インターネット非接続時における複数NAT越え P2P通信方式による情報共有アプリケーションの提案

田中 有彩[1] 前野 誉[2] 高井 峰生[1,3] 大和田 泰伯[4] 小口 正人[1]

概要：現在，インターネットは人々にとって欠かせない生活の一部となっており，重要な情報インフラとして広く普及している．中でも多くの人が通話や電子メール，SNS など，他の人と連絡を取る手段としてインターネットを利用している．そのため，災害時のネットワーク設備の物理的破損による通信不能や，多数の人が一斉に安否を確かめるために発生する輻輳などの通信障害などは，多くの人々に不安と困惑を招いてしまう．そこで本研究では，インターネット非接続時でも接続時のようにプライベートネットワーク間で NAT 越えを実現し，無線 LAN に参加している端末同士がメッセージ・ファイルなどのデータ共有を行えるような情報共有システムの構築を目的とする．本稿では，プライベートネットワーク間で NAT を越えて情報共有を行う通信環境実現性の確認，またそのような環境下で機能するアプリケーション設計の提案を行なった．

A Proposal of Information Sharing Application by P2P Communication System Using Multiple NATs Traversal not on the Internet

ARISA TANAKA[1] TAKA MAENO[2] MINEO TAKAI[1,3] YASUNORI OWADA[4] MASATO OGUCHI[1]

1. はじめに

現在，インターネットは人々にとって欠かせない生活の一部となっており，重要な情報インフラとして広く普及している．中でも多くの人が通話や電子メール，SNS など，他の人と連絡を取る手段としてインターネットを利用している．そのため，災害時のネットワーク設備の物理的破損による通信不能や，多数の人が一斉に安否を確かめるために発生する輻輳などの通信障害などは多くの人々に不安と困惑を招いてしまう．特に地震大国である日本にとって，その影響は大きいと言える [1]．通信障害の主な原因は，基地局や基幹ネットワークである [2]．基地局や基幹ネットワークが災害により破損，機能不全となってしまうとインターネットに繋がらない，もしくは衛星回線等によりインターネット接続回線が非常に細くなることにより，通信による情報共有ができなくなってしまう．そこでインターネット非接続時でも接続時のようにプライベートネットワーク間で NAT 越えを実現し，無線 LAN に参加している端末同士がメッセージ・ファイルなどのデータ共有を行えるような情報共有システムが必要だと考えた．

通常，既存の個人や小規模な組織単位で利用されるネットワークの多くは，プライベートネットワークを構成し，NAT(Network Address Translator) ルータを介してインターネットに接続されている．ここでこれらのプライベートネットワークには何も手を加えず，NAT ルータ同士がインターネットを介さずにアドホックに自律的にネットワークを構成し，接続できるシナリオを想定する．その際，他のプライベートネットワークのノード同士で電話やメッセージ等の P2P 通信を実現する手法を検討する．例としては，緊急車両間のプライベートネットワーク同士や，災害時に臨時に構築した避難所間のプライベートネットワーク同士をアドホックに接続した場合などを想定している．

1 お茶の水女子大学
〒 112-8610 東京都文京区大塚 2-1-1
2 株式会社スペースタイムエンジニアリング
〒 101-0025 東京都千代田区神田佐久間町 3-27-3 ガーデンパークビル 7F
3 UCLA
3532 Boelter Hall, Los Angeles, CA 90095-1596, USA
4 情報通信研究機構
〒 980-0821 宮城県仙台市青葉区片平 2-1-3

解決策として，震災時には NAT ルータに端末が繋がっている環境がいくつも孤立しているため，その孤立したプライベートネットワーク間を繋げ，端末間で情報共有の通信を可能にするため NAT ルータに仕掛けを作り，NAT 越え技術を応用し，孤立したプライベートネットワーク間を繋げられるような仕組みを作るのが現実的に最も有用であると考えられる（図 1）．

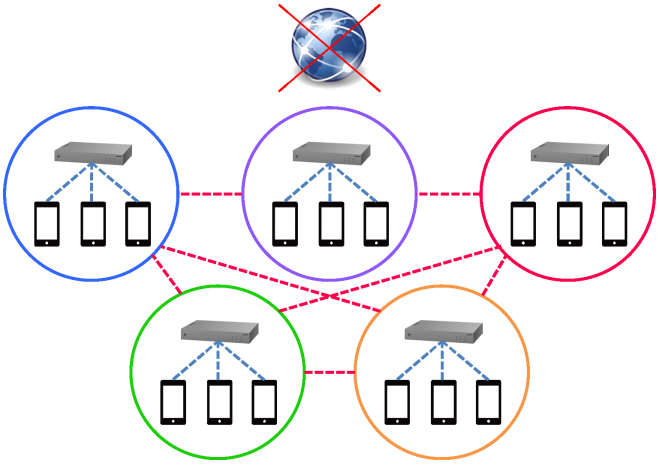

図 1　目標図

2.　想定システムの概要と課題

2.1　概要

想定システムの概要は，災害などによりインターネットが使えず相手と連絡が取れない状況において，無線 LAN に参加している端末同士がメッセージ・ファイルなどのデータ共有を可能にすることである．

具体的には，Wi-Fi AP 兼 NAT ルータ (以降 Wi-Fi ルータと呼ぶ) を用い，そこから無線 LAN を飛ばすことで基地局や基幹ネットワークなどの影響を受けないプライベートネットワークを Wi-Fi ルータごとに構築し，プライベートネットワークを通して通信を行う．つまり Wi-Fi ルータ同士のモバイル・アドホックネットワーク (MANET) を構築する．ここで，MANET とは基地局や固定網に依存せず，移動端末を構成要素とする自律分散形のネットワークである [3]．既存の施設や設備を必要とせずにネットワークを構成できるという利点にも関わらず，なかなか実用化はされていない．これは，MANET を利用した多くのものが，端末側での事前準備を必要とする方式であったことが一因であると言える．そこで通常の MANET は端末同士間でネットワークを構成することが多いが，今回は先ほど述べたように Wi-Fi ルータ同士でネットワークを構成する．つまり，この仕組みを端末自体に作るのではなく，Wi-Fi ルータ自体に作ることで，Wi-Fi ルータ同士の MANET を構築し，端末側に手を加えること無く Wi-Fi ルータを利用して端末同士の通信を可能とする．

しかしプライベートネットワーク下にあるアドレスには外部ネットワークからの直接接続が不可能であるため，NAT 超え技術を利用する．今回の NAT 超え技術として STUN/TURN サーバ・シグナリングサーバを利用する．ここで STUN/TURN サーバ・シグナリングサーバは Wi-Fi ルータ全てに搭載する．どのようにして端末がサーバをアドホックに見つけ出すかは今後の課題である．端末は通信相手を特定するため，STUN/TURN サーバ，シグナリングサーバをアドホックに見つけ出し，NAT 越え通信を実現する．

続いて端末同士で P2P 通信を行うシステムには WebRTC(Web Real-Time Communication) を用いた．これはブラウザ上でビデオ通話などのリアルタイムコミュニケーションを実現するためのフレームワークである．WebRTC を用いることで，プラグインなしでウェブブラウザ間のボイスチャット，ビデオチャット，ファイル共有などが利用可能である．また直接相手と通信する P2P 型通信，NAT 超えを実現する仕組みなどが含まれている．しかしウェブブラウザは全てに対応しておらず，Chrome，Firefox などと限られてはいるが，専用アプリケーションのインストールの不要，大量のデータを高速に送ることができる，通信は DTLS が採用されており暗号化がなされているといった利点を持つ．これにより，無線 LAN に参加している端末同士のメッセージ・ファイルなどのデータ共有を可能にする．

2.2　課題

まず通常のインターネット上での P2P 通信に必要な技術について説明する．

(1) NAT 越え
(2) 通信相手がどのサービスを利用可能かを知る
(3) 相手の ID と IP アドレスを解決する仕組み
(4) 相手がオンラインなのか，オフラインなのかを知る仕組み

この 4 つが必要であり，これらを行うため何らかのサーバ機能を通常はインターネット上で提供している．これらをインターネットに繋がらない環境下で，アドホックに接続されたネットワーク同士でどう実現するかが課題である．現段階では 1 のみ概要で述べたように STUN/TURN サーバを使用することで対応しているが，2・3・4 はまだ課題として残っている．そのため今回は 2 については，全てのノードが共通のサービスを利用可能である前提を置き，相手の ID と IP アドレスを解決する仕組みもできており，両方ともオンラインであると仮定して実験を行なった．まとめると実験としては，Wi-Fi ルータ 2 つを用いてプライベートネットワークを 2 つ作成し，その間での NAT 越えて情報共有を行う通信環境実現性の確認，またそのような

環境下で機能するアプリケーション設計の提案を行なった．また今回は端末が STUN/TURN サーバ・シグナリングサーバを見つけ出せると仮定するため，片方の Wi-Fi ルータに STUN/TURN サーバ・シグナリングサーバを搭載した．実験については第 7 節で述べる．

3. 関連研究

現在，NAT の外部から内部のネットワークへ通信を開始できないという NAT 越え問題に対する研究が多くなされている．

[4] では，外部ノードとホームゲートウェイが連携することにより NAT 越え問題を解決する従来の NAT-f(NAT-free protocol) の誰でもホームネットワーク内の内部ノードにアクセスできてしまうという課題に対し，サービス単位でグルーピングすることにより，外部ノードからのアクセス制御と内部ノードが提供するサービスの制御を同時に実現できることが示されている．

[5] では，DNS サーバと NAT ルータを利用して両者を協調させることにより NAT 越えを実現する方式が提案されている．その提案方式は NTS(NAT-Traversal Support) システムと呼ばれており，DNS サーバを改造した NTS サーバ，NAT ルータを改造した NTS ルータ，実行するプロトコルとして NTS プロトコルが使用されている．また端末の機能追加が必要な NAT-f とは異なり，一般のユーザ端末に機能を追加することなく，かつエンドエンドで NAT 越えを実現できる方式である．

[6] では，STUN では本来対応することができないシンメトリック型 NAT を STUN を拡張することで超えて，NAT の外側から TCP 通信を開始する手法の実装が行われている．

どの関連研究も NAT ルータとは別にサーバを用意することで，NAT 超えを行なっており，またプライベートネットワーク同士での NAT 超えではなくプライベートネットワークと外部のネットワーク間での NAT 超えとなっている．そのため，本研究の特徴とも言える NAT ルータ自体にサーバを搭載するという点・またプライベートネットワーク同士での NAT 超えは非常に有益だと考えられる．

4. P2P (Peer to Peer) 方式

P2P 方式とは，ネットワーク上で対等な関係にある端末間を相互に直接接続し，データを送受信する方式 (図 2) である．また，そのような方式を用いて通信するソフトウェアやシステムの総称でもある．特定のサーバを用意して情報をやり取りするクライアント・サーバ方式 (図 3) などと対比される用語で，利用者間を直接繋いで音声やファイルを交換するシステムが実用化されている．これにより，特定のサーバを介さずに，端末同士の通信を可能にする．

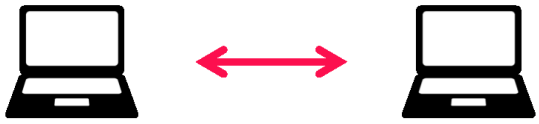

図 2 P2P 方式

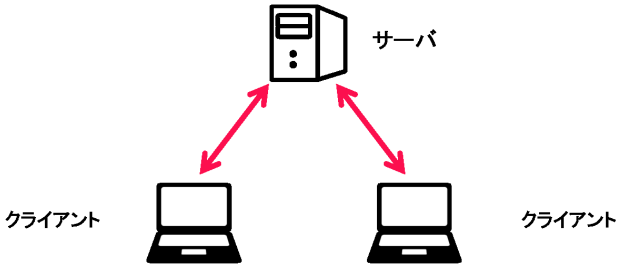

図 3 クライアント・サーバ方式

5. NAT 越え技術と WebRTC

5.1 NAT 越え

多くの端末はプライベートネットワークに所属して NAT 配下にあり，端末から外のネットワークへ通信をすることが通常である．しかし逆に，外から NAT 配下の端末へ直接通信することはできない．

この問題を解決する手法を NAT 越え（NAT Traversal）という．

5.2 STUN と TURN

STUN (Session Traversal Utilities for NATs) とは，NAT 越えの方法の一つである．通信するホストが STUN サーバに UDP 接続を行い，NAT が割り当てたグローバル IP アドレスとポート番号を取得する．NAT の種類にはフルコーン型・制限コーン型・ポート制限コーン型・シンメトリック型と全部で 4 種類ある (表 1)．STUN が対応できる NAT は，フルコーン型・制限コーン型・ポート制限コーン型の 3 つに限られる．

表 1 から分かるように，表が下に行くほど利用制限が厳しくなっている．特に一番厳しいシンメトリック型は STUN で対応できない．そのため，シンメトリック型には TURN(Traversal Using Relay NAT) を使用することで，全ての NAT に対応する．しかしすべての通信を TURN サーバ経由で行うため，P2P 通信ではなくなり，またサーバにも高負荷が掛かってしまう．

表 1　NAT の種類

種類名	NAT が割り当てたポートにアクセスできるインターネット側の端末の制限	利用制限の厳しさ
フルコーン型	どの端末でもアクセス可	緩やか
制限コーン型	NAT 配下の端末がアクセスした端末からアクセス可	厳しめ
ポート制限コーン型	NAT 配下の端末がアクセスした端末とポート番号からだけアクセス可	制限コーン型より厳しい
シンメトリック型	通信元の端末と通信先の端末が 1 対 1 の場合にしか使えない	かなり厳しい

5.3　WebRTC

WebRTC(Web Real-Time Communications) とは，ブラウザでリアルタイムコミュニケーションを実現するための仕組みである．つまり，P2P 通信を利用して端末間の相互接続が可能である．プロトコルには UDP が使われているため，高速な通信ができる．WebRTC による P2P 通信をするためには，シグナリングと ICE(Interactive Connectivity Establishment) が必要である．

シグナリングとは，通信相手の IP アドレス情報やポート番号等の情報を解決する手法である．ICE は後述する NAT 越え通信のためのセッション確立のための情報交換を行うプロトコルである．

5.4　NAT 越えと P2P 通信

WebRTC を用いてプライベートネットワーク内の端末同士の P2P 通信を行うためには， NAT 越えが必要となる．この時，シグナリングと ICE という仕組みが利用される．端末同士が P2P セッションを確立するには，シグナリングにより通信相手の IP アドレスや接続ポート番号等の情報を互いに交換しなければいけないため，どちらの端末からもアクセスできるシグナリングサーバが必要となる．

ICE とは STUN や TURN などの NAT 越えの手順をまとめたものであり，通信できそうな候補を集め，シグナリングにより相手とその候補を交換し，相手との通信を試みる仕組みである．また，通常のネットワークにおける STUN による P2P 通信を図 4 に，TURN による通信を図 5 に示す．

6.　想定環境

災害時の通信環境を想定したものを図 6 に示す．この時，プライベートネットワーク同士が NAT ルータを介してアドホックに接続した際に自律的にシグナリングサーバ，STUN/TURN サーバを一意に検出し，NAT 越えの P2P 通信による情報共有が行える仕組みを備えた Wi-Fi ルータを用いることを想定している．

災害時においてインターネットに繋がらない状況でも，

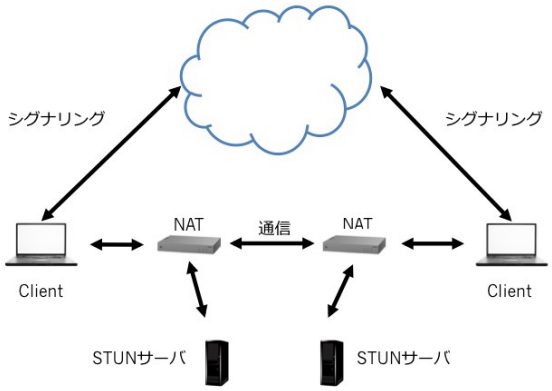

図 4　STUN による P2P 通信

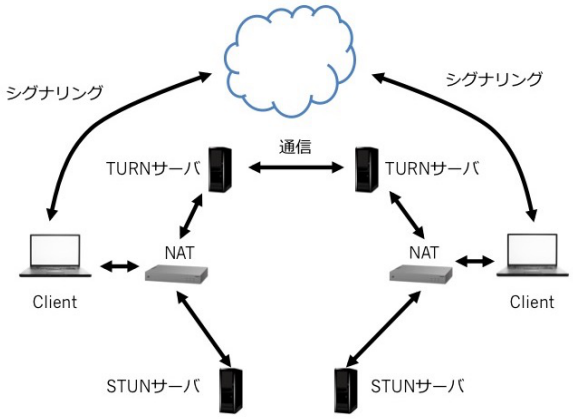

図 5　TURN によるサーバ経由通信

この NAT ルータの Wi-Fi につなぐことで，Wi-Fi に繋がっている端末同士が情報共有を行うことが可能になる．本研究ではこのような通信環境を想定して検討を行った．

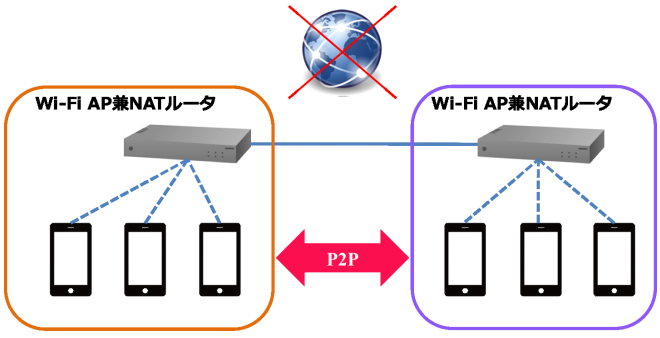

図 6　想定環境

7.　ローカル環境における NAT 越え実験

7.1　動作の流れ

本研究での端末の動作の流れを図 7 に示す．これには NAT 越え技術と P2P 通信技術が使用されている．動作は以下の通りとなっている．

1)　相手と通信するには相手の外から見た IP アドレスが

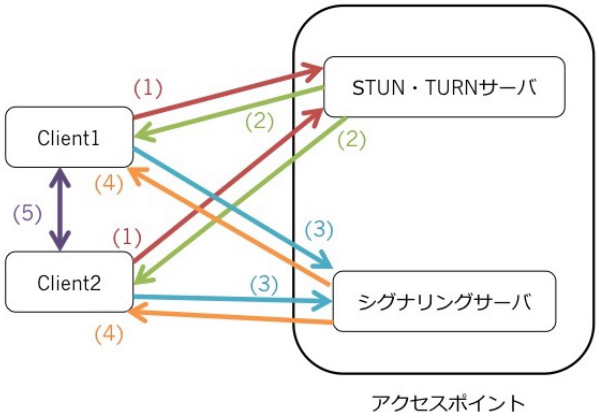

図 7 NAT 越えと P2P 通信の処理

必要であるため，STUN・TURN サーバに自分の外から見た IP アドレス情報を問い合わせる．

2) 自分の外から見た IP アドレス情報と様々な相手との経路情報を獲得．これにより，NAT 越え問題を解決．

3) 相手と通信したいことを通知．

4) 相手の情報や通信経路情報を交換．

5) P2P 通信または TURN による通信の開始．

7.2 実験環境

本研究では，WebRTC のオープンソースソフトウェアである easyRTC， STUN 兼 TURN サーバが構築できるオープンソースソフトウェアである coturn を用い，実験用のローカル環境を構築した．その環境を図 8 に示す．これにより，ローカル環境における NAT 越え情報共有の実験を行う．この時プライベートネットワーク同士が NAT ルータを介してアドホックに接続した際に，自律的にシグナリングサーバ，STUN/TURN サーバを一意に検出できると仮定している．

また easyRTC は WebRTC のビデオ・オーディオ・データアプリケーションなどが利用でき，Node.js の WebSocket 実装である Socket.io で構築されたシグナリングサービスを使用している．

無線 LAN を搭載した Debian GNU/Linux8 ubilinux サーバ 2 台にそれぞれ hostapd をインストールし，Wi-Fi ルータとして動作させた．このルータの仕様を表 2 に示す．ESSID は AP(COM9) が”test-p2p”，AP(COM10) が”test-nat”となっている．これにより 2 つの異なるプライベートネットワークを作成した．この時の Wi-Fi の仕様を表 3 に示す．

Client としては PC2 台をそれぞれ別の Wi-Fi AP に繋げて設置した．IP アドレスは Client1 が 192.168.42.100，Client2 が 192.168.50.100 となっている．

サーバとして，アクセスポイント (COM9) のイーサネットである IP アドレス 192.168.1.144 のポート番号 8080 にシグナリングサーバである easyRTC を，ポート番号 3478 に STUN 兼 TURN サーバである coturn を構築した．またシグナリングサーバにおける STUN/TURN サーバの設定は，IP アドレス 192.168.1.144 のポート番号 3478 とした．2 台の AP とサーバは有線 LAN でつなぎ，同じネットワーク内となっているため，ネットワークは全部で赤色と青色と緑色の 3 つとなっている．そしてプライベートネットワーク間で情報共有ができるかどうかを試す．今回は Client1 から Client2 へ test.zip という 2MB の ZIP ファイル送受信を行う．

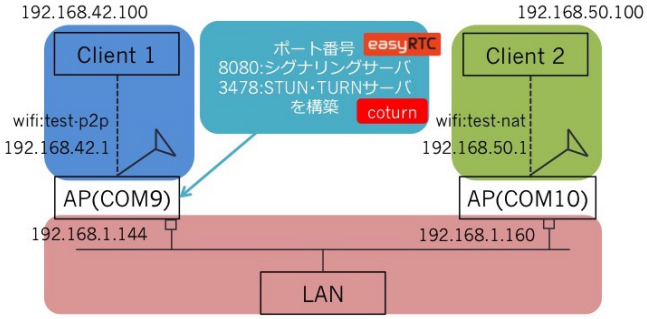

図 8 実験環境

表 2 Wi-Fi ルータの仕様

CPU	Genuine Intel(R) CPU 4000@500 MHz
Storage	Compact Flash 6GB
OS	Debian GNU/Linux8 ubilinux
WirelessLAN	IEEE 802．11n

表 3 Wi-Fi の仕様

SSID	プロトコル	セキュリティの種類	ネットワーク帯域	ネットワークチャンネル
test-p2p	802.11n	WPA2-パーソナル	2.4GHz	7
test-nat	802.11n	WPA2-パーソナル	2.4GHz	4

7.3 実験結果

Client1 における Wireshark によるパケットキャプチャを図 9・図 10 と図 11 に，Client2 におけるものを図 12 と図 13 に示す．

図 9 から，STUN プロトコルで Client1 とサーバ間で Binding Request が互いに送られていることが分かる．また互いに Binding Success Response とあるように，成功していることが分かる．これにより，Client1 である 192.168.42.100 とサーバである 192.168.1.144 の間で STUN による P2P 通信が可能になる．このことは次の図から見て分かる．図 10 から，プロトコル DTLS で Client1 とサーバが Application Data のやり取りをしていることが分かる．これは先ほど述べたように，STUN による Binding Request の成功により，P2P 通信が可能となったためである．図

11 は図 10 のキャプチャの 1 つをより詳しく見たものである．これによると，Src:192.168.42.100 Src Port:50909 と Dst:192.168.1.144 Dst Port:57775 でのやり取りであることが分かる．これは図 9 の 3, 4 行目の info から分かるように，Binding Success Response による IP アドレスとポート番号が使用されている．

続いて Client2 について見ていく．図 12 から，Client2 とサーバ間で TURN による情報共有が行われていることが分かる．このときプロトコルが STUN となっているが，これは Wireshark の仕様による表記であり，本来は TURN プロトコルが使用されている．また図 13 から，Client2 の Peer，つまり通信相手が Client1 の IP アドレス 192.168.42.100 であることが分かる．つまり，Client2 の通信相手は Client1 であることが分かる．以上のことから Client1 から Client2 への情報共有に成功していることが分かる．このことよりローカル環境における 2 つのプライベートネットワークを接続した NAT 越え情報共有ができていることが確認できた．

Source	Destination	Protocol	Info
192.168.42.100	192.168.1.144	STUN	Binding Request user: cA3u:vN5T
192.168.1.144	192.168.42.100	STUN	Binding Request user: vN5T:cA3u
192.168.42.100	192.168.1.144	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.1.144:57775
192.168.1.144	192.168.42.100	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.42.100:50909

図 9 Client1 におけるパケットキャプチャ 1

Source	Destination	Protocol	Info
192.168.42.100	192.168.1.144	DTLSv1.2	Application Data
192.168.42.100	192.168.1.144	DTLSv1.2	Application Data
192.168.1.144	192.168.42.100	DTLSv1.2	Application Data
192.168.42.100	192.168.1.144	DTLSv1.2	Application Data
192.168.42.100	192.168.1.144	DTLSv1.2	Application Data
192.168.42.100	192.168.1.144	DTLSv1.2	Application Data
192.168.42.100	192.168.1.144	DTLSv1.2	Application Data

図 10 Client1 におけるパケットキャプチャ 2

```
Internet Protocol Version 4, Src: 192.168.42.100, Dst: 192.168.1.144
User Datagram Protocol, Src Port: 50909 (50909), Dst Port: 57775 (57775)
```

図 11 Client1 におけるパケットキャプチャ 3

Source	Destination	Protocol	Info
192.168.1.144	192.168.50.100	STUN	ChannelData TURN Message
192.168.1.144	192.168.50.100	STUN	ChannelData TURN Message
192.168.1.144	192.168.50.100	STUN	ChannelData TURN Message
192.168.1.144	192.168.50.100	STUN	ChannelData TURN Message
192.168.1.144	192.168.50.100	STUN	ChannelData TURN Message
192.168.50.100	192.168.1.144	STUN	ChannelData TURN Message

図 12 Client2 におけるパケットキャプチャ 1

Source	Destination	Protocol	Info
192.168.50.100	192.168.1.144	STUN	Send Indication XOR-PEER-ADDRESS: 192.168.42.100:50909
192.168.1.144	192.168.50.100	STUN	Data Indication XOR-PEER-ADDRESS: 192.168.42.100:50909

図 13 Client2 におけるパケットキャプチャ 2

7.4 考察

これらの結果からプライベートネットワーク間での情報共有は，Client1 とサーバ間では STUN が成功し，P2P 通信がなされており，Client2 とサーバ間では TURN による中継通信となっていることが分かった．これは Client2 が多段 NAT となっているからだと考えられる．STUN はその特性から NAT の最も外側から見た（すなわちサーバ側から見た) アドレスしか知り得ないため，多段 NAT に対応することができない．またその場合，TURN サーバがピア間のパケットを中継することによってピア間の通信を実現されている．以上のことから Client1 では P2P 通信，Client2 では多段 NAT となってしまっているため，TURN による通信になったと考えられる．

8. システム概要

WebRTC によって，無線 LAN に参加している端末同士がメッセージ・ファイルなどのデータ共有を可能にする．

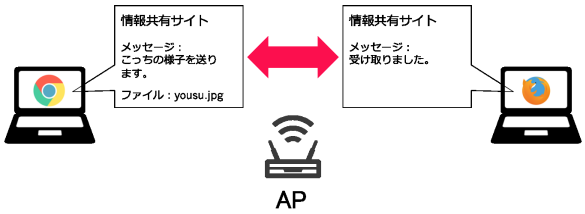

図 14 システム図

現在の実装は，

- メッセージのやり取り，ファイル送受信が別々のページで利用可能
- 意味を持たない英数字の羅列の ID を発行
- 1 対 1 での通信

となっているため，今後は

- メッセージとファイル送受信が同じページで利用可能に
- 相手とわかるような ID もしくはわかるように通信
- 1 対多の通信に対応
- リアルタイム以外にも対応
- ブラウザの制約をなくす

といった改善をしていく．

8.1 使い方

(1) AP から出ている無線 LAN に接続
(2) ブラウザ（Chrome または Firefox）を開き，サイトを開く
(3) 表示される ID 一覧の中から，送信相手の ID を選び，コネクションを押す
(4) テキスト入力・データ送受信

9. まとめと今後の課題

今回は最終目標に対し，プライベートネットワーク同士が NAT ルータを介してアドホックに接続した際に自律的にシグナリングサーバ，STUN/TURN サーバを一意に検出できると仮定した上で Wi-Fi ルータを 2 つ用いてプライベートネットワーク 2 つ間における STUN/TURN を用いた NAT 越え情報共有を行う通信環境実現性の確認，またそのような環境下で機能するアプリケーション設計の提案を行なった．

異なるプライベートネットワーク同士だと TURN サーバを経由しての通信ではあったが，情報共有はできることが確認できた．またシステムには第 8 節で述べたように多くの改善点があり，今後使いやすいようより完成に向け実装を行う．

今後は，システムの実装を行い，プライベートネットワーク同士が Wi-Fi ルータを介してアドホックに接続した際に自律的にシグナリングサーバ，STUN/TURN サーバを一意に検出し，NAT 越え通信による情報共有が行える手法を検討する．具体的には，これらには Dynamic DNS, XMPP や SIP といったプロトコル及びサーバを利用していくことを考えている．

謝辞

本研究の一部はお茶の水女子大学と情報通信研究機構との共同研究契約に基づくものである．

参考文献

[1] 内閣府防災情報．”首都直下地震の被害想定と対策について” http://www.bousai.go.jp/jishin/syuto/taisaku_wg/pdf/syuto_wg_report.pdf，2017 年 4 月参照．

[2] 中村 功．”大規模災害と通信ネットワーク –東日本大震災に思う–” http://nakamuraisao.a.la9.jp/CIAJ.pdf，2017 年 4 月参照．

[3] 間瀬憲一”モバイル・アドホックネットワーク”，シンポジウム (47)，pp.13-26，2002 年 3 月．

[4] 鈴木秀和，渡邊晃” 通信グループに基づくサービスの制御が可能な NAT 越えシステム”，マルチメディア，分散，協調とモバイル (DICOMO2009) シンポジウム，pp.372-378，2009 年 7 月．

[5] 宮崎悠，鈴木秀和，渡邊晃” 端末に依存しない NAT 越えシステムの提案と実装”，マルチメディア，分散，協調とモバイル (DICOMO2008) シンポジウム，pp.587-592，2008 年 7 月．

[6] 黒田隼之輔，中山泰一” TCP における STUN を用いた対称型 NAT 越え手法の実装と評価”，情報処理学会全国大会講演論文集，73rd，p.3.421-3.422，(2011)．

[7] easyRTC https://easyrtc.com/，2016 年 10 月参照．

[8] coturn https://github.com/coturn/coturn，2016 年 9 月参照．

[9] 本橋 史帆，高井 峰生，黒崎 裕子，小口 正人: 「サーバ機能付き Wi-Fi AP を利用したファイル共有方法の提案と実装」，DICOMO2015，2G-1，2015 年 7 月．

[10] udonchan．”WebRTC のデータチャネル解説” Qiita. http://qiita.com/udonchan/items/7f5ffa9e8982ae1636c3，2017 年 4 月参照．

[11] massie_g．”WebRTC の簡易シグナリング” Qiita. http://qiita.com/massie_g/items/f5baf316652bbc6fcef1，2017 年 4 月参照．

[12] ”Real time communication with WebRTC” Google Developers. https://codelabs.developers.google.com/codelabs/webrtc-web/index.html?index=..%2F..%2Findex#0，2017 年 4 月参照．