修士論文　　2008年度 (平成20年度)

同報通信環境における
探索的コミュニケーションアーキテクチャの構築

慶應義塾大学大学院 政策・メディア研究科

遠峰 隆史

修士論文要旨 2008 年度 (平成 20 年度)

同報通信環境における
探索的コミュニケーションアーキテクチャの構築

インターネットを用いた相互通信アプリケーションとして，グループコミュニケーションを実現する会話型アプリケーションが頻繁に利用されている．これらの会話型アプリケーションでは，ユーザー状況を管理するインターネット上のサーバーに接続し，あらかじめ登録されたユーザーの状況を逐次把握する必要がある．これより，実世界における会話コミュニケーションとは異なる環境が必要となり，グループコミュニケーションを通信に抽象化することは困難と考えられる．

本研究では，サーバーに依存しないグループコミュニケーションのための同報通信手法，及び，そのための個人特定技術を実現した．実世界で行う距離的な範囲を想定すべきコミュニケーションに対して，本研究ではその概念を「共有空間」として通信環境に適した形に抽象化した．

本研究における同報通信手法の実現により，外部接続環境の無い電車の中や，限られた空間におけるアドホックなネットワーク，外部接続環境が整ったインターネットまでの幅広い環境においてグループコミュニケーションが可能となった．個人特定技術は，PGP を用いた信頼の輪モデルを採用した．評価として,「共有空間」に近いコミュニケーション手法との比較を行った．

キーワード

1．ネットワークコミュニケーション, 2．マルチキャストネットワーク,
3．個人特定, 4．分散環境, 5. インターネット

慶應義塾大学大学院 政策・メディア研究科

遠峰 隆史

Abstract of Master's Thesis

Academic Year 2008

Design of Explorative Communication Architecture in a Multicast Network Environment

Group communications like interactive communication applications are very common as the Internet applications. These applications requires to access a central server that maintains users who can communicate each other, and to verify each condition sequecially if it is active or not. Accourding to this thought, abstraction of the human interactive communication to the group communication over the network is challenge.

We proposes to create a group communication without requiring a central server sequencially, but to enable a personal identification method. While the human communication generally has the assumption of physical range, our proposed method provides the concept as a "communication domain".

We implemented a group communication method that enables to create a communication environment, not only the Internet but even without an external communiction gateway like inside a train or an adhoc network. The personal identification method is realized by a chain of trust model using PGP. We evaluated our proposed method by using qualitative approach that showed the effort of the use of "communication domain".

Keywords :
1. Network Communication, 2. Multicast Network, 3. Personal Identification,
4. distributed environment, 5. Internet

Keio University Graduate School of Media and Governance

Takashi Tomine

目 次

図 目 次

表 目 次

第1章　序論

本章では，本研究の背景として実世界のコミュニケーションと，インターネット上のコミュニケーションの対比と，そのコミュニケーションへの要求，本研究の目的について述べる．

1.1　背景：実世界のコミュニケーションとインターネット上のコミュニケーション

実世界において，人と人とのコミュニケーションは，共有空間における呼びかけによって行われている．この共有空間に対しての呼びかけは，伝わる範囲を発生する音量を無意識に調節しながら行われている．目の前にいる相手との1対1のコミュニケーションを行う場合には，周りの環境に合わせたそれほど大きくない音量で，同じ空間にいる複数人とのコミュニケーションは，自分と関連のあるグループ内に聞こえる程度の音量で各々がコミュニケーションを取っている．これは，実世界では普通に行われているコミュニケーションである．図1.1に，実世界におけるコミュニケーションのモデルを示す．

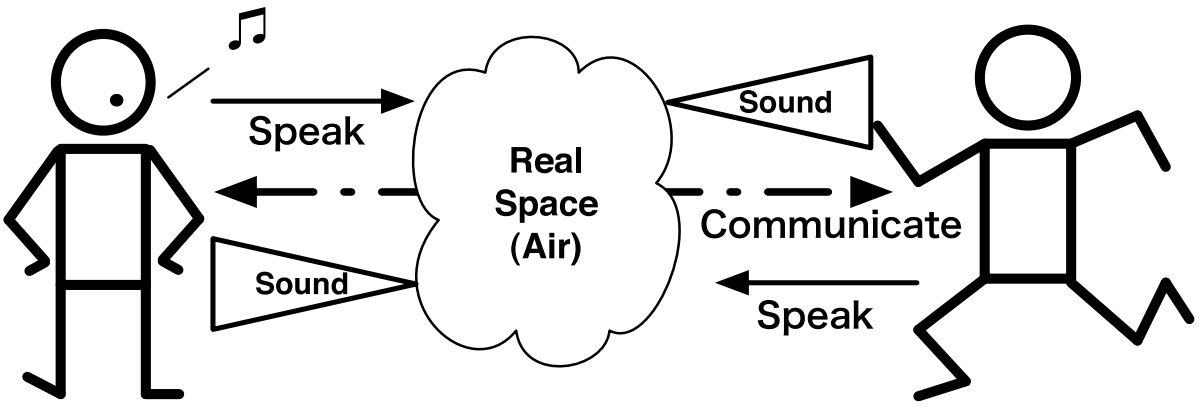

図 1.1: 実世界におけるコミュニケーション

インターネットにおいても，Windows Live Messenger や Skype などのアプリケーションを利用して，人と人とがコミュニケーションを行うことが一般的になってきた．これらのアプリケーションは，ディレクトリサーバや P2P ネットワークを用いてコンタクトリストを作成し，コミュニケーションを取りたい相手を指定し，サーバ経由あるいは直接，Unicast を用いてコミュニケーションを行う．これらのアプリケーションにおいて多人数

でコミュニケーションを行う場合は，Unicast を多重に利用することによって実現している．そのため現状のアプリケーションでは，処理能力などの問題から多人数間でのコミュニケーションには最大接続人数に制限が設けられていることが多い．

このように，インターネットにおけるコミュニケーションを提供する多くのアプリケーションは，インターネットのアーキテクチャ上の理由から 1 対 1 のコミュニケーションを対象としていて，多人数での共有されたコミュニケーションを行う手法はそれほど多くない．

このような中で，インターネットにおける多人数で共有された空間におけるコミュニケーションに対する要求は高まってきている．最近では，Twitter[1] という Web 上で個々のつぶやきを共有するようなシステムが多くの利用者を集めている．また，Nintendo DS のピクトチャット [2] といったような，アドホックに構成されたネットワーク内に仮想的に構成されたチャットルームにおいて，文字や手書きの絵によるコミュニケーション環境が実現されている．

インターネットのインフラストラクチャを利用する上で，多くの場合，人が用いるノードは IP アドレスなどの設定を行う必要がある．これは，インターネットアプリケーションの多くが，宛先として直接相手の IP アドレスを指定する Unicast を用いてコミュニケーションを行っているためである．しかし，インターネットアーキテクチャとしては IP Broadcast や IP Multicast といった，直接相手の IP アドレスを指定せずに通信を行う手法が用意されている．これらの IP Broadcast や IP Multicast は，一定のネットワークの範囲に対して一斉に情報の送信を行い，受信者が自分で取得する情報を選択し，利用するといった仕組みで動いている．そのため，各ノードにインターネットに到達可能な IP アドレスが付いていなくても，自動的に付与される IPv6 link-local アドレスなどが付いていれば，通信を行うことができる．また，近い距離での共有空間の創出は，ほとんどのノードに無線 LAN 機器が搭載されているので，モバイルアドホックネットワークを構成することにより可能である．

1.2 目指す世界

インターネットにおけるコミュニケーションは，1.1 節で述べたように基本的に 1 対 1 の Unicast コミュニケーションにより行われている．一方，実世界では共有空間における会話がコミュニケーションの基本となっている．そこで，インターネット上にアマチュア無線や特定小電力無線などで用いられている共有されたコミュニケーション空間を構成する．また，インターネット上に構成された共有コミュニケーション空間において，文字，音声，映像などといった様々なメディアを用いたコミュニケーションを行うことができる様にする．これにより，今までインターネット上では行われていなかった新しいコミュニケーションの場を創出し，その上で新しいコミュニケーションが行われることを支援する．

1.3 本研究の目的

本研究では，信頼関係にある人同士がコミュニケーションを行う空間を共有空間と定義し，実世界で行われている共有空間におけるコミュニケーションをインターネット上で実現するためのアーキテクチャを構築する．インターネット上における様々な制約を踏まえ，その上で共有空間を創出するためのプロトコルを提案し，その上で，自律的に共有空間が人間関係に適応する環境を構築する．

1.4 論文の構成

本論文は 7 章から構成される．第 2 章において，現在インターネット上で行われているコミュニケーションをサービスごとに述べ，その上で，現状のサービスおよび技術における問題点をまとめる．第 3 章において，第 2 章で述べた問題点に対する解決手法を，利用する技術を上げながら述べる．第 4 章において，第 3 章で述べた解決手法を実現するための，本システムの設計について述べる．第 5 章で設計に基づいた実装に関して述べる．第 6 章で本システムの実装に対する評価を行い，既存の問題を解決したか否かを述べる．第 7 章で本研究のまとめ，及び今後の展望を述べる．

第2章　インターネット上のコミュニケーション技術とその課題

本章では，既存のインターネット上で行われているコミュニケーションサービスとその技術を挙げる．その上で，実世界で行われているコミュニケーションとインターネット上で行われているコミュニケーションの違いを述べ，新しいコミュニケーションに対する要求と現状のインターネット上におけるコミュニケーションモデルとの差異について述べる．

2.1　既存のインターネット上のコミュニケーション

インターネットを用いてコミュニケーションを行うことは一般的になっている．現在，インターネットを用いたコミュニケーションは，文字，音声，映像といった様々なメディアを用いて行われている．また，それぞれのメディアは様々な形で提供されたサービスを用いてコミュニケーションに利用されている．

- サーバベースのコミュニケーション
- P2P コミュニケーション
- Point-to-Point コミュニケーション

本節では，現在のインターネットで行われているコミュニケーションを以上のように分類し，それぞれのモデルがどのように実現されているかを述べる．

2.1.1　サーバベースのコミュニケーション

現在，インターネット上で行われるコミュニケーションで多く用いられている手法が，通信相手をディレクトリーサーバに問い合わせ，ディレクトリサーバから得た情報を基に，直接あるいはサーバ経由でコミュニケーションを行う，サーバベースのコミュニケーション手法である．

この手法の代表的な例として，Windows Live Messenger[3] が挙げられる．この手法では基本的に，アカウント情報や各ユーザの状態やアドレスなどの情報はディレクトリサーバによって管理されている．メッセージのやりとりも基本的にはサーバ経由で行われる．サーバへの到達生がなくなったり，サーバがダウンしてしまうとサービスを利用することができなくなる．

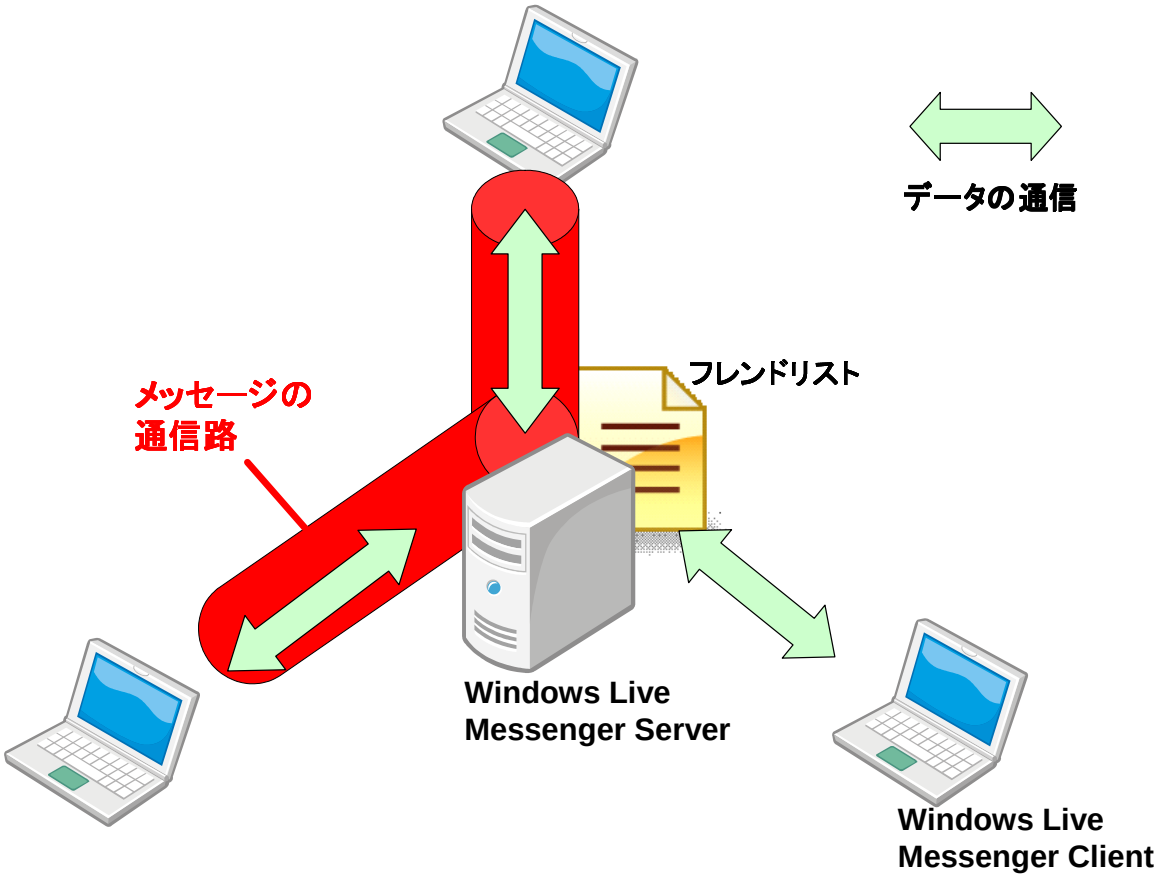

図 2.1: Windows Live Messenger のコミュニケーションモデル

また，インタラクティブな Web ベースのコミュニケーションサービスも，サーバベースのコミュニケーションの一部として考えることができる．

2.1.2 P2P コミュニケーション

近年，インターネット上の新しいコンピュータネットワークの形態として，P2P(Peer-to-Peer) が利用されるようになってきた．インターネット上のコミュニケーションの手法として，P2P を利用する物も現れている．

この手法の例として，Skype[4] が挙げられる．Skype はユーザのディレクトリ情報は P2P 網内で管理されている．また，NAT 下のノードへのアクセス補助は P2P 網内のスーパーノードを利用して行われる．しかし，ログイン時などの各々のアカウントの管理，認証は，Skype のサーバによって提供される．また，P2P 網の初期ノード情報もサーバから取得する．

それ以外の P2P 網は，初期ノード情報をあらかじめ持っておかなければならない．あらかじめ提供されている初期ノードへの到達性が失われると新規ノードが P2P 網に参加することが困難になる．

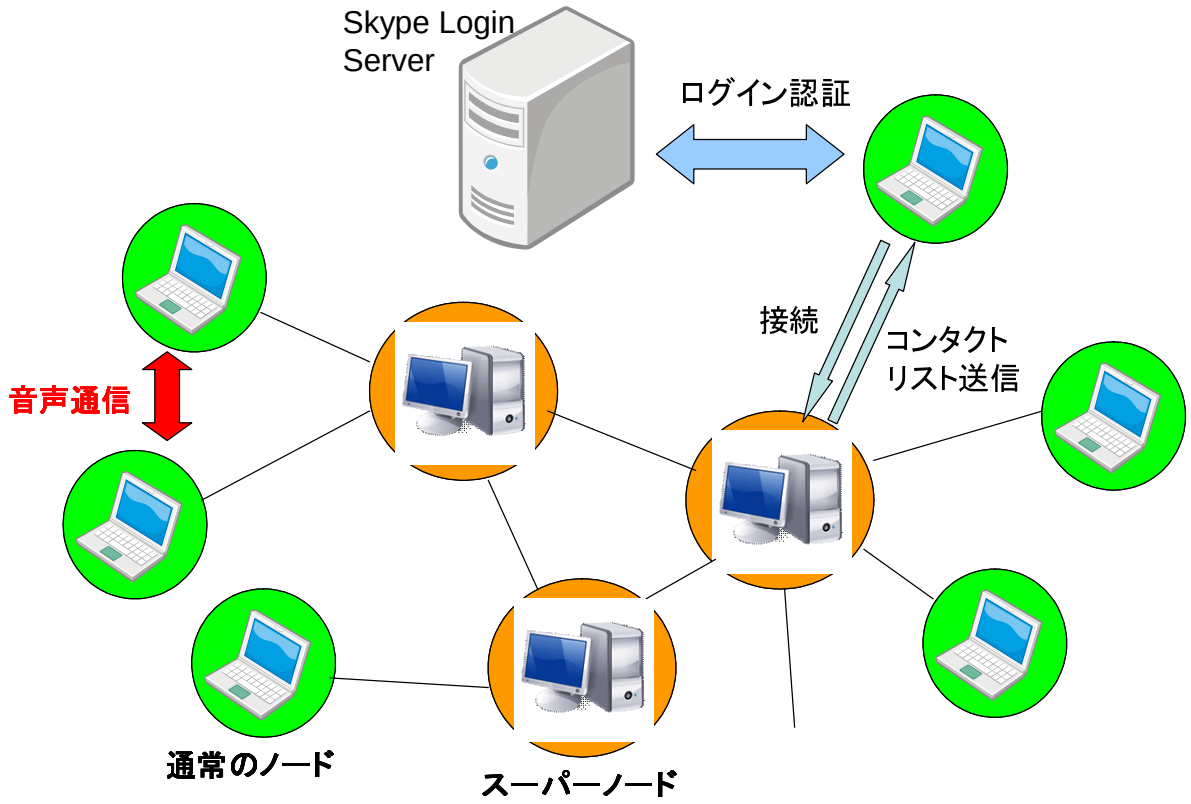

図 2.2: Skype のコミュニケーションモデル

2.1.3 Point-to-Point コミュニケーション

サーバベースおよび P2P コミュニケーション以外に利用されているコミュニケーションは，Point-to-Point コミュニケーションである．この手法は，コミュニケーションを行うユーザ同士が，直接接続先を指定してコミュニケーションするモデルである．手法としては一番基本的なモデルであるが，現在でも Polycom[5] や DVTS[6] などといった主に高品質映像コミュニケーションシステムではよく利用されている．

この手法では，コミュニケーションを行うユーザ同士が，接続を行う相手先の IP アドレスなどの情報をあらかじめ別の手段で交換する必要があるので，インターネットに関する知識や別途コミュニケーション手段が必要になる．

2.2 実世界のコミュニケーション

実世界における人と人とが直接会話を行うコミュニケーションの多くは，直接対面する人に対して声を発することによって行われる．これは，1 対 1 の会話から多対多の会話まで人数にかかわらず同じように行われている．人は会話の際に，無意識に会話の対象を認識し，声の音量を調節することで会話の範囲を限定して，現在話している人物が誰である

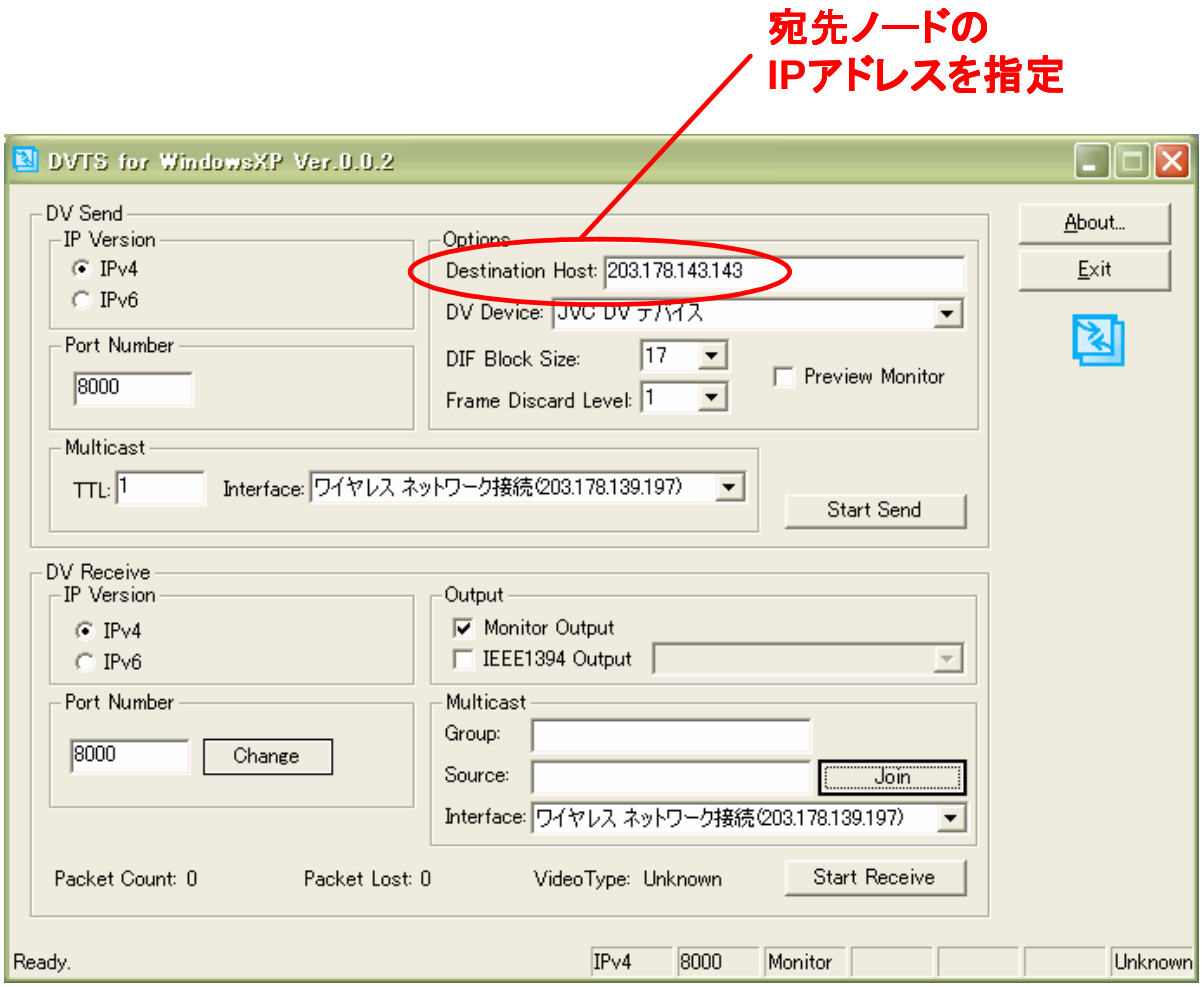

図 2.3: DVTS

かを判断する．また，近隣の会話に対しても無意識に取捨選択され，自分に関係のない会話や自分と関係のない人物の会話は聞き取らず，自分に関係のある会話は聞き取り新たに会話に参加する．

このように，実世界におけるコミュニケーションでは，インターネット上でコンピュータに支援されたコミュニケーションと異なり，会話が行われる範囲がその場その場で変化している．このようなコミュニケーションは現状のインターネットのサービスモデルやコミュニケーションアーキテクチャでは実現が困難である．

2.3 新しいコミュニケーションに対する要求

インターネット上においても，実空間で行われているコミュニケーションのように，コミュニケーションを行う相手同士が同じ空間を共有し，そこでの発言をお互いに共有するという要求が顕在化してきている．本節では，そのような要求に応じて現れた現在提供さ

れているサービスとして，Twitter とピクトチャットを挙げる．

2.3.1 Twitter

空間を共有するモデルを Web で実現したものとして Twitter が挙げられる．Twitter は，Twitter 社が開発し 2006 年 7 月に運営を開始した Web サービスである．2008 年現在，無料でサービス提供されている登録制 Web サービスとして運営されている．各ユーザは登録後，専用のスペースを提供される．

Twitter では，各ユーザは「What are you doing?（今，何しているの？）」という質問に関する回答を書き続ける．各回答は 140 文字以内のコメントという形で投稿される．この「今，何をしているの？」に対する回答は「つぶやき」と呼ばれている．

Twitter ではユーザがユーザを follow(フォロー) する事ができる．フォローを行ったユーザは，フォロー先ユーザの「つぶやき」を Web 上または IM(Instant Messenger) 経由で受信できる．また，各「つぶやき」には専用 URL が割り当てられるため，過去の「つぶやき」をアーカイブできる．

Twitter では，このフォローの関係を元に「コミュニケーション空間」が構成されている．この「コミュニケーション空間」は，主に同じ興味や実世界での知人関係によって構成されている．

Twitter は Web 上での「空間」を実現しているため，Twitter 社が用意しているインターネットを経由した Web サーバへの到達性が要求される．また，一箇所のサーバに負荷が集中する単一障害点になっている．2007 年及び 2008 年時点では，頻繁にサーバが利用不能になり，ユーザがサービスを利用できない状態が多発している．

2.3.2 ピクトチャット

無線機器を有するモバイルデバイスを利用し，近距離コミュニケーションを実現する例としてピクトチャットが挙げられる．ピクトチャットは Nintendo DS 上で動作するアプリケーションである．ユーザは Nintendo DS に内蔵された IEEE 802.11 機能を使いながら，絵や文字をやり取りしながら「チャット」を行う．

絵は Nintendo DS のタッチスクリーンによって入力し，他のユーザが描いた絵を改変することもできる．Nintendo 社の Web サイトでは，絵を使ったしりとりや、友人同士で 4 コマ漫画を描くなどの遊び方が紹介されている．

ピクトチャットは，Nintendo DS 上でのみ動作するため利用可能プラットフォームが限られている．また，同時にプレイできる人数も最大 16 人である．

2.4 まとめ

本章では既存のインターネット上におけるコミュニケーションモデルとその問題点について述べ，検証を行った．また，実世界におけるコミュニケーションと新しいコミュニ

図 2.4: Twitter

ケーションに対する要求をまとめた．

次章ではこれまで述べてきた事柄を元に，現在満たされていない，インターネット上における新しいコミュニケーションアーキテクチャを提案する．

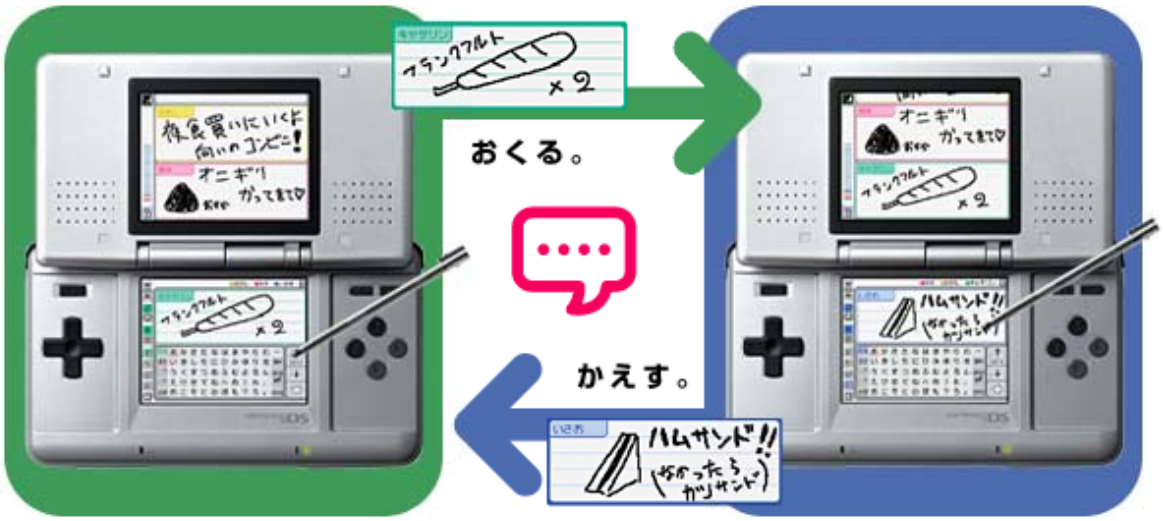

図 2.5: ピクトチャット

第3章　同報通信環境におけるコミュニケーションアーキテクチャの提案

本章では，インターネット上における共有空間におけるコミュニケーションのためのアーキテクチャを提案する．

3.1　インターネット上の共有空間におけるコミュニケーションの実現

1.1 節で述べたように，実世界で行われているコミュニケーションは，共有された空間へ発話し，また，目や声で相手を特定することで実現している．そのため，インターネット上に同じような環境を構築するために必要な要件として以下の要素が必要である．

- インターネット上への共有空間の構築
- コミュニケーションにおける発話の受話者間における発話者の認識手法の提供

次節以降では，それぞれの用件に対する本研究におけるアプローチを述べる．

3.2　インターネット上への共有空間の構築

実世界で行われるコミュニケーションは，共有された空間へ発話することによって行われる．従って，インターネット上に同様のコミュニケーション環境を構築するためには，インターネット上に仮想的に共有空間を構築する必要がある．

インターネット上に共有空間を構築するためには，以下の手法が考えられる．

- サーバ利用モデル
- Unicast の多重化
- 同報通信の利用

本節では，それぞれの手法について検討を行い，本研究に最適な手法を選択する．

3.2.1 サーバ利用モデル

現在のインターネットでは，特定のサービスを提供する中央サーバを構築するサーバ利用モデルが最も基本的なモデルである．中でも，Web 技術を利用したサーバ利用モデルによる共有空間の構築は，最も一般的な手法である．近年では，Web2.0 というバズワードと共に共有空間の構築を目的とする Web サービスが爆発的に増加した．2.3.1 節にて述べた Twitter も，Web2.0 の流行と共に生まれたサーバ利用モデルのサービスである．

サーバ利用モデルの利点としては，サーバが一箇所に集中するため，管理運営が容易であるという点が挙げられる．また，特定のサーバに対する通信を想定すれば良いため，実装も容易になる．

一方で，特定のサーバに対するインターネットを介した接続性が要求されるため，外部接続が存在しないネットワークでの利用ができないなどの欠点も存在する．また，管理が一箇所に集中するため，その一箇所が停止するとサービス全体が停止してしまうため，サーバが単一障害点になりやすい．

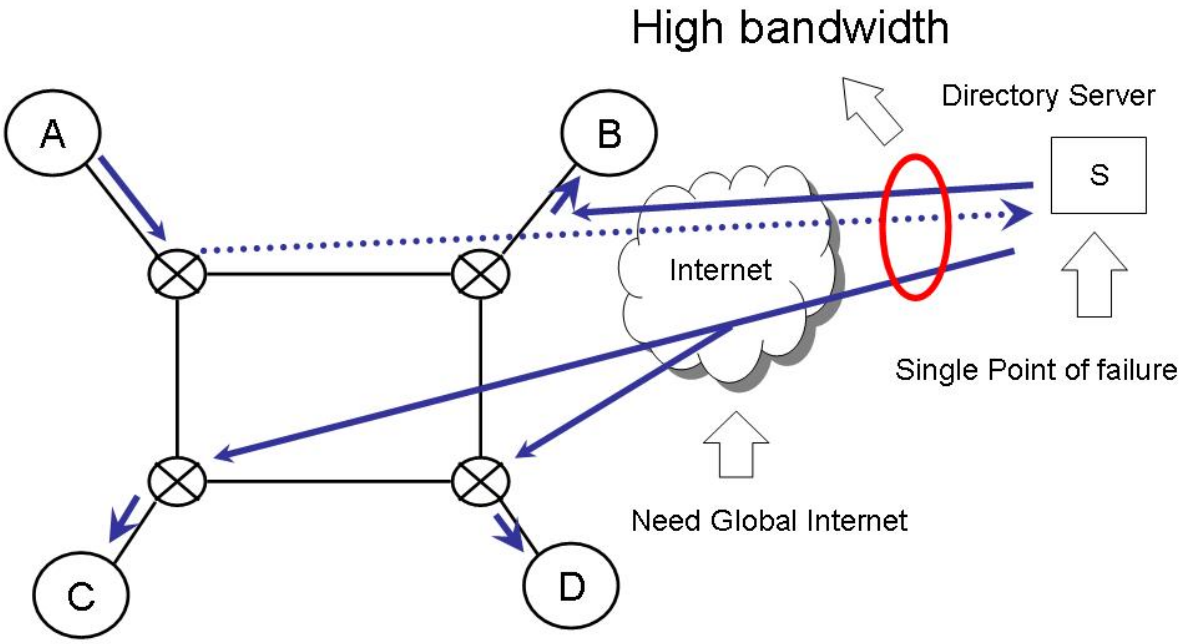

図 3.1: サーバ利用したモデル

3.2.2 同報通信

サーバ利用モデルと Unicast 多重化の欠点を補えるモデルとして，同報通信を利用するモデルが挙げられる．同報通信を用いることにより，同報通信の到達可能な範囲内であれば，容易に多対多の通信環境を構築することができる．

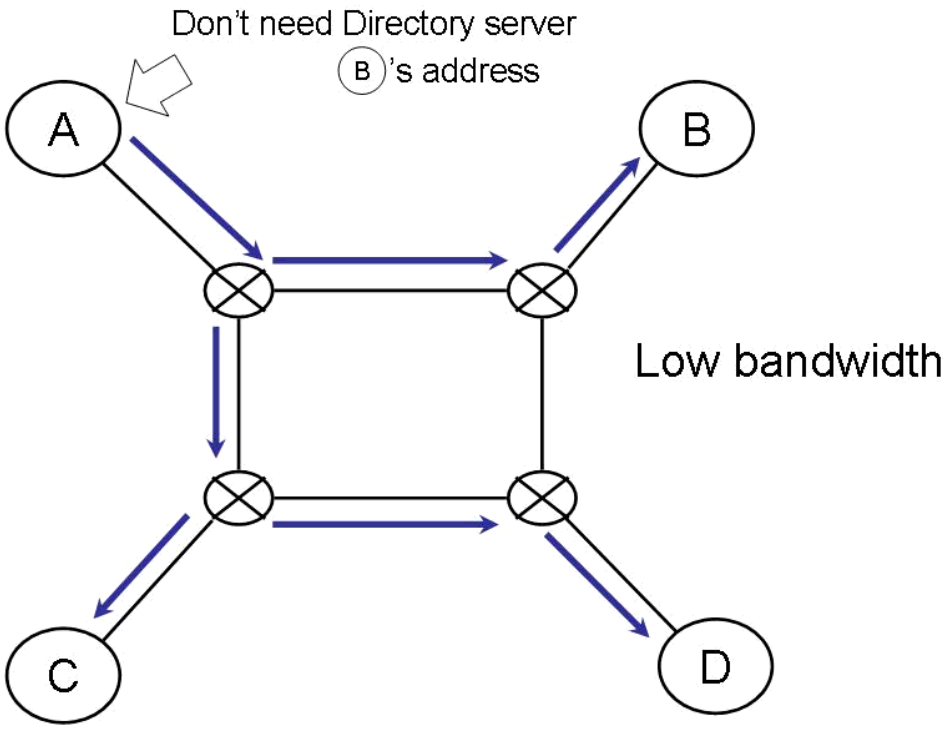

図 3.2: 同報通信を利用したモデル

本研究では，この同報通信を利用して共有空間を構築する．各ユーザは，特定のサーバに依存せず，ネットワークの外部接続性や状態に左右されずに，共有空間を構築するためのメッセージを送信できる．共有空間に参加する各ユーザは自分に関係するメッセージのみを受信する．

次章にて，同報通信の実現方法を述べる．

3.3 同報通信環境の構築

3.2 節で，インターネット上にコミュニケーションを行うための共有空間を実現するために最適な手法として IP Broadcast や IP Multicast といった同報通信を採用した．本節

では，同報通信環境を構築するためにどのような条件の設定が必要でどのようなプロトコルが必要か考察する．

3.3.1 同報通信の利用

本機構では，インターネット上にコミュニケーションを行うための共有空間を実現する手法として，同報通信を用いる．

現在，インターネット上の同報通信の手法として，IP Broadcast と IP Mulitcast が主に用いられている．同報通信環境を構築するためには，どちらの手法も利用可能である．IPv4 では IP Broadcast を行うためには IP Broadcast アドレスが用意され，IP Multicast を行うためには IP Multicast アドレスが用意されているため，それぞれを分けて取り扱う必要がある．しかし，IPv6 では IP Broadcast アドレスは，IP Multicast の一種類として扱われている．また，IP Multicast はインターネットへの到達可能性がないネットワークにおいても，同一ネットワーク内で通信を行うことが可能である．そのため，本機構で同報通信環境を構築するために，IP Multicast を用いる．

IP Multicast を用いて，到達範囲を調整するためにはパケットの TTL(Time To Live) を適切に調整する必要がある．TTL とは，パケットの生存期間を表した値で，この値はパケットがルータを通過するごとに 1 ずつ減算される．家庭内で利用することを想定されて策定された，DLNA[7] による DLNA ガイドラインでは，ホームネットワークというものが定義されている．また，DLNA の DLNA リンクプロテクションガイドラインでは，著作権保護技術として DTCP-IP[8] を必須としており，DTCP-IP においてコンテンツの宅外流出を防ぐために IP ヘッダの TTL を 3 以下にすることが規定されている．本研究で想定する同報通信環境には，DLNA におけるホームネットワークの概念と似た近接の概念が必要である．よって，本機構の同報通信環境の構築に用いられる IP Multicast の TTL は，DTCP-IP の規定に倣い，3 以下とする．

3.3.2 通信に必要な機構

インターネット上で同報通信を利用する場合，通信は UDP を用いたコネクションレスのデータグラムによって行われる．これは，IP Multicast に代表される同報通信は，一対多の通信であるため，送信者と受信者の間における通信の協調が困難なためである．そのため，データの完全性や送信者の一意性の確認を同報通信で実現するには，UDP などオペレーティングシステムから提供される手段を利用するのではなく，別の手段で行う必要がある．

本機構では，この問題を解決するために，送信するデータに新たに規定したプロトコルに基づくヘッダを付加する．データの完全性や送信者の一意性を確認するために必要だと考えられる情報を以下に挙げる．

- 送信者の一意性
  UDP では，セッションの概念がないため，送信元 IP アドレスの情報は得られるも

のの，受信したデータフローの送信者が常に同じである保証はされない．そのため，送信者側でそれを識別するため，**送信されたデータフローに一貫した ID** が必要である．このような ID を振ることにより，受信者側でデータフローの一貫性を確認することができる．

- データの欠落の検証
  UDP では，再送などをせずにデータが送受信される TCP のように受信したデータの欠落情報などを検知する手法は提供されていない．インターネット上のコミュニケーションでは，完全なデータの受信が保証されていないと，文字や音声，映像などどのようなメディアを用いていても，データの再構成が困難になる．特に，文字などの小さなデータが欠落した場合，コミュニケーションに大きな影響が出る．しかし，受信者側で途中経路でデータが欠落したのか，送信された時点でそのようなデータであったのかを判別することはできない．そのため，受信したデータが連続したデータであるかどうかを表す，**データフロー内におけるシーケンス番号**を送信者側で付加する必要がある．これにより，受信者側でデータが連続した物であるかどうかを確認できる．

- パケット単位でのデータの整合性の保証
  IP Multicast では，複数の送信者が同時に同じ Multicast Group Address へデータを送信した場合，データの輻輳が発生し，受信者側では複数のデータが混ざった状態で受信されてしまう．そのような状態でデータの再生が行われると，特に音声や映像などのメディアの場合，データを正しく再生することができない．そのため，送信者側で**送出するパケットの大きさの情報**を付加することにより，受信者側で受信したデータの確からしさを確認することができる．データの完全性を確認するためには，CRC などのチェックサムを用いられることが多いが，このような演算には時間がかかるため，音声や映像などのメディアの場合，再生に影響が出ることが考えられる．そのため，本機構ではパケットのトータルサイズ情報によってのみ，データの確からしさの確認を行う．

3.4 インターネット上の共有空間における個人特定

インターネット上に共有空間を作る手法として，3.3 節で述べたように同報通信を利用する場合，サーバを用いたり，Unicast を用いた通信を行う場合より，送信者および受信者の特定を行うことは困難である．そのため，同報通信環境という特殊な環境で，お互いを特定するための手法が必要である．

3.4.1 同報通信環境における通信の制約

同報通信環境では，送信者がネットワーク上の一定の範囲に対して，送信先を特定せずに一斉にデータを送信する．受信者は，一斉に送信された取得可能なデータを，受信する

のみである．そのため，送信者が受信者および受信者の数を特定することは困難である．また，受信者はそのデータの送信元アドレスを知ることはできるが，そのアドレスを用いて 1 対 1 の通信を開始し，送信者の情報を得ることは現実的ではない．

同報通信環境において，送信者の個人の特定は一斉に行われるデータ送信と同時に行われるべきである．次項では，データ送信のみにより，事後の送信者への確認要求なしに送信者個人を特定する手法について検討を行う．

3.4.2 相互通信を必要としない個人の特定

コミュニケーションに用いられるデータの送信者と各受信者の間で，それぞれ通信を行って送信者を特定するのは前項で述べたように現実的ではない．そのため，送信者は一度のデータの送信のみで，受信者は受信したデータのみで送信者の特定を行う必要がある．また，同報通信の特徴として，インターネットに接続できないアドレスのみを持つノードでも通信可能であるという点があるため，個人の特定の際にインターネット上のノードとの通信が必要とされる手法では，インターネットに接続できないアドレスのみを持つノードはコミュニケーションに参加できなくなってしまう．

このような環境での個人の信頼性を保証する手法として，本研究では公開鍵と秘密鍵からなる鍵対を用いる．送信者はデータを送信する際に自分のキーの ID とその ID を秘密鍵で暗号化した物を付与して送信する．受信者は受信したデータの ID に対応する公開鍵を用いて暗号化されたキーを復号化し，元の ID との対応が確認できたら送信者を特定することができる．また，受信者はデータに付与された ID に対応する公開鍵を持っていないと，データを復号化できず，データの送信者を特定できないため，送信者を特定することができないデータを受け取らない．これにより，実世界のコミュニケーションで自然に行われている，会話の取捨選択をインターネット上のコミュニケーションアーキテクチャで実現する．

3.4.3 個人特定に利用可能な技術

本機構では，同報通信環境における個人特定の手法として，デジタル署名技術に着目した．本節では，デジタル署名技術，信頼の輪モデルおよび PGP 信頼モデルについて解説を行い，本機構への応用可能性について検討する．

デジタル署名技術

以下にデジタル署名技術の基本的な考え方を紹介する．

Alice(A) が公開鍵と秘密鍵のペア $\langle K_A, K_A^{-1} \rangle$ を持つとする．単純化のため，各個人はそれぞれ 1 対の鍵ペアを保有するとする．

デジタル署名には，以下の二つの目的がある．

1. Alice がメッセージ m を認めた事を証明する

2. m が変更されていない事を証明する

これらは，Alice の秘密鍵 K_A^{-1} を使って暗号化を行い，Alice の公開鍵 K_A を使ってのみ複合可能な $\{m\}_{K_A^{-1}}$ を生成することによって実現される．K_A^{-1} が Alice のみが知る秘密であるため，Alice の公開鍵でしか複合できない $\{m\}_{K_A^{-1}}$ は Alice によって暗号化されたと推論できる．また，$\{m\}_{K_A^{-1}}$ を複合した結果と m を比較することにより，$\{m\}_{K_A^{-1}}$ が生成された時点から m の内容が変更されていない事も確認できる．

一般的には，効率を高めるために m そのものを暗号化するのではなく，m に対してセキュアハッシュ関数 H を適応した結果に対して暗号化が行われる．このとき，$m' \neq m$ であるにも関わらず $H(m) = H(m')$ とならないように，H の選択に気をつけなければならない．

定義 1(デジタル署名): A が m と暗号化された $H(m)_{K_A^{-1}}$ を提供するとき，$A \xrightarrow{signs} m$ と記述する．このとき，$H(m)_{K_A^{-1}}$ を m に対する署名と呼ぶ．

Alice の公開鍵を持つ Bob は，署名を検証することができる．Bob は，$H(m)$ を計算し，K_A を使って $H(m)_{K_A^{-1}}$ を複合して得られる結果と比較する．

ただし，Bob が持つ Alice の公開鍵の妥当性に対する疑問は残る．

定義 2(関係の妥当性): x が y の公開鍵コピー K_y を保持し，そのコピーが本物であると推察するとき，$x \xrightarrow{v} y$ と記述する．

また，$x \xrightarrow{v} y \wedge y \xrightarrow{v} x$ であるとき，$x \xleftrightarrow{v} y$ と記述する．

公開鍵の妥当性に関する信頼モデルは，$\xrightarrow{v}$ の組み合わせによって実現される．一般的に，公開鍵の妥当性は鍵に対する署名や，証明書によって証明される．例えば，Bob(B) が Cameron(C) を知り，$B \xrightarrow{v} C \wedge C \xrightarrow{signs} K_A$ であるとき，C の証明書が信頼できるのであれば $B \xrightarrow{v} A$ が成り立つ．この関係は再帰的であるため，最終的にはどこかで自分自信を認証できる存在が必要となる．

公開鍵方式は，認証機関 (Certificate Authority) のツリー構造を持つ．それぞれの公開鍵は，上位の認証局によって認証される．ツリー構造の頂点に位置する認証局がルート CA である．

Web of Trust

Web of Trust(以下，WOT) においては，認証局に頼らずに個人間の相互認証による輪によって公開鍵の妥当性を確保する．WOT は，公開鍵を認証し合う人と人の輪である．

$\xrightarrow{s}$ をサインすることによって，公開鍵のコピーが信頼できると表明することができる．

定義 3(サイン): $\xrightarrow{s}$ は以下のように定義される．

1. $x \xrightarrow{s} x$
2. $x \xrightarrow{s} y$ if $x \xrightarrow{signs} K_y$

$x \xrightarrow{s} y \wedge y \xrightarrow{s} x$ であるとき，$x \xleftrightarrow{s} y$ と記述する (相互サイン関係)

定義 4(signing-apart relation): $\xrightarrow{s[n]}$ は以下のように定義される．

1. $x \xrightarrow{s[0]} x$
2. $x \xrightarrow{s[1]} y$ if $x \xrightarrow{s} y \wedge x \neq y$.
3. $x \xrightarrow{s[a+b]} z$ if $x \xrightarrow{s[a]} y \wedge y \xrightarrow{s[b]} z$.

信頼の輪の中間に誰が存在するのかを示すために，$A \xrightarrow{s[n+1]} C$ if $A \xrightarrow{s} B \wedge B \xrightarrow{s[n]} C$ (expansion of signing-apart relation) の代わりに $A \xrightarrow{s} B \xrightarrow{s[n]} C$ と記述する．

定義 5(Web of Trust, 信頼の輪): x にとっての WOT とは，$n \geq 0$ のときの $x \xrightarrow{s[n]} y$ となる全ての y の集合である．

WOT では，認証関係が厳密に示される必要がある．PGP(Pretty Good Privacy) は，このような関係を表現するための暗号化技術である．

PGP 信頼モデル

$\mathcal{T}_x$ を “完全に信頼できる”ユーザ群 x とし，$\mathcal{T}'_x$ を “ある程度信頼できる”ユーザ群 x とする．

PGP 鍵の署名というコンテキストにおいては，“完全に信頼できる” とは，公開鍵の保有者が鍵署名を完全に理解しており，署名者がその公開鍵が自分の物と同等に信頼できるものであるという事を示す．“ある程度信頼できる” とは，公開鍵保有者が鍵署名の意味を理解しており，署名する前に正しく確認を行った事を示す（参考文献 4）．

PGP 信頼モデルは，*validating relation* $\xrightarrow{v}$ を WOT に対して行うものである．

定義 6(PGP 信頼モデル): $x \xrightarrow{v} y$ if

1. 十分な数の有効な鍵を持つユーザが y の公開鍵を署名している．
   例えば，
   (a) $x \xrightarrow{s} y$, または
   (b) 少なくとも f 個の z が存在する場合，$z \in \mathcal{T}_x$, $x \xrightarrow{v} z \wedge z \xrightarrow{s} y$, または
   (c) 少なくとも m 個の z が存在する場合，$z \in \mathcal{T}'_x$, $x \xrightarrow{v} z \wedge z \xrightarrow{s} y$; そして
2. $n \leq h$ のとき $x \xrightarrow{s[n]} y$

このとき，

f : “完全に信頼できる”鍵の持ち主の数

m : “ある程度信頼できる”鍵の持ち主の数

h : WOT において y から x を辿るパスのステップ数の上限

である．

GnuPG の初期設定では，$f = 1, m = 3, h = 5$ としている．

本機構への応用可能性

本研究で想定している同報通信環境は，3.4.1 項で述べたように，双方向でない一方向の通信によって，データが送信される．また本研究では，3.4.2 項で述べたように，インターネットに接続できないアドレスのみを持つノードも，利用対象として想定している．そのため，PGP 信頼モデルにおける鍵の利用は，サーバなどインターネット上にある別のノードに依存せず，デジタル署名の信頼性を確認できるので，本機構で用いるのに最適である．よって，本機構では 3.4.2 項で述べた公開鍵と秘密鍵の鍵対の方式として PGP を用いる．

3.4.4 自律的な鍵の伝播

本機構では，同報通信環境において，PGP 鍵を利用した信頼関係にある人同士の会話の輪を共有空間として，コミュニケーションを実現する．その際に，本研究では知り合いである人同士では，あらかじめメールや名刺交換などによって，お互いに公開鍵の交換を完了していることとする．

本機構では，PGP を用いて，3.4.2 項で述べた個人の特定の連鎖によって，公開鍵の交換を行っている人の輪の中で共有空間を構成し，コミュニケーションを行う．この様子を図 3.3 に示す．

また，すでに構成された共有空間であるグループがコミュニケーションを開始している際，グループの一部のメンバとのみ鍵交換を行っている新規メンバが新しくコミュニケーションに参加したい場合が発生することが考えられる．本機構では，知り合いの知り合いは知り合いであると仮定し，グループ内のメンバおよび新規メンバ双方の公開鍵を持っているユーザが，それぞれに不足する公開鍵を提供し，新規メンバを共有空間に招待することができる．この様子を図 3.4 に示す．

このように，信頼関係の成立している間での鍵の伝播を自律的に行うことにより，共有空間を自律的に拡大することが可能になる．

3.5 本研究のアプローチ

本研究では，実世界で行われている共有空間におけるコミュニケーションをインターネットで実現するために，同報通信を利用したコミュニケーションアーキテクチャを提案する．本手法の実現のために以下に述べる手法を確立する．

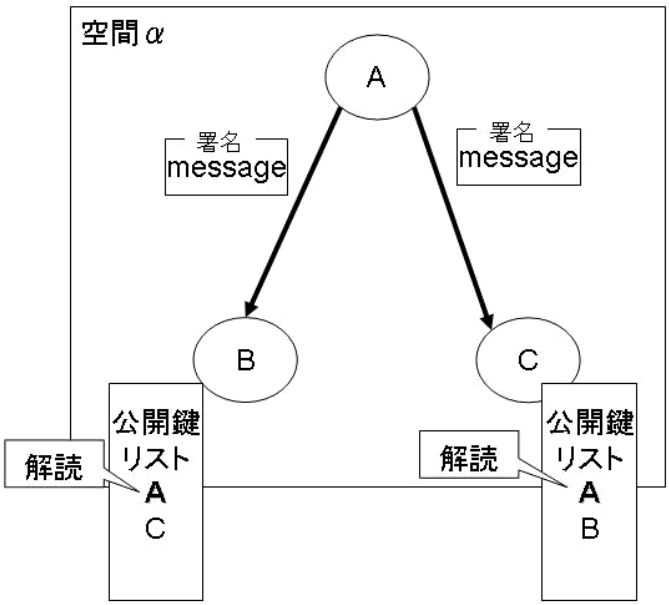

図 3.3: 信頼関係の輪による共有空間の構成

3.5.1 仮想的な共有空間の構築

インターネット上で，実世界で行われている共有空間におけるコミュニケーションを実現するためには，仮想的なコミュニケーションを行うための共有空間を構築する必要がある．本研究では，仮想的な共有空間を同報通信を用いて構築する．また，同報通信を用いたコミュニケーションを実現するために，コミュニケーションに必要なデータのやりとりを支援する情報を定義し，この情報を効率的に共有するプロトコルを定義する．インターネット上の同報通信を本研究で定義したプロトコルを用いて行う環境を，同報通信環境とする．

3.5.2 送信者の個人の特定

実世界で行われているコミュニケーションにおいて，コミュニケーション相手の認識は重要な要素である．インターネットでは，コミュニケーションの相手とネットワークを介して通信するために，直接認知することは難しい．また，同報通信環境においては，コミュニケーションを行う人同士で情報を交換し合い，相手を認知することは困難である．

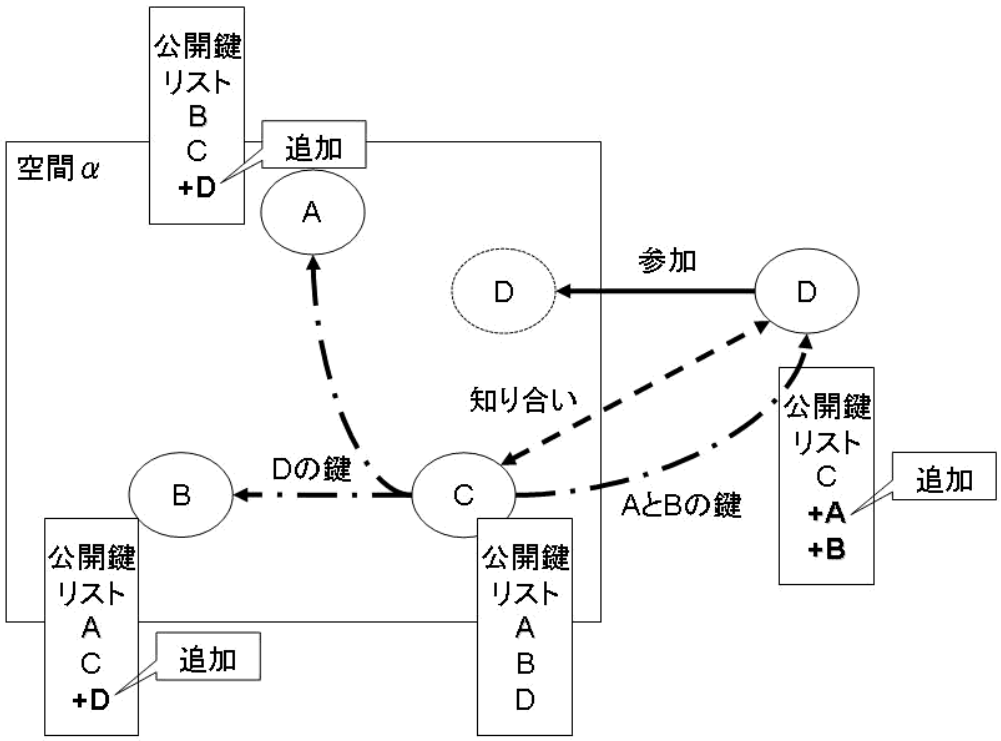

図 3.4: 自律的な鍵の伝播と共有空間の広がり

そこで，本研究では，送信者側から自身を識別するための署名を送信し，受信者側においてその署名を検証することで，送信者を特定する．送信者が受信者を特定してコミュニケーションを行うのではなく，受信者が送信者を特定することで，コミュニケーションを行う相手を特定する．また，同じ同報通信環境にいる人がそれぞれに発言を行い，それぞれが送信者となることにより，お互いがお互いを認識することができ，特定できる人の間でのコミュニケーションを成立させることができる．

3.6 まとめ

本章では，インターネット上における共有空間の構築および共有空間における，送信者の特定を行うために必要な技術の調査および手法の提案を行った．また，共有空間の構成に必要な要件をまとめた．

次章では本章で提案した手法を実現するための，コミュニケーションアーキテクチャの設計を行う．

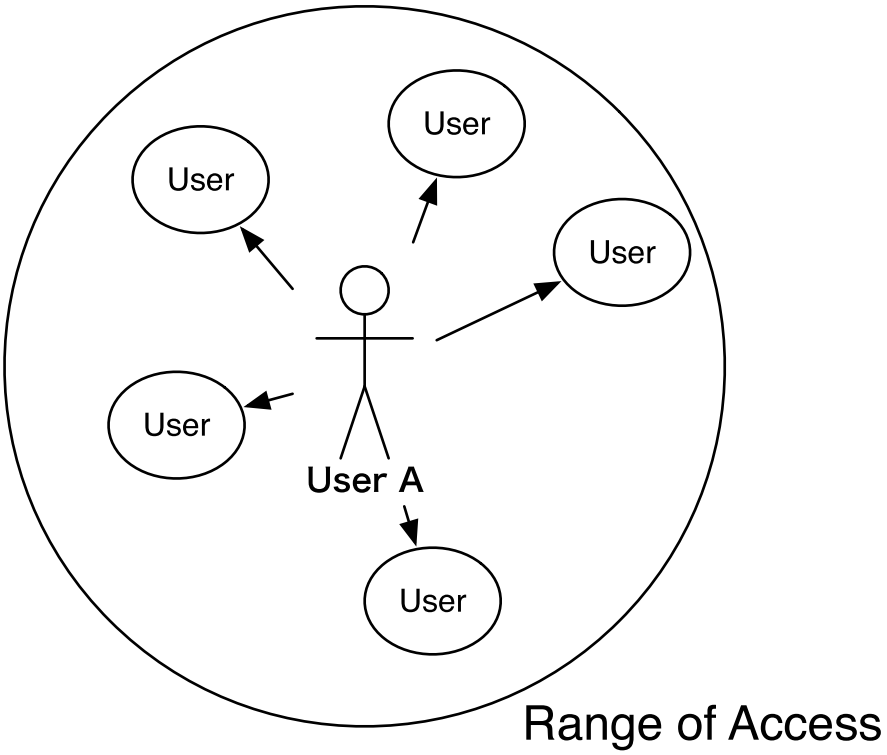

図 3.5: 共有空間内でのコミュニケーション

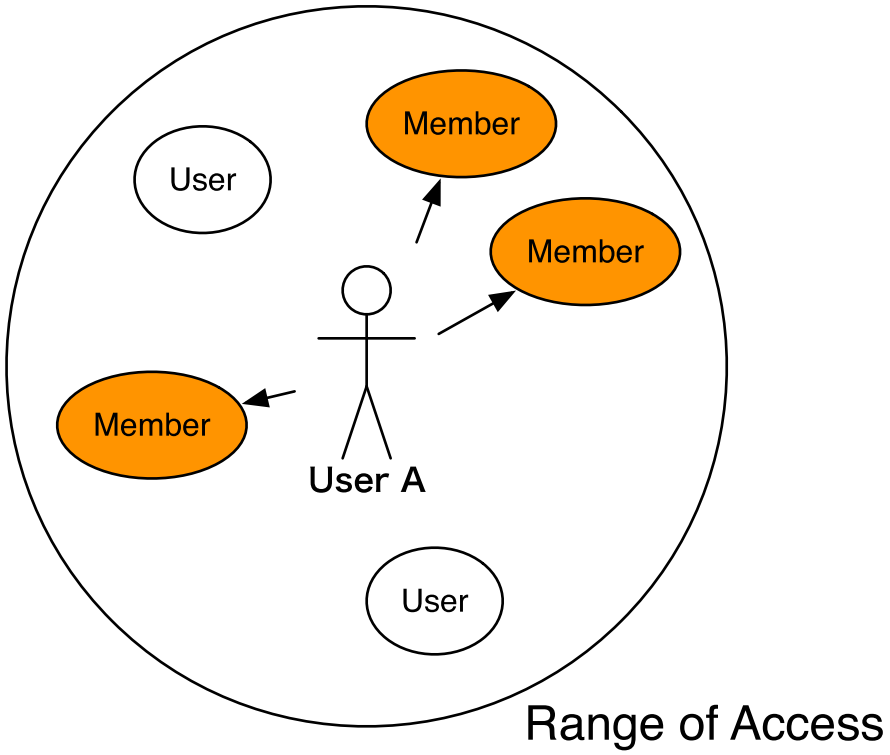

図 3.6: 個人の特定によって生まれるグループ内コミュニケーション

第4章　同報通信環境におけるコミュニケーションアーキテクチャの設計

本章では，同報通信環境を構築するための設計を行う．また，その上でコミュニケーションを行うために必要なアーキテクチャの設計を行う．

4.1　設計概要

本研究では，まずコミュニケーションを行うために必要な同報通信環境を構築する．その上で，構築された同報通信環境を用いてコミュニケーションを行うのに使われる，個人特定手法の実装を行う．

4.2　同報通信環境の構築

本研究では，ユーザ間のデータのやりとりに IP Multicast を用いる．そのため，ユーザはデータを UDP で送出する．UDP では TCP のようにデータの完全性の保証やデータの送信元の特定は行われない．

本研究ではこれらを補うために新たにプロトコルを作り，そのプロトコルによってデータの確からしさと送信元の一意性の確認を行う．

4.2.1　共有空間制御プロトコル - Shared Space Control Protocol

同報通信環境では，3.4.1 項で述べたように，一方向の通信によってデータの送信が行われる．そのため本研究では，データの確からしさと送信元の一意性を保証するために，データに関する情報を記述したヘッダを各送信データに付加する．このヘッダに記述された情報を基に，受信者はデータの確からしさと送信元の一意性の確認する．

本プロトコルに記述する情報は，3.3.2 項で述べた必要な機構を満たす必要がある．従って，ヘッダに記述すべき情報は以下の通りである．

- ソース識別子
  送信者の一意性を表す
- シーケンス番号
  データの欠落の検証に用いる

- パケットの全長
  パケット単位でのデータの整合性の検証に用いる

これらの情報を，送信者はプロトコルヘッダとしてコミュニケーションに用いられるデータに付与し，パケットの同報通信環境への送信を行う．また，受信者は付与されたプロトコルヘッダの情報を基に，受信したデータの検証を行う．
次項では，プロトコルヘッダのデータ構造の詳細について述べる．

4.2.2 共有空間制御プロトコルヘッダ

4.2.1 節で述べたプロトコルを実現するためのヘッダを図 4.1 のように定義した．

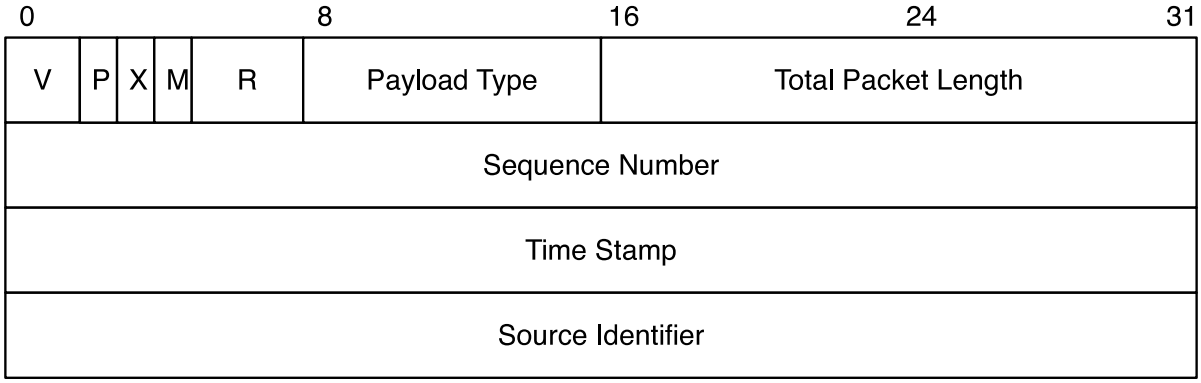

図 4.1: SSCP ヘッダ

ヘッダの各フィールドの詳細を以下で述べる．

V :
2 ビット．SSCP のバージョン番号を表す．本研究ではバージョンを 1 とした．

P :
1 ビット．パディングがあるかを表す．SSCP のペイロード部分の長さは，4 バイトの倍数でなくてはならない

X :
1 ビット．エクステンションヘッダがあるかを表す．エクステンションヘッダビットが立っている場合には，SSCP ヘッダの直後にエクステンションヘッダが続かなくてはならない．

M :
1 ビット．マーカービット．プロファイル毎に使い方を定義しても良い．

R :
3 ビット．Reserved．将来拡張がなされる場合に用いられる．

Payload Type :
8 ビット．ペイロードタイプを表す．SSCP のペイロードの種類を表す．

Total Packet Length :
16 ビット．ヘッダを含んだパケットの全長を表す．

Sequence Number :
32 ビット．シーケンス番号を表す．パケット一つ毎に増加する．シーケンス番号の初期値はランダムな値でなくてはならない．

Timestamp:
32 ビット．タイムスタンプを表す．タイムスタンプのためのクロック周波数はペイロードの種類によって定義されている．

Source Identifier :
32 ビット．ソース識別子を表す．異なる Source Identifier 値を使用することにより送信者のネットワークアドレスが同一でも区別することができる．

4.2.3 IP Multicast の制御

本機構では，同報通信の手法として IP Multicast を用いる．IP Multicast を利用する上では，3.3.1 項で述べたように，データの到達範囲を調整するために，パケットの TTL を適切に調整する必要がある．

4.3 個人特定手法

IP Multicast の環境ではセッションがないのでデータのやりとりで個人特定を行うことが難しい．そのため，送信者からのデータ送信のみに依る個人特定手法が必要である．本機構では，3.4.3 項で述べたように，その手法として PGP を用いる．

送信者からのデータ送信のみで，受信者が送信者が誰であるかという確認を行うために，本機構では，送信者が自分の秘密鍵を用いて署名を作成し，同報通信環境に送信を行う．受信者は，受信した署名を確認できる公開鍵を持っていた場合のみ，送信者の特定を行うことができる．送信者の特定できたデータのみ，出力することにより，公開鍵を交換しているグループ内におけるコミュニケーションが成立する．

4.3.1 個人特定情報の付加

PGP の秘密鍵から署名を作成して付加送信するデータに付加する．受信側が公開鍵束から署名を検証し，検証できた場合は送信者を特定することができ，知り合いであることが分かる．知り合いからのデータであることを特定できた場合は，受信者はデータをアプ

リケーションに渡す．特定できない場合は，知り合いからのデータではないので．アプリケーションに渡さずにデータを破棄する．

4.3.2 公開鍵の伝播

3.4.4 項で述べたように，本機構では知り合いの知り合いは知り合いという，知り合いをすでに信頼していることを前提に公開鍵を伝播させる．これにより，コミュニケーションを行う共有空間を自律的に拡大することができる．公開鍵をもっていない知り合いに，すでにお互いの鍵を持っている共通の知り合いから鍵を渡してもらうことにより，公開鍵の自律的な伝播を実現する．

4.4 まとめ

本章では，第 3 章で述べた提案に基づき，インターネット上における共有空間の構築を行うための設計について述べた．

次章では本章で述べた設計を基に行った実装について述べる．

第5章　同報通信環境におけるコミュニケーションアーキテクチャの実現

5.1　実装概要

本章では，第 4 章の設計に基づいた実装に関して述べる．

5.1.1　実装環境

本機構の実装を行った環境を表 5.1 に示す．

表 5.1: 実装環境

OS	Mac OS X 10.5.3 (Intel)
プログラミング言語	Java Version 1.5.0_13

5.2　実装クラス

本研究では，同報通信環境を構築し，その上でコミュニケーションを実現するプロトタイプ実装を行った．この実装は，送信側の処理と受信側の処理に大別できる．以降では，送信側，受信側それぞれの処理について述べる．

5.2.1　送信処理

送信側の処理に用いられるクラスとして，以下の各クラスを実装した．

1. McastSend クラス
2. SendSscp クラス
3. AddSig クラス

以下で，それぞれのクラスについて説明する．

McastSend クラス

同報通信環境を構成するあらかじめ決められた IP Multicast Group Address へ実際のデータの送出を行う．引数として送信されるデータを取る．本クラスからは，SSCP ヘッダを付加するために後述する SendSscp クラス，署名を追加するために AddSig クラスが呼ばれる．

SendSscp クラス

同報通信環境へデータを送出する際に，共有空間制御プロトコルに基づいた SSCP ヘッダを付加する必要がある．本クラスのインスタンスが作成される際に，ヘッダを作成するのに必要な以下の変数が初期化される．

- sequenceNum
  ヘッダに付与するシーケンス番号を格納
- timeStamp
  ヘッダに付与するタイムスタンプを格納
- syncSourceId
  ヘッダに付与する Flow ID を格納

本クラスでは，SSCP ヘッダ長を定義する変数として HEADER_SIZE が定義されており，値は 16 である．

本クラスでは，addSscpHeader() 関数を用意し，この関数でデータへ SSCP ヘッダの付加を行う．addSscpHeader() 関数は引数として送信されるデータを取り，そのデータより送信データの全長を計算する．また，クラスのインスタンスが作成されて初めてこの関数が呼ばれた際に，sequenceNum, timeStamp, syncSourceId の各変数を初期化する．

```
/* 初回実行時の変数初期化処理 */
if(!sscp.sscpInitFlag){
    Random rnd = new Random();

    sscp.sequenceNum = (short)rnd.nextLong();
    sscp.timeStamp = (int)rnd.nextLong();
    sscp.syncSourceId = (int)rnd.nextLong();

    sscp.sscpInitFlag = true;
}
```

図 5.1: addSscpHeader() 関数における各変数の初期化処理

ここまでで用意された情報から，ヘッダを作成する．

```
/* ヘッダ生成処理 */
sscpHeader[0] = (byte)(version|padding|extension|marker|reserved);
sscpHeader[1] = (byte)(payloadType);
sscpHeader[2] = (byte)(totalSize >> 8);
sscpHeader[3] = (byte)(totalSize >> 0);
sscpHeader[4] = (byte)(sscp.sequenceNum >> 24);
sscpHeader[5] = (byte)(sscp.sequenceNum >> 16);
sscpHeader[6] = (byte)(sscp.sequenceNum >> 8);
sscpHeader[7] = (byte)(sscp.sequenceNum >> 0);
sscpHeader[8] = (byte)(sscp.timeStamp >> 24);
sscpHeader[9] = (byte)(sscp.timeStamp >> 16);
sscpHeader[10] = (byte)(sscp.timeStamp >> 8);
sscpHeader[11] = (byte)(sscp.timeStamp >> 0);
sscpHeader[12] = (byte)(sscp.syncSourceId >> 24);
sscpHeader[13] = (byte)(sscp.syncSourceId >> 16);
sscpHeader[14] = (byte)(sscp.syncSourceId >> 8);
sscpHeader[15] = (byte)(sscp.syncSourceId >> 0);
```

図 5.2: addSscpHeader() 関数におけるヘッダ生成処理

ヘッダの作成を終えると，次回の送信のために，sequenceNum, timeStamp の各変数の値を更新する．addSscpHeader() 関数は返値として送信されるデータに SSCP ヘッダを付加したものを返す．

```
/* 返値用の変数にヘッダとデータを代入 */
ByteArrayOutputStream out = new ByteArrayOutputStream(sscp.totalSize);
out.write(sscpHeader, 0, HEADER_SIZE);
out.write(payloadData, 0, payloadSize);

/* 返値を返す */
return out.toByteArray();
```

図 5.3: addSscpHeader() 関数におけるパケットデータの生成と返値の処理

AddSig クラス

同報通信環境に，コミュニケーションを行うための共有空間を創出するために必要な，デジタル署名の付加を行う．本クラスでは GnuPG を用いて，秘密鍵より署名を生成する．

5.2.2 受信処理

受信側の処理に用いられるクラスとして，以下の各クラスを実装した．

1. McastRecv クラス

2. RecvSscp クラス

3. CheckSig クラス

以下で，それぞれのクラスについて説明する．

McastRecv クラス

同報通信環境を構成するあらかじめ決められた IP Multicast Group Address へ送信されたデータの受信を行う．本クラスでは受信したデータをパケットサイズから推測し，適切なサイズの受信バッファを用意し受信する．これは，受信したデータが音声や映像であった場合，バッファサイズが不適切だと受信したデータを正しく再生できないためである．

本クラスからは，SSCP ヘッダを適切に処理するために，後述する RecvSscp クラスが呼ばれる．本実装では，RecvSscp クラスの返値により，データが正しく受信できていないことが分かった場合，“SSCP Header Error.” というメッセージを出力し，受信データの出力処理を行わない．

また，送信者によってデータに付加された署名を検証するために，後述する CheckSig クラスが呼ばれる．本実装では，CheckSig クラスにより，送信者の特定を行い，送信者の情報を出力する．

RecvSscp クラス

同報通信環境から受信したデータには，送信側で付加した共有空間制御プロトコルに基づいた SSCP ヘッダが付加されている．

本クラスでは，SSCP ヘッダ長を定義する変数として HEADER_SIZE が定義されており，値は 16 である．この値は，McastRecv クラスにおいてバッファ量を設定する際に用いられる．

本クラスでは，processSscpHeader() 関数を用意し，この関数で受信データ内の SSCP ヘッダの処理を行う．processSscpHeader() 関数は引数として受信したデータと受信したパケット長を取る．受信した SSCP ヘッダから，totalLength, squenceNum, timeStamp, syncSourceId の各値が評価される．

それぞれの値が正しくなかった場合には，受信エラーとして長さ 1 ，値 -1 の byte 型配列変数を返値とする．それぞれの値が正しかった場合には，データの先頭からヘッダを除いた物を返値とする．

CheckSig クラス

送信者より送信されたデータに付加されている，送信者の署名データの検証を行う．署名データの検証により確認された送信者の情報を返値とする．

```
/* ヘッダ内データの読み込み */
int headerTotalLength1 = recvData[2];
int headerTotalLength2 = recvData[3] & 0xFF;
int headerTotalLength = (headerTotalLength1 << 8) | headerTotalLength2;
:
int headerSequenceNum = recvData[4];
int headerSequenceNumTmp;

for(i = 5; i < 8; i++){
    headerSequenceNumTmp = recvData[i] & 0xFF;
    headerSequenceNum = (headerSequenceNum << 8) | headerSequenceNumTmp;
}

int headerTimeStamp = recvData[8];
int headerTimeStampTmp;

for(i = 6; i < 12; i++){
    headerTimeStampTmp = recvData[i] & 0xFF;
    headerTimeStamp = (headerTimeStamp << 8) | headerTimeStampTmp;
}

int headerSourceId = recvData[12];
int headerSourceIdTmp;

for(i = 13; i < 16; i++){
    headerSourceIdTmp = recvData[i] & 0xFF;
    headerSourceId = (headerSourceId << 8) | headerSourceIdTmp;
}
```

図 5.4: processSscpHeader() 関数におけるヘッダ内データの読み込み

5.3 動作概要

本機構の動作イメージを図 5.5 に示す.

5.4 まとめ

本章では，第 4 章で述べた設計に基づいた実装について述べた．次章では実装された本機構を計測，評価を行う.

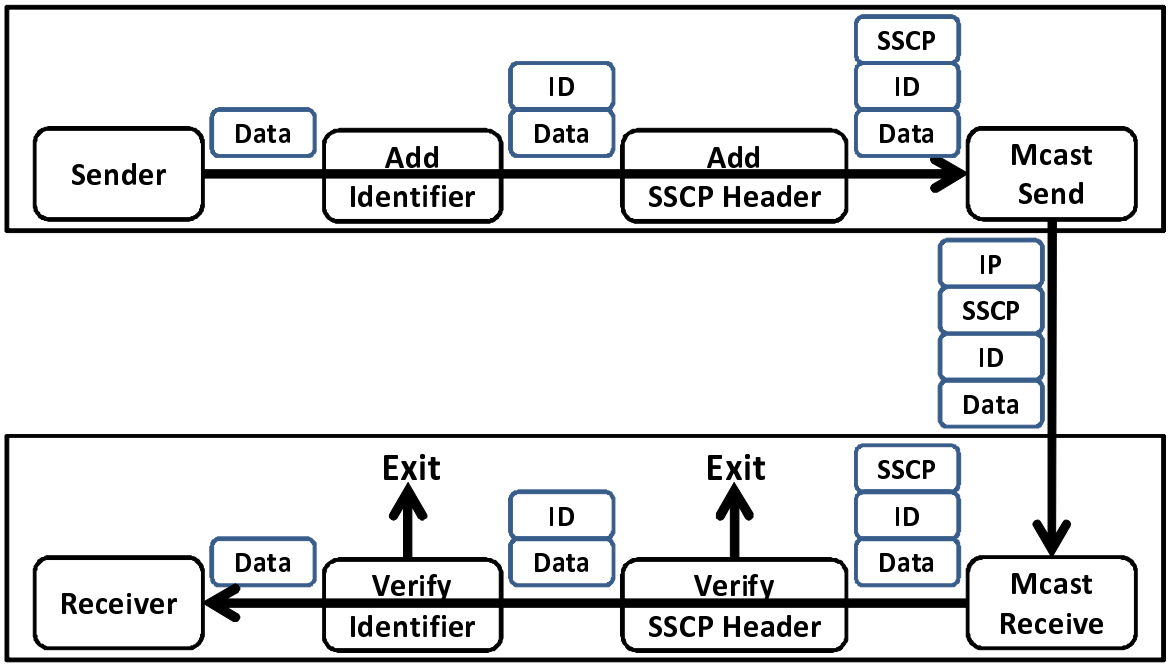

図 5.5: 本機構の動作イメージ

第6章　評価

本章では，設計を元に実装を行ったものの評価について述べる．

6.1　評価概要

本評価は，本研究が提案する同報通信環境におけるコミュニケーションアーキテクチャの有効性の検証を目的とする．定性評価ならびに定量評価は以下の項目について行う．

定性評価： 本機構によって実現した機能
　公開鍵の自律的な伝播におけるセキュリティの検証

6.2　本機構によって実現した機能

本機構では以下の機能を実現した．

- 共有空間（同報通信環境）におけるコミュニケーションの実現
- サーバ非依存のコミュニケーション環境の実現
- インターネット接続性を必要としないコミュニケーション環境の実現

他のインターネット上で用いられているコミュニケーションサービスと本機構の実現機能の比較を行った．比較の結果を表 6.1 に示す．
比較に用いたそれぞれの指標を以下に挙げる．

即時性 通信相手へのデータ到達の即時性．○は直ちにコミュニケーション相手に到達．△は非同期のため直ちに反映されるが，受信者が能動的に確認しなければならない．

規模性 サービスを多人数で同時に利用可能であるかどうか．

起動時間 アプリケーション起動後，サービスをすぐにサービスを利用可能かどうか．○は直ちに利用可能．△は認証などに数秒以上の時間を要する．

単一障害点の有無 サービスに単一障害点により利用不可能になる可能性があるか．○はない．×は利用不可能になる可能性がある．

表 6.1: 実現機能の比較

比較対象	Windows Live	Skype	DVTS	Twitter	ピクトチャット	本機構
即時性	○	○	○	△	○	○
規模性	△	○	×	×	×	△
起動時間	△	△	○	○	○	○
単一障害点の有無	×	×	○	×	○	○
インターネット非依存	×	×	○	×	○	○
共有空間の構築	△	△	×	△	△	○
実装難易度	×	×	×	○	×	○

インターネット非依存 サービスの利用にインターネットへの接続性が必要かどうか．○はインターネットへの接続性がなくても利用可能．×はインターネットへの接続性が必須．

共有空間の構築 複数人数によるコミュニケーションが可能かどうか．○は可能．△は人数に限りがある．×は不可能．

実装難易度 それぞれのサービスを利用するためのアプリケーションを実装することが容易か．○は容易．×は容易でない．

6.2.1 公開鍵の自律的な伝播におけるセキュリティの検証

本研究では，共有空間の自律的な拡大を行うために，知り合いの知り合いは知り合いであるという前提のもとに，まだ鍵の交換を行っていない第三者との鍵の交換を行う．本来，PGP など公開鍵暗号方式においては，すでに認証された人を経由して第三者を認証することはできるが，鍵の受け渡しを行うことは推奨されていない．しかし，本研究では公開鍵をすでに交換している人と人の間では，信頼関係が成立していることを前提としてコミュニケーションを行っている．そのため，自分と第三者がお互いに信頼している人物を経由して公開鍵の交換を行うので，ここでのセキュリティは満たされていると言える．

6.3 まとめ

以上の評価により，実世界で行われている共有空間におけるコミュニケーションが，本機構を利用することで，同報通信環境を用いることにより実現できた．また，共有空間を構成するために用いられる公開鍵により，実際の人間関係をインターネット上のコミュニケーションに適応した．

第7章　結論

第5章で本研究の実装に関して述べた．本章では第5章から導きだされた内容より考察，まとめを行う．

7.1　まとめ

本研究では，インターネット上における，コミュニケーションを行うための，同報通信を用いた共有空間を実現した．

本研究では，インターネット上において実世界で行われている共有空間におけるコミュニケーションを実現するための要件を，以下のように定義した．

- コミュニケーションを行うための共有空間の創出
- コミュニケーションを行う相手の特定

インターネット上において，実世界と同じようなコミュニケーションを実現するサービスは，今まで提供されてこなかった．これは，インターネットにおける通信が，通信相手をあらかじめ特定し，宛先となるIPアドレスを指定して送受信するモデルだったためである．インターネットには，このようなIP Unicast通信だけではなく，IP BroadcastやIP Multicastといった同報通信の手法も用意されている．しかし，IP BroadcastやIP Multicastは一対多の通信であるため，一対一で双方向で通信可能なIP Unicastと異なり一方向の通信手法しか提供されておらず，お互いのデータのやりとりによって双方の確認を行うことはできなかった．

本研究では，このようなインターネット上の同報通信を用いて，データの完全性や送信者の一意性を保証するために，新たにプロトコルを設計し，実装を行い，同報通信環境を実現した．

また，インターネットではコミュニケーション相手とネットワークを介して通信を行うため，相手を直接認識することはできなかった．

本研究では，PGPを用いることにより，一方向の通信における，コミュニケーション相手の特定を実現した．また，PGPの公開鍵の広がりを人間関係と見なすことにより，信頼関係のある人の間でのグループコミュニケーションを実現した．

本研究では，このようなインターネット上における，共有空間でのコミュニケーションを行うフレームワークを実現した．

7.2 今後の課題

7.2.1 Multicast TTL による到達範囲の調整

本研究では，手法の有効性の評価を優先させるために，同報通信環境を構築するために用いる IP Multicast の TTL の値を 1 に固定して評価を行った．しかし，本来同報通信環境を構築するのに適切な IP Multicast の TTL は，3.3.1 節で述べたように 3 以下である．従って，適切な同報通信環境を構築するためには，ユーザの要求に合わせて適切な TTL を設定すべきである．そのため，ユーザの要求に合わせた同補通信環境を構築するためには，ユーザの要求を受け付けるインターフェイスと，その要求に合わせた IP Multicast の TTL を適応できる機構が必要である．

7.2.2 利用する IP Multicast Group Address の選定

本研究では，利用する IP Multicast Group Address にあらかじめ決まったアドレスを利用した．しかし，同じネットワークで複数のグループによるコミュニケーションが行われる場合，同じ IP Multicast Group Address を用いていると，通信の輻輳が発生し，コミュニケーションを十分に行うことができなくなる．そのため，利用する Multicast Group Address を動的に選定し，グループ内で共有する機構が必要である．

7.2.3 コミュニケーションメンバのグルーピング

本研究では，公開鍵を GnuPG を用いて管理している．そのため，公開鍵の鍵束をコミュニケーションを行うグループによって使い分けることが困難である．しかし，コミュニケーションメンバを場合によって使い分けることは，本研究の目的としている，インターネット上の共有空間におけるコミュニケーションの実現とは，実現するレイヤが異なる．そのため，コミュニケーションメンバのグルーピングは，7.2.2 項で述べた，利用する IP Multicast Group Address の選定機構を用いた上で，アプリケーションによって実現されることが望ましい．

7.2.4 公開鍵の自律的な伝播におけるセキュリティ

本研究では，ある人物とべつの人物との公開鍵の交換を，お互いが信頼する人物を経由して交換することにより，セキュリティを実現した．しかし，本研究では，お互いの公開鍵をすでに持っていることを，信頼の指標として用いている．そのため，公開鍵の交換を行う上でのセキュリティの強度には疑問が残る．実際に厳密に公開鍵の交換におけるセキュリティを担保するために，より確実なセキュリティを確保する手法を用いる必要がある．

7.3 本機構の応用例

本研究で構築したアーキテクチャを用いて，以下のようなアプリケーションを作成可能になる．これらのアプリケーションは，今までのインターネットにはなかった新しいコミュニケーションの形態である．

7.3.1 無設定会話アプリケーション

ネットワークの種類に限らず，音声によって「オーイ」と呼びかけると周辺に音声が配送されるアプリケーションを構築する．このアプリケーションは，マルチキャストが可能な設定のネットワークでは広範囲に音声を伝播可能になる．一方で，電車の中や駅前などでアドホックに構成されるネットワークでも半匿名性を保ちながら会話を出来るツールになる．

7.3.2 テキスト/動画アプリケーション

開発当初は音声通話に着目しているが，テキストによる通信や動画による通信も開発する．テキストや動画は，多人数が同時に発言をしたときの合成方法に工夫が必要である．そのため，本研究のような用途における最適な UI も合わせて模索する．

7.3.3 仮想トランシーバ延長ケーブル

インターネットを利用して IEEE1394 ケーブルの物理長を仮想的に延長する DVTS というアプリケーションがある．DVTS は DV 用に IEEE1394 を仮想延長するものであるが，本研究を用いることで，同様の思想で仮想的にトランシーバ (WalkyTalky) を延長できるシステムを構築できる．

謝辞

本研究を進めるにあたり，主査である慶應義塾大学常任理事 村井純博士に感謝いたします．また，副査である慶應義塾大学環境情報学部教授 中村修博士，慶應義塾大学大学院政策・メディア研究科准教授 朝枝仁博士に感謝いたします．

また，絶えずご指導とご助言をいただきました，慶應義塾大学デジタルメディア・コンテンツ統合研究機構専任講師 斉藤賢爾博士，慶應義塾大学環境情報学部専任講師 重近範行博士，慶應義塾大学大学院メディアデザイン研究科准教授 杉浦一徳博士，慶應義塾大学 SFC 研究所上席所員 小川晃通博士に感謝いたします．

本研究を進めていく上で，様々な励ましと助言，お手伝いをいただきました，慶應義塾大学大学院政策・メディア研究科後期博士課程 堀場勝広氏，三島和宏氏，久松剛氏，松園和久氏，修士課程 金井瑛氏，空閑洋平氏，奥村祐介氏，共に春学期修士論文を頑張った，松谷健史氏，中村友一氏と徳田・村井・楠本・中村・高汐・重近・バンミーター・植原・三次・中澤合同研究プロジェクトおよびモービル広域ネットワークプロジェクトメンバの諸氏に感謝します．

最後に，いつ終わるとも分からない学生生活を，病気を乗り越えて応援し続けてくれた父親，そしてその父親を支えていた母親，一緒に応援してくれた祖父と，最後まで見届けてもらうことはできなかったもののずっと応援してくれた祖母に感謝します．

以上を持って謝辞といたします．

参考文献

[1] Twitter, Inc. Twitter. *http://twitter.com/*.

[2] Nintendo, Inc.. PictoChat. *http://www.nintendo.co.jp/ds/dpct/index.html*.

[3] Microsoft Corporation. MSN Messenger / Windows Live WWW page. *http://ideas.live.com/whatis.aspx*, 2006.

[4] Skype Limited. Skype WWW page. *http://www.skype.com/*, 2006.

[5] Polycom, Inc. Polycom WWW Page. *URL: http://www.polycom.com/home/*.

[6] A.Ogawa. *DVTS (Digital Video Transport System) WWW page*, November 2001. http://www.sfc.wide.ad.jp/DVTS/.

[7] Digital Living Network Alliance. DLNA. *URL: http://www.dlna.org/*.

[8] Digital Transmission Licensing Administrator. DTCP Volume 1 Supplement E Mapping DTCP to IP, Revision 1.2 (Informational Version). *http://www.dtcp.com/data/info%2020070615%20DTCP%20V1SE%201p2.pdf*, June 2007.

[9] J. Postel. User Datagram Protocol. RFC 0768, IETF, August 1980.

[10] J. Postel. Transmission Control Protocol. RFC 0793, IETF, September 1981.

[11] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889, IETF, January 1996.

[12] M. Handley, C. Perkins, and E. Whelan. Session Announcement Protocol. RFC 2974, IETF, October 2000.

[13] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, IETF, June 2002.

[14] The ITU Telecommunication Standardization Sector (ITU-T). H.245. *http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-H.245*, 2005.

[15] The ITU Telecommunication Standardization Sector (ITU-T). H.323. *http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-H.323*, 2005.

[16] Ed. S. Bhattacharyya. An Overview of Source-Specific Multicast. RFC 3569, IETF, July 2003.

[17] L. Clark and M. Angela Sasse. Conceptual Design Reconsidered: The Case of the Internet Session Directory Tool. In *HCI '97: Proceedings of HCI on People and Computers XII*, pages 67–84. Springer-Verlag, 1997.

[18] Hitoshi Asaeda, Wacharapol Pokavanich, and Soh Yamamoto. Channel reflector: An interdomain channel directory system. *IEICE Transactions*, 89-B(10):2860–2867, 2006.

[19] P. Namburi and K. Sarac. Multicast session announcements on top of SSM. In *IEEE ICC '04: Proceedings of International Conference on Communications*, volume 3, pages 1446–1450. IEEE, 2004.

[20] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *Procedings of USENIX Technical Conference*, 2004, pp. 127–140.

[21] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *Lecture Notes in Computer Science*, vol. 2218, pp. 329–350, 2001.

[22] Rongmei Zhang and Y. Charlie Hu. Assisted peer-to-peer search with partial indexing. *IEEE Trans. Parallel Distrib. Syst.*, 18(8):1146–1158, 2007.

[23] Reaz Ahmed and Raouf Boutaba. Distributed pattern matching: a key to flexible and efficient p2p search. *IEEE Journal on Selected Areas in Communications*, 25(1):73–83, 2007.

[24] Baruch Awerbuch and Christian Scheideler. Peer-to-peer systems for prefix search. In *PODC*, pages 123–132, 2003.

[25] James Aspnes and Gauri Shah. Skip graphs. *ACM Transactions on Algorithms*, 3(4), 2007.

[26] Ashwin R. Bharambe, Mukesh Agrawal, and Srinivasan Seshan. Mercury: supporting scalable multi-attribute range queries. In Raj Yavatkar, Ellen W. Zegura, and Jennifer Rexford, editors, *SIGCOMM*, pages 353–366. ACM, 2004.

[27] Matthew Harren, Joseph M. Hellerstein, Ryan Huebsch, Boon Thau Loo, Scott Shenker, and Ion Stoica. Complex queries in dht-based peer-to-peer networks. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *IPTPS*, volume 2429 of *Lecture Notes in Computer Science*, pages 242–259. Springer, 2002.

[28] Flemming Andreasen. *SDP Capability Negotiation*, December 2007. draft-ietf-mmusic-sdp-capability-negotiation-08.

[29] Robert R Gilman, Roni Even, and Flemming Andreasen. *SDP media capabilities Negotiation*, February 2008. draft-ietf-mmusic-sdp-media-capabilities-03.

[30] Flemming Andreasen, Gonzalo Camarillo, David Oran, and Dan Wing. *Connectivity Preconditions for Session Description Protocol Media Streams*, January 2008. draft-ietf-mmusic-connectivity-precon-04.

[31] Kenji SAITO. Wot for wat : Spinning the web of trust for peer-to-peer barter relationships(¡special section¿internet technology v). *IEICE transactions on communications*, 88(4):1503–1510, 20050401.

[32] Anwitaman Datta, Manfred Hauswirth, and Karl Aberer. Beyond "web of trust": Enabling p2p e-commerce. *cec*, 0:303, 2003.

[33] Martin J. Kollingbaum and Timothy J. Norman. Supervised interaction: creating a web of trust for contracting agents in electronic environments. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 272–279, New York, NY, USA, 2002. ACM.

[34] Philip Zimmermann. Pretty good privacy: public key encryption for the masses. pages 93–107, 1995.

[35] Jon Callas, Lutz Donnerhacke, Hal Finney, and Rodney Thayer. *OpenPGP Message Format*, November 1998. RFC 2440.

[36] Alfarez Abdul-Rahman. The PGP trust model. *EDI-Forum: the Journal of Electronic Commerce*, April 1997.

[37] The Free Software Foundation. The GNU Privacy Guard. Hypertext document. Available electronically at http://www.gnupg.org/.

[38] The Free Software Foundation. The GNU Privacy Handbook. Available electronically at http://www.gnupg.org/.

[39] Ken Birman, André Schiper, and Pat Stephenson. Lightweight causal and atomic group multicast. *ACM Transactions on Computer Systems*, Vol.9(No.3), August 1991.

[40] O'Reilly. What Is Web 2.0. Hypertext document. Available electronically at http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.