在している通信経路上では、グローバルネットワーク側から NAT 配下のプライベートネットワーク側に対して通信を開始することができず、エンドツーエンドの通信が阻害されている。そのため、端末同士で情報を交換する場合であっても、グローバルネットワーク上にサーバを設置し、端末がサーバとの間でデータをアップロード/ダウンロードする方法が取られている。

チャットアプリケーションにおいても、インターネット上に存在するサーバを介してチャットを行う必要がある。このシステムでは、クライアントがサーバに対してメッセージを送付し、サーバが各クライアントに同一メッセージを配信する。クライアントはサーバに対して、常時 Keep Alive を行うことにより NAT テーブルを維持する。これにより、NAT 越え問題の解決を図ってきた。しかし、クライアントサーバシステムではサーバの管理者が全ての情報を取得できるため、セキュリティ上問題があるという指摘がある。また、サーバの開発負荷を軽減するために HTTP で実現されることが多く、一度のメッセージ送信に多くのパケットを送信する必要がある。

我々は、端末がどのようなネットワーク環境に存在してもエンドツーエンドで通信を行うことができる NTMobile(Network Traversal with Mobility) を提案している [1-4]。NTMobile を適用すると NTMobile のシグナリング機能により、通信開始側の端末と通信相手の端末との間にエンドツーエンドのトンネル経路が構築される。これにより、ユーザは NAT の存在を意識することなくエンドツーエンドの通信を行うことができる。NTMobile は、一意に割り当てられる仮想 IP アドレスに基づくパケットを実 IP アドレスによりカプセル化し、実 IP アドレスの変化をアプリケーションに対して隠ぺいすることによってこれを実現する。

本論文では NTMobile を用いたエンドツーエンド通信によるチャット通信方式を提案する。送信データがキャラクタデータの場合は UDP で通信を行い、ファイル（長データ）の場合は TCP で通信を行う。提案方式ではサーバを利用しないため、サーバから情報漏洩する心配がなく、サーバの二重化・障害等に対する管理負荷が軽減される。また、UDP トンネルを構築するまでのシグナリング処理は初回のみ行うだけでよく、メッセージ送信にかかるパケット数も少なくなるという特徴がある。

以後、2 章では既存のチャットアプリケーションについて、3 章では NTMobile について述べる。4 章では提案する NTMobile を用いたチャット通信方式について説明し、5 章では提案手法の実装

と動作検証の結果，性能評価を示す．6章では従来方式との比較を行い，最後に7章でまとめる．

## 第2章 既存のチャットアプリケーション

本章では、既存のチャットアプリケーションの実現方法について述べる。

### 2.1 クライアントサーバ型チャットの概要

既存のチャットアプリケーションは、インターネット上に存在するサーバとクライアントから成るクライアントサーバシステムで実現される。図1にクライアントサーバ型チャットシステムの概要を示す。通信開始側の端末をイニシエータ、通信相手端末をレスポндаと表す。また、イニシエータはグローバルネットワーク、レスポндаはNAT配下のプライベートネットワークに存在するものとする。

NAT配下に存在するレスポндаは通信経路を確保するため、サーバに対してKeep Aliveを行う。クライアントサーバ型チャットシステムでは、ユーザがチャットメッセージを送信すると、イニシエータからサーバに対してチャットデータを送信する。その後、サーバがレスポндаにメッセージの受信を通知することでレスポндаがサーバにチャットデータを取りに行く。このようにクライアントサーバ型のチャットでは、クライアントの存在するネットワーク環境に関わらず、必ずインターネット上のサーバを経由して通信を行う。

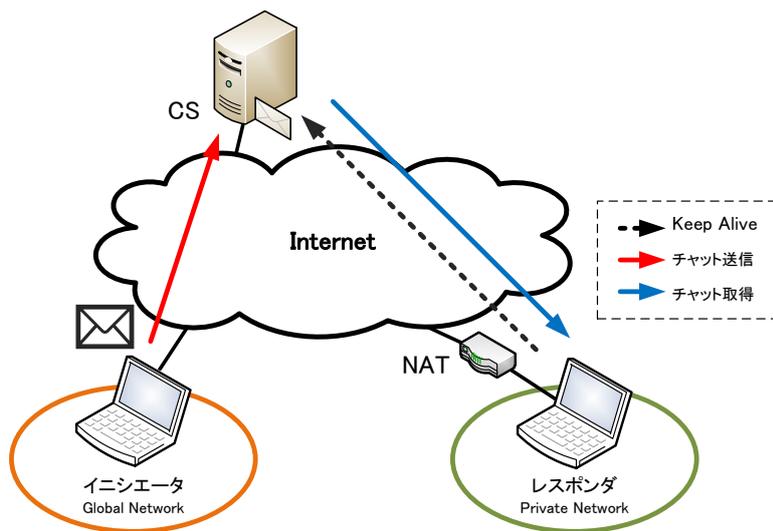


図1 クライアントサーバ型チャットの概要

## 2.2 クライアントサーバ型チャットの構成

本研究室では、クライアントサーバシステムを用いたチャットアプリケーションとして、Mobilineを開発した実績がある。図 2 に Mobiline のチャット通信シーケンスを示す。イニシエータはグローバルネットワーク、レスポンドは NAT 配下のプライベートネットワークに存在するものとする。

NAT 配下に存在するレスポンドは CS(Chat Server) に対して、常時 Keep Alive を行うことにより NAT テーブルを維持し、通信経路を確保する。ユーザがチャットメッセージを入力して送信すると、イニシエータが CS に対してメッセージを送付する。この時、イニシエータは自身の FQDN(Fully Qualified Domain Name)、Application ID、Auth\_token を記載して CS に対して送信する。FQDN はホスト名、ドメイン名等全てを省略せずに指定した記述形式である。Application ID と Auth\_token は、それぞれユーザが登録したユーザ名、パスワードを格納している。CS はこれらの情報を基に正規ユーザであるか認証を行う。また、CS はデータを識別する Hash ID を生成し、レスポンドにメッセージの受信を通知する。CS はイニシエータにメッセージ送信に対する応答を返す。メッセージの受信を知ったレスポンドは、CS にメッセージ取得要求を行い、データを取りに行く。この時、CS は Hash ID で自身のテーブルを検索し、該当するデータを返す。正常にデータを受信すれば、CS はレスポンドにメッセージ取得要求に対する応答を返す。

サーバは機能が多彩であり、PHP 言語で記述するケースが多く、返信手順は HTTP で実現するのが一般的である。そのため、各処理において TCP によるコネクション確立・コネクション切断が行われる。このチャットシステムは HTTP で実現されているため、応答が返ってきたレスポンドは、メッセージ取得が完了したことを CS に知らせる。また、サーバを利用した通信を行っているため、CS はイニシエータの FQDN を用いて自身のデータベースを検索し、該当レコードの状態が既読通知が必要であることを示している場合、レスポンドが既読したことをイニシエータに通知する。

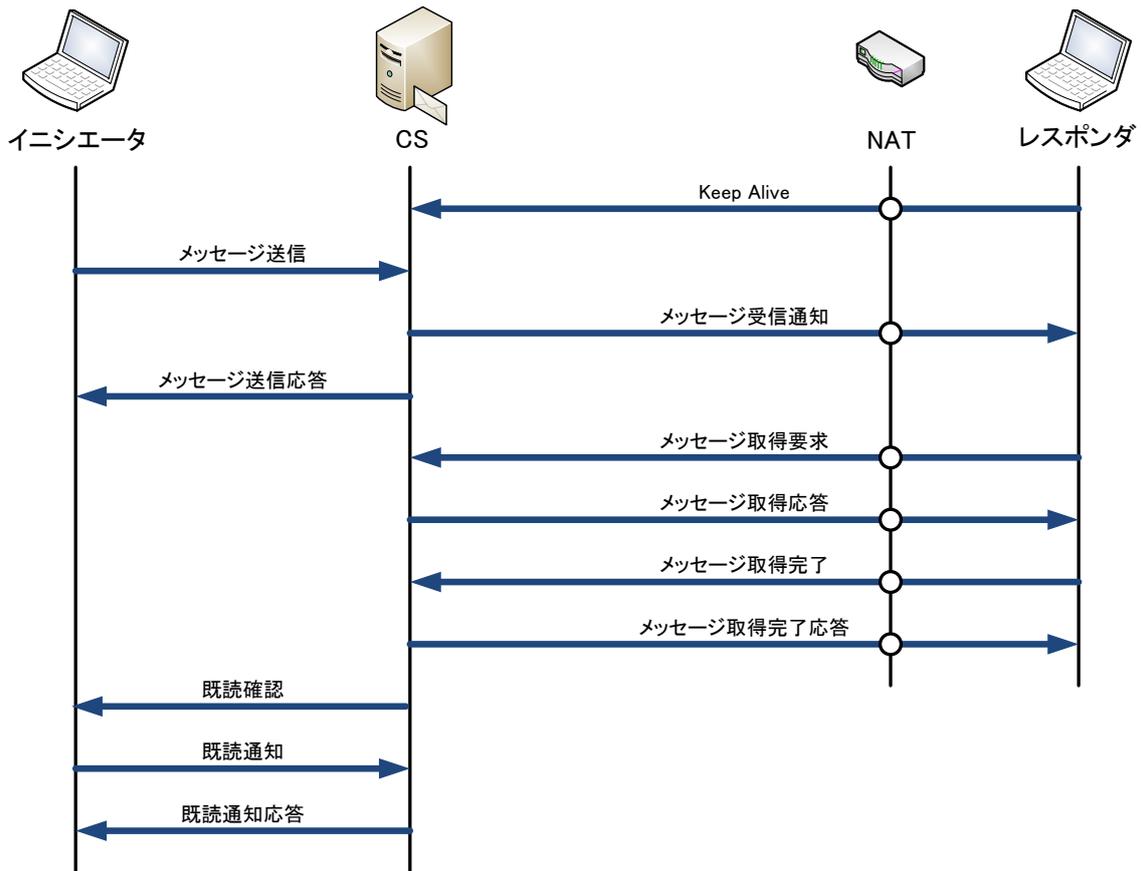


図 2 Mobicline チャットシーケンス

## 2.3 クライアントサーバ型チャットの課題

クライアントサーバシステムでは、サーバの管理者が情報を取得できるためサーバから情報漏洩する懸念があり、業務での利用が難しい。また、サーバの障害・二重化等の管理が必須であり、管理負荷が大きい。さらに HTTP は、コネクション確立で 3 パケット、コネクション切断で 4 パケットを必要とする。すなわち、一回のチャットメッセージ送信に 48 パケット必要である。メッセージ送信毎に全ての処理を実行しなければならないため、無駄なトラフィックが大きい。

## 第3章 NTMobile

本章では、提案方式で用いる NTMobile の概要について述べる。

### 3.1 NTMobile の概要

図 3 に NTMobile の構成を示す。NTMobile は、NTMobile の機能を実装した端末 (NTM 端末) と NTM 端末に対してアドレス情報の管理やトンネル構築指示を行う DC (Direction Coordinator) により構成される。NTM 端末は、起動時に自身の実 IP アドレスが等の情報を DC に対して登録する。また、DC から FQDN と仮想 IP アドレスが割り当てられ、アプリケーションは仮想 IP アドレスに基づいて通信を行う。DC は仮想 IP アドレスの割り当て管理や、NTM 端末に対してトンネル構築等の指示を出す。NTM 端末に割り当てられる仮想 IP アドレスは一意的なアドレスであり、各 DC は自身に割り当てられたアドレス空間から重複が起きないように割り当てを行う。

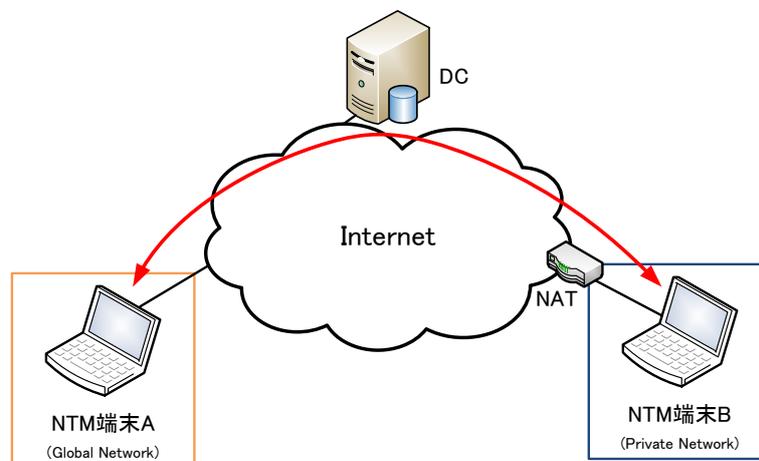


図 3 NTMobile の構成

### 3.2 NTMobile の動作

以降の説明では、通信開始側の NTM 端末をイニシエータ、通信相手側の NTM 端末をレスポンドと表記する。

### 3.2.1 アドレス情報の登録処理

図 4 に IPv4 グローバルネットワークへ接続したイニシエータ、および IPv4 プライベートネットワークへ接続したレスポндаがアドレス情報を登録する際の処理を示す。NTM 端末は端末起動時に、DC に対して自身のアドレス情報の登録を行う。NTM 端末は、自身の IP アドレスや FQDN 等の情報を記載した NTM Registration Request を DC に対して送信する。DC は、NTM 端末の FQDN から NTM 端末が一意に決まる Node ID を生成する。また、DC のデータベースに NTM 端末の端末情報を登録し、NTM 端末に仮想 IP アドレスを割り当てる。DC は NTM 端末に対して仮想 IP アドレスを記載した NTM Registration Response を返信する。アドレス情報の登録完了後、NAT 配下に存在するレスポндаと DC との間で定期的なメッセージの交換 (Keep Alive) を行うことにより、制御メッセージ用の通信経路を確保する。

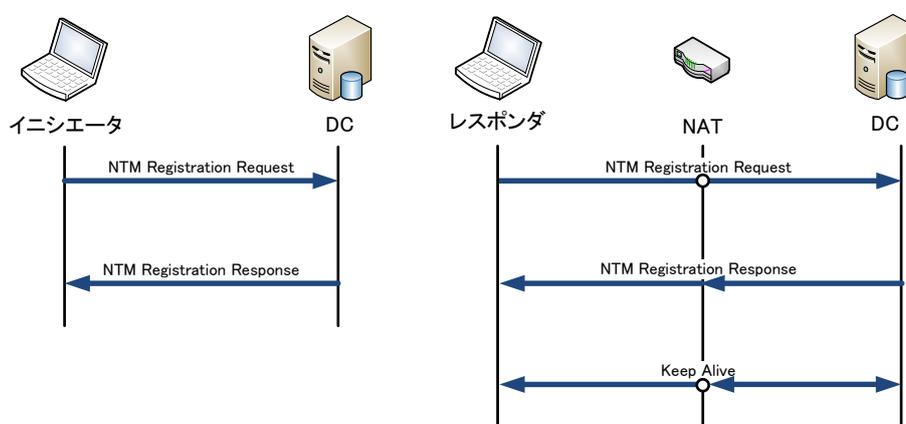


図 4 アドレス情報の登録処理

### 3.2.2 トンネル構築処理

図 5 にイニシエータからレスポндаに対してトンネル通信経路を確立するまでのシーケンスを示す。尚、イニシエータは IPv4 グローバルネットワーク、レスポндаは IPv4 プライベートネットワークに存在していることとする。

はじめに、イニシエータは DC へ NTM Direction Request を送信することにより、レスポндаと通信を行うためのトンネル構築指示を要求する。NTM Direction Request にはイニシエータおよびレスポндаの FQDN が記載されている。DC はレスポндаの FQDN を基に、自身のデータベースからレスポндаのアドレス情報を取得する。DC は取得したイニシエータとレスポндаのアドレス情報から端末の位置関係を認識し、トンネル通信時の経路を決定する。この場合、IPv4 グローバルネットワークと IPv4 プライベートネットワーク間の通信であるため、エンドツーエンドのトンネル通信経路となる。経路を決定した DC はイニシエータとレスポндаへ NTM Route Direction を送信し、レスポндаからイニシエータへ Tunnel Request を送信するよう指示する。このとき DC とレスポнда間には、Keep Alive により通信経路が確保されているため、DC から NAT 配下のレ

スポンダに対して通信を行うことができる。イニシエータおよびレスポндаに正常に NTM Route Direction が受信されれば、その応答として NTM ACK を DC に返す。NTM Route Direction が受信されなければ、NTM NACK を DC に返す。NTM ACK が返信された時、レスポндаは DC の指示に従ってイニシエータへ NTM Tunnel Request を送信する。NAT 配下に存在するレスポндаからイニシエータへ NTM Tunnel Request を送信することにより、レスポндаとイニシエータとの間に実 IP アドレスによる UDP トンネルを構築する。

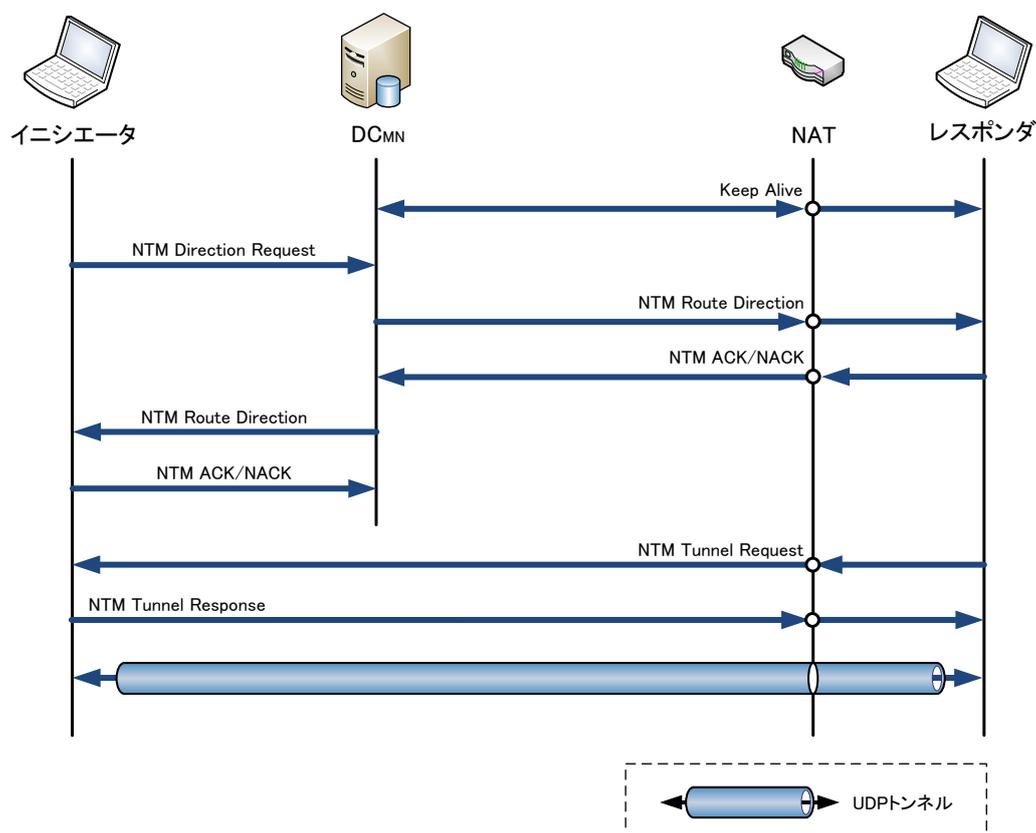


図 5 トンネル構築処理

### 3.3 NTMobile を用いる利点

3.2.2 項で構築した UDP トンネルにより、NTM 端末同士がエンドツーエンドの通信を行うことができるようになる。すなわち、NTMobile を適用することにより、ユーザは NAT の存在を意識せず通信を行うことができる。

NTMobile のシーケンスは全て UDP で通信を行っている。図 5 において、NTMobile を用いてイニシエータとレスポндаの間にトンネル通信経路を確立するまでに、8 パケット必要である。一度トンネル経路をエンドツーエンドで確立したら、以後の通信はトンネル構築処理は不要である。

## 第4章 提案方式

本章では、NTMobile を用いたエンドツーエンド通信によるチャット通信方式について述べる。NTMobile による UDP トンネル上で UDP/TCP パケットを送受信し、イニシエータとレスポнда間でエンドツーエンドのチャットを実現する。

### 4.1 提案方式の概要

図 6 に提案方式の概要を示す。本提案では、NTMobile の NTM Signaling 処理により構築された UDP トンネルを経由してチャットを行う。

本提案方式では、一度 NTMobile によりイニシエータとレスポндаの間に UDP トンネルを構築し、その後のチャットメッセージの交換はエンドツーエンドで行うことができる。この手法ではサーバを利用しないため、サーバからの情報漏洩の懸念が無く、サーバの二重化等の管理負荷低減が期待できる。また、通信を行う度にサーバを経由することがなく、エンドツーエンドで通信が可能となり、チャット送信にかかるパケット数が少なくなる。

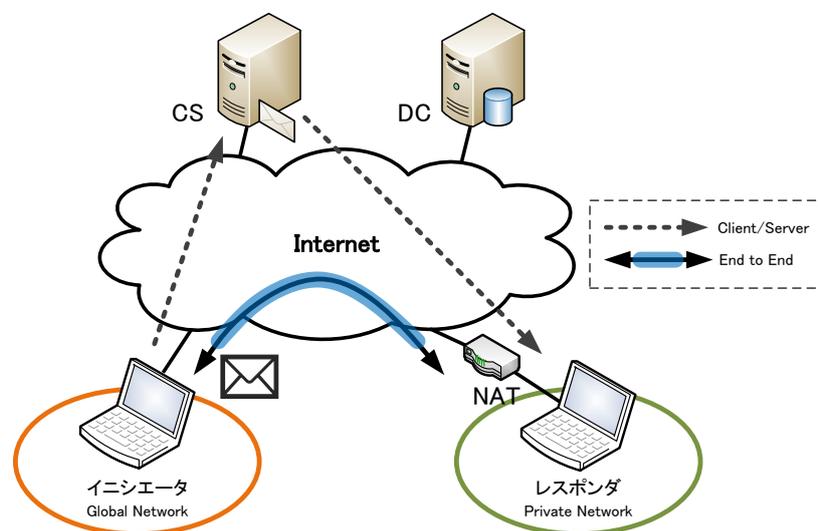


図 6 提案方式の概要

## 4.2 通信方式

図 7 に NTM Mobile を用いたエンドツーエンド通信によるチャットのシーケンスを示す。尚、イニシエータはグローバルネットワーク、レスポンドは NAT 配下のプライベートネットワークに存在していることとする。イニシエータがキャラクタデータを送信しようとする時、NTM Signaling 処理により UDP トンネルが構築される。これにより、レスポンドが NAT 配下に存在していても通信の開始が可能である。また、2 回目以降の通信では NTM Signaling 処理は不要である。

実際にチャットデータを送信する際、送信データがキャラクタデータであれば、単一のパケットの通信で済むため UDP で通信を行う。この場合、正常にチャットデータを受信できたことを確認するため、レスポンドはイニシエータに対してアプリケーションレベルで応答を返す。送信データがファイル（長データ）であれば、確実にデータを送信することができるため TCP で通信を行う。TCP 通信の場合、送達確認は TCP の機能に任せる。

このように NTM Mobile を用いたチャット通信では、キャラクタデータを送信する場合は初回の通信では 9 パケット、2 回目以降の通信では 2 パケットの通信でチャットを行うことができる。NTM Signaling 処理は初回のみ行うだけで良く、実際にチャットを行う場合は数パケットの通信で済むため、クライアントサーバ型チャットに比べてシーケンスを大幅に簡略化できるという特徴がある。

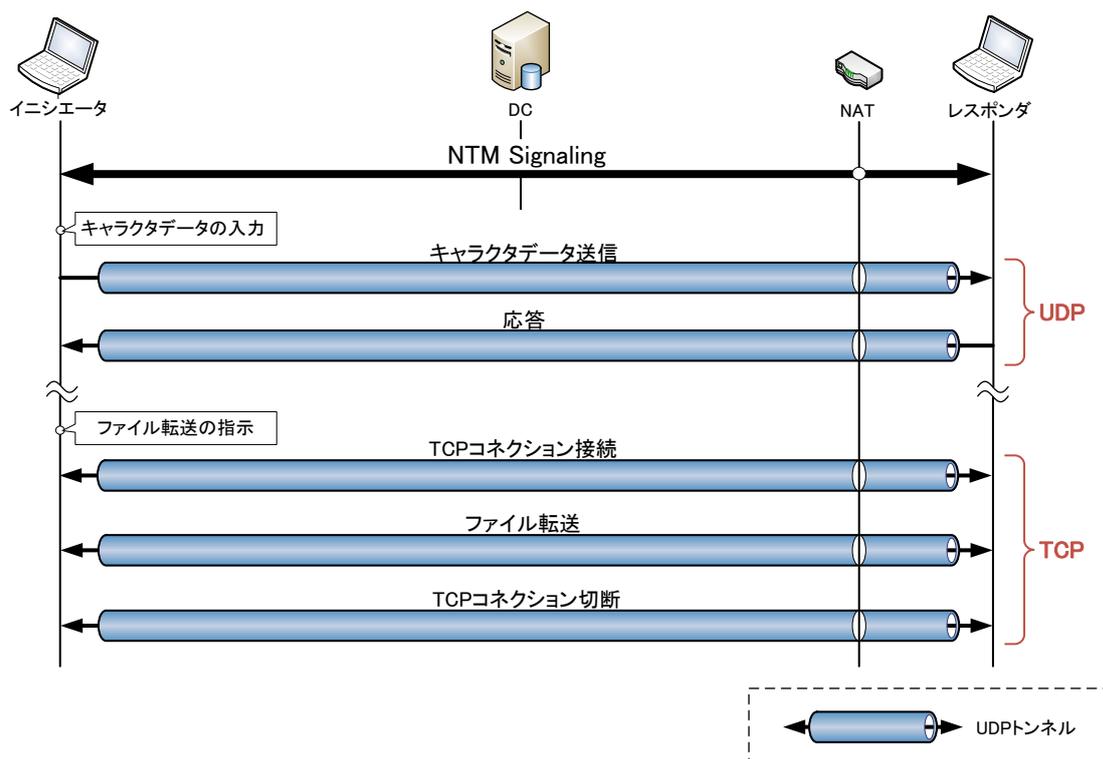


図 7 エンドツーエンド通信によるチャットシーケンス

## 第5章 実装と性能評価

本章では、提案方式の実装とその動作検証、および性能評価について述べる。

提案方式のプロトタイプを作成し、NTM 端末へ実装を行った。動作検証として、提案する NTM-Mobile 上でチャット通信を正常に行われるかどうか確認した。また、提案方式の評価として、NTM-Mobile 上でチャットを行った時の送達時間を測定した。また、提案方式の評価として、チャットを行った時の送達時間を測定した。

### 5.1 実装

図 8 に実装構成を示す。NTM 端末はユーザ空間の NTM デーモンと、カーネル空間の NTM カーネルモジュールにより構成される。NTM デーモンは DC に対する NTM 端末情報の登録と仮想 IP アドレスの取得、およびトンネル構築を行う。NTM カーネルモジュールはパケットのカプセル化/デカプセル化、および暗号化処理を行う。アプリケーションは NTM カーネルモジュールを通して仮想 IP アドレスで通信を行う。アプリケーション部分にチャットプログラムのプロトタイプの実装を行う。

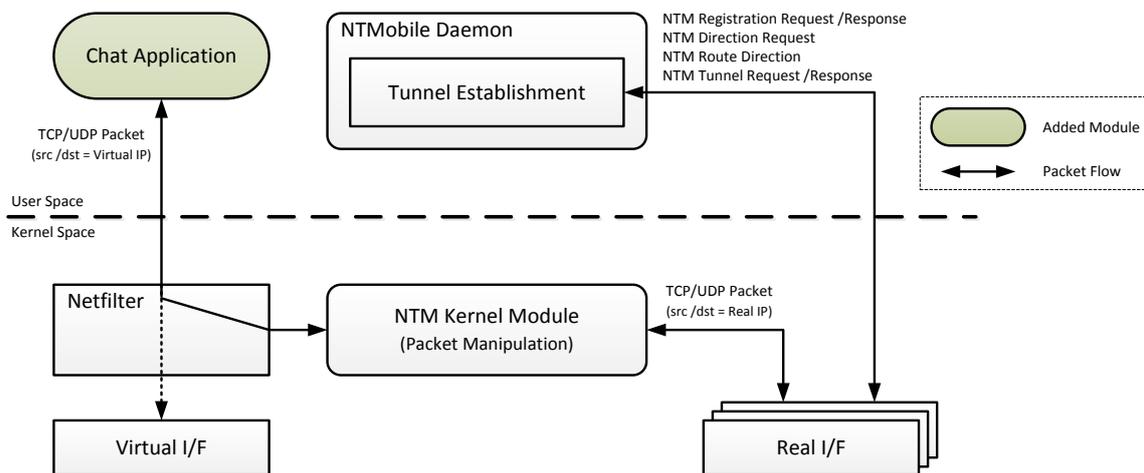


図 8 実装構成

## 5.2 動作検証

図 9 に動作環境におけるネットワーク構成，表 1 にホスト PC の構成，表 2 に仮想マシンの構成を示す。1 台のホスト PC 上に VMware Player<sup>\*1</sup> を用いて，NTM 端末 2 台（イニシエータ，レスポнда）を構築した。それぞれを IPv4 プライベートネットワークに接続した。DC は既に研究室内サーバに構築されているものを利用し，グローバルネットワークに接続されている。この動作環境において，チャットが正常に行われることを確認した。

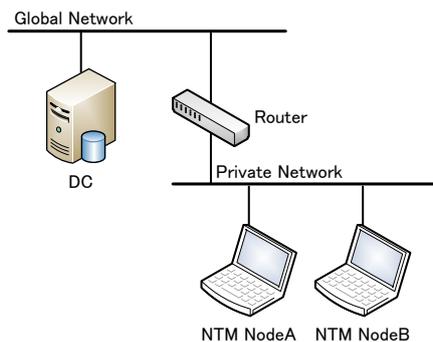


図 9 動作検証におけるネットワーク構成

表 1 ホスト PC の構成

|        | ホスト PC                       |
|--------|------------------------------|
| OS     | Windows7 64bit               |
| CPU    | Intel Core i7-2600 (3.40GHz) |
| Memory | 8.00GB                       |

表 2 仮想マシンの構成

|              | NTM NodeA, NTM NodeB         |
|--------------|------------------------------|
| OS           | Ubuntu 12.04 LTS             |
| Linux Kernel | 3.2.0-97-generic-pae         |
| CPU          | Intel Core i7-2600 (3.40GHz) |
| Memory       | 1GB                          |

<sup>\*1</sup><http://www.vmware.com/jp/>

### 5.3 性能評価

2台の仮想端末（イニシエータ，レスポнда）間でチャットシステムを動作させたときのチャット送達時間を計測し，性能評価を行った．送達時間は Wireshark<sup>\*2</sup> により取得した．

イニシエータがチャットメッセージ送信してから応答が返信されるまでの送達時間を測定した．尚，キャラクタデータは”test”のメッセージを送信した．これを NTMobile 上で動作させた場合と，NTMobile に載せず直接動作させた場合で計測を各 5 回行い，その平均を算出した送達時間を表 3 に示す．

イニシエータがキャラクタデータを送信した場合，直接動作させたときの送達時間は 0.047ms であり，NTMobile 上で動作させたときの送達時間は 1.429ms であった．キャラクタデータを NTMobile 上で送信した場合，僅かな遅延が発生しているが，ユーザの利用には影響無いと考えられる．

表 3 チャット送達時間

|                | Direct    | NTMobile  |
|----------------|-----------|-----------|
| キャラクタデータ (UDP) | 0.047[ms] | 1.429[ms] |

---

<sup>\*2</sup><https://www.wireshark.org/>

## 第6章 従来方式との比較

表 4 にクライアントサーバシステムによるチャット通信方式と提案するチャット通信方式の比較を示す。

- セキュリティ

従来方式では管理者が情報を取得でき、サーバから情報漏洩する懸念がある。それに伴い、業務でチャットを利用することは困難であった。一方で、提案方式ではサーバを利用しないため、サーバから情報漏洩する心配は無い。

- サーバ管理

従来方式ではサーバを利用するため、サーバの障害や二重化等に対する管理が必須である。提案方式ではチャット利用のために必要であった CS を総合的なサーバとして利用できる DC を用いることで、その実用性を含めて△とした。

- トラフィック

従来方式ではチャットデータを送信して相手端末に受信されるまでのシーケンスが複雑であり、メッセージ送信毎に全ての処理を実行しなければならない。提案方式では NTM Signaling 処理は初回のみ行うだけで良く、シーケンスがシンプルになり、従来方式と比較してトラフィックが少なくなる。

表 4 従来方式のチャット通信方式と提案方式の比較

| 比較項目   | 従来方式 | 提案方式 |
|--------|------|------|
| セキュリティ | ×    | ○    |
| サーバ管理  | ×    | △    |
| トラフィック | ×    | ○    |

## 第7章 まとめ

本論文では、NTMobile を用いてエンドツーエンド通信によるチャット通信方式を提案した。提案方式では、NTMobile を用いることにより、エンドツーエンドの UDP トンネルを経由してチャットデータを送信する。この方式により、ユーザはインターネット上の NAT の存在を意識することなくチャットを行うことができ、サーバから情報漏洩する心配が無くなる。また、サーバ管理の負担が減り、トラフィックを軽減することが可能となる。

提案方式のプロトタイプを NTM 端末への実装を行った。動作検証を行った結果、2 台の NTM 端末間でチャットメッセージの送信が正常に完了し、NTMobile 上でチャットが動作することを確認した。



## 謝辞

本研究を進めるにあたり，多大なる御指導と御教授を賜りました，指導教官である名城大学大学院理工学研究科 渡邊晃教授に心から感謝致します。

本研究を進めるにあたり，ご意見並びにご助言を賜りました，名城大学大学院理工学研究科 鈴木秀和助教，愛知工業大学情報科学部情報科学科 内藤克浩助教に感謝致します。

最後に，本研究を進めるにあたり，数々の有益なご助言を賜りました，渡邊研究室および鈴木研究室の諸氏に感謝致します。



## 参考文献

- [1] 鈴木秀和, 上酔尾一真, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃 : NTMobile における通信接続性の確立手法と実装, 情報処理学会論文誌, Vol. 54, No. 1, pp. 367–379 (2013).
- [2] 内藤克浩, 上酔尾一真, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄 : NTMobile における移動透過性の実現と実装, 情報処理学会論文誌, Vol. 54, No. 1, pp. 380–393 (2013).
- [3] 上酔尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃 : IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価, 情報処理学会論文誌, Vol. 54, No. 10, pp. 2288–2299 (2013).
- [4] 納堂博史, 鈴木秀和, 内藤克浩, 渡邊 晃 : NTMobile における自律的経路最適化の提案, 情報処理学会論文誌, Vol. 54, No. 1, pp. 394–403 (2013).



# 研究業績

## 研究会・大会等

- (1) 中村隼大, 鈴木秀和, 内藤克浩, 渡邊 晃 : NTMobile を用いたエンドツーエンド通信によるチャットアプリケーションの提案, 平成 27 年度電気・電子・情報関係学会東海支部連合大会論文集, Sep. 2015.

