社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＱＡＩ－7　(8)
2003／5／21

親密度の高い発信者からの呼のみを着信させるVoIP通信システムの提案と実装

斉藤　潤†　奥田　剛†　横山　輝明†　山口　英†

†奈良先端科学技術大学院大学 情報科学研究科
〒630–0192 生駒市高山町 8916-5
E-mail: †{jun-sa,okuda,terua-yo,suguru}@is.aist-nara.ac.jp

あらまし　IP電話の普及に伴い，既存の電話以上に迷惑電話が増える可能性がある．既存の迷惑電話の対策手法では，明示的に許可されていない呼を拒否することが可能であるが，必要な呼までも着信を拒否してしまうという問題がある．インターネットを利用している電話の利点を活用できていないというのがこの問題の原因である．本研究では，インターネット特有の迷惑電話対策手法として，Web of Trustを用いた手法を提案する．この手法では，電話番号の伝搬により受信者と発信者の親密さが低下することを利用しており，着信者からの伝搬経路が長いほど親密度が低下し，その親密度が閾値以下になると着信を拒否するという方法を用いている．本論文では，迷惑電話対策手法への親密度の導入し，電話番号の伝搬経路長によって親密度が減衰する仕組みについて説明する．最後にプロトタイプシステムの設計と実装について説明し，提案手法の有効性と実現可能性についての議論を行っている．

キーワード　IP電話，迷惑電話，親密度，Web of Trust

A Nuisance Call Block System for VoIP Using Web of Trust

Jun SAITO†, Takeshi OKUDA†, Teruaki YOKOYAMA†, and Suguru YAMAGUCHI†

† Graduate School of Information Science, Nara Institute of Science and Technology
Takayama-cho 8916–5, Ikoma-shi, Nara, 630–0192, Japan
E-mail: †{jun-sa,okuda,terua-yo,suguru}@is.aist-nara.ac.jp

Abstract　Telephony systems are changing over from traditional PSTN to Voice over IP. In the VoIP environment, nuisance calls may increase due to the lower cost of VoIP and the ease of automatic calling system using computer. To block these nuisance call, we can make use of white list access control list. But this method blocks not only nuisance calls, but also necessary calls. To solve this problem, we introduced call blocking mechanism using notion of web of trust. In this mechanism, the longer the telephone number propagates, the less the familiarity between caller and callee becomes. In this paper, we explain the model of our call blocking mechanism for VoIP, and discuss the effectiveness of our method using prototype system.

Key words　VoIP, nuisance call, web of trust

1. はじめに

近年，各家庭がADSLや光ファイバーなどの常時接続が可能な広帯域ネットワークに接続され，身近にIPネットワークを利用したブロードバンドコンテンツを楽しめるようになった．最近では特に，通話料金が低価格であるインターネット電話（IP電話）が普及しつつあり，2005年に日本のIP電話は605万回線まで普及し，IP電話の市場は400億円以上になると予測されている[1].

しかし，IP電話機による通話が普及すると同時に電話の発信者に対して通話料が安いなどの特段の負担がかからないことからセールス電話や嫌がらせ電話などの迷惑電話が今よりも大幅に増加すると予測する．また，IP電話機はIPネットワークを用いることから計算機を用いた自動発信による宣伝電話などが増加すると考える．

既存のPSTNでは，迷惑電話を防ぐためにさまざまな迷惑電話の対策手法が提案されており，電話機や回線交換機などに迷惑電話対策機能が組み込まれている．IP電話でも，同様のアクセス制御を行うことが可能であり，こうした制御のなかで効果的と考えられているものに「着信許可機能」がある．ところがこの方式では自分の指定した電話番号以外の電話をすべて着信拒否するため，自分に必要な電話までも拒否してしまうという

問題が存在する．この問題を解決するには，呼を識別し，必要な呼を特定する必要がある．

本研究では，「着信許可機能」を実現することにより自分の知らない人からの迷惑電話を防止するとともに，自分に必要である呼を自動的に判別し，必要であると判断されれば着信許可リストに登録されていない相手からの呼も着信させる方式を提案する．提案方式を実現するために，本方式を用いたディレクトリサーバを実装し，SIP（Session Initiation Protocol）[2] ネットワークに適用する．また，実生活におけるさまざまな事例検証を行うことでシステムの有効性を検証する．

2. 親密度を用いた着信制御システムの提案

2.1 既存の着信制御システム

既存の PSTN で使われている迷惑電話対策のための着信制御手法として，指定着信というものがある．これは，電話帳あるいは指定着信リストに明示的に指定された電話番号からの呼しか着信を許可しないというものである．指定されたものしか着信できないので，迷惑電話対策として有効である．しかし，いかなる場合も明示的に指定された相手からの呼しか受け取れないため，あらかじめ登録されていない発信者からの呼は着信できないなど，柔軟性と利便性に欠ける．

2.2 親密度の導入

上記の問題を解決するために，指定着信機能に登録された情報と，親密度という情報を用いて，着信者にとって必要な呼を識別する手法を提案する．親密度とは，電話の世界における人と人との間の親密さを数値化したものである．人と人との間の親密さを用いて呼を着信させる手法は Web of Trust [5] の概念を利用し，自分との親密度が高い人を信用できる人として電話の着信を受けつける．これは，個人が個人を信用する相互信用モデルである．個人は，信用先が信用する相手をも信用し，信用の鎖を繋いでいく．このように信用の連鎖で相互信用の範囲を拡大し，実社会での人間同士の信頼関係形成のモデルに類似した方法で呼が着信者にとって必要かどうかの判断を行う．

自分の着信許可リストに登録されていない発信者から呼を受けた場合，その発信者へ自分の電話番号を伝えた人との信用関係によって，その発信者からの電話を迷惑電話かどうか判断する．親密度の低い友人が別の友人に電話番号を伝えた場合，その別の友人の親密度ははさらに低くなり、呼が着信する可能性が下がることになる．また，その発信者に電話番号を伝えた友人・知人がいない場合，親密度は皆無であるため，呼が着信することはない．

2.3 電話番号を伝えた相手の特定

電話番号の伝搬経路によって受信者からみた発信者の親密度が変わるため，発信者に電話番号を教えた相手を特定する必要がある．

自分の着信許可リストに登録されていない発信者からの呼には，発信者の名前や発信者の IP アドレスという情報のみが含まれており，電話番号を誰から教えてもらったかを表す情報は含まれない．そこで，電話番号を他人に伝えるとき，誰に教えたか，誰に教えてもらったかという情報を残しておく．この情報を発信者から逆にたどることで，発信者は誰から自分の電話番号を入手しているのかを知ることができる．

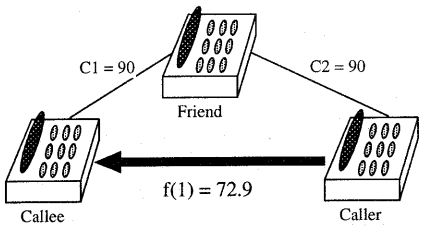

図 1　三人の時における親密度の計算

2.4 親密度の計算

次に，着信許可リストに登録されていない発信者には親密度が設定されていないため，電話番号を伝えた人の親密度を利用して発信者の親密度を計算する．電話の世界において，着信許可リストに登録されていない相手は，登録されている相手と比較すると自分との親密度が低いと考えることができる．また，電話番号を又聞きにより取得した場合，その伝搬経路のホップ数に比例して，親密度を希釈させる必要がある．そこで親密度を式（1）により算出する．

$$f(x)=\frac{\Pi_{k=1}^{x+1}[\{(1-\alpha)^{x-1}\}C_k]}{100^{(x+1)}} \tag{1}$$

式（1）では，自分の電話番号がより多くの人を経由して伝えられると，その人数に応じて親密度がより低くなる．x は自分の電話番号が x 人を経て伝達されたことを表す．着信許可リストに登録されている相手を $k=1$ とし，発信者に自分の電話番号を伝えた人を $k=x+1$ とする．C_k は自分の電話番号を又聞きした人同士の間の親密度および自分から自分の電話番号を伝えた人との間の親密度である．

α は 0 から 1 までをとる希釈係数であり，個人が設定できるものとする．$\alpha=0$ の場合，単純に親密度の乗算になり，希釈しない．また，$\alpha=1$ の場合，自分の着信許可リストに登録されていない発信者は親密度が 0 になり，着信しない．

例えば，希釈度を $\alpha=0.1$ に設定し，すべての人との親密度を $C_k=90$ と仮定する．自分の友人が電話番号を別の友人に伝えると．そのときの別の友人から自分への呼の親密度は式（1）より $f(1)=72.9$ と計算される（図 1）．

2.5 親密度の判定

自分の着信許可リストに登録されていない電話番号からの呼では，呼の親密度の高低を判断するために親密度の閾値を設ける．自分に着信する呼の親密度と着信を許可する閾値を比較して着信の可否を判断する．

2.6 親密度露見への対処

この方式では親密度の露見に留意しなくてはならない．これは第三者を介して，特定の相手から自分への親密度が露見する問題である（図 2）．例えば，自分への親密度を知りたい特定の相手の電話番号を自分の友人に教え，その友人に電話を特定の相手に電話を発信してもらう．この時，自分から友人への親密度を十分高く設定しておく．もし，友人が発信した電話が特

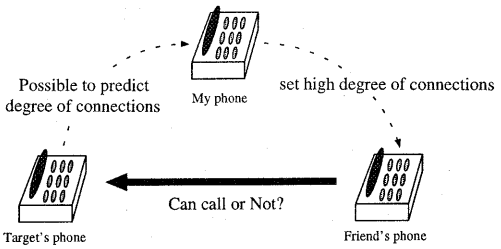

図 2　親密度が露見する例

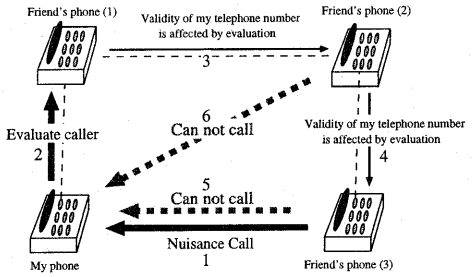

図 3　自分の電話番号を伝えた人の評価

定の相手に着信した場合，特定の相手から自分への親密度は高いと推測することができる．逆に，友人の発信した電話が特定の相手に着信しなかった場合，特定の相手から自分への親密度は低いと推測することができる．

本提案手法では，親密度の推測を困難にするために二種類の手法を提供することが可能である．一つ目は自分の着信許可リストに登録されていない人から電話がかかってきた場合，通話が終わった後に自分の電話番号を電話の発信者に伝えた人の評価をすることができるシステムである．二つ目はユーザの電話機の使用に応じて動的に親密度を変化させる方法である．

前者は通話終了後に自分の電話番号を伝えた相手を「非常に良い」,「良い」,「普通」,「悪い」,「非常に悪い」の 5 段階で評価する．その友人から自分の電話番号を教えてもらった人は、この影響を受け同様に呼の親密度が下がる．これで、自分の電話番号を不特定多数の人に伝えた友人から伝播した自分の電話番号を無効化することができる（図 3）．自分と自分の電話番号を伝えた相手との間の親密度は大幅に変動し，二人の間の親密度を推測することが困難になると考えられる．

後者は，ユーザの動作に応じて動的に親密度を変化させる方法である．例えば，自分から友人に電話をかけた場合，自分から友人に対する親密度を自動的に上昇させる．自分から相手への親密度が上昇するのは以下の場合であると考えられる．

- 相手からの電話の受話回数
- 自分から相手への発信回数
- 自分の着信許可リストに登録されていない電話番号からの電話で内容が自分にとって良かった場合

3.　提案システムの実装

提案手法を今後 IP 電話の標準プロトコルになると考えられる SIP に適用する．本章では、提案システムの設計とその実装の詳細について説明する．

3.1　提案システムの設計

提案システムの動作要件は以下の四点である．

- 自分の着信許可リストに登録されている人からの電話のみを着信させる
- 自分の着信許可リストに登録されていない相手からの電話は発信者からの呼の親密度を計算し，自分との親密度が高い場合のみ呼を着信させる
- 他人に自分の電話番号や他人の電話番号を伝えた場合，自分が伝えたという記録を残し、否認を防止する
- 個人のプライバシーを保護するために親密度を他人から隠蔽する．

これらの要件を満たすためには．ディレクトリサーバの設置位置を考慮する必要がある．SIP ネットワークアーキテクチャでディレクトリサーバを設置する位置として，

- ディレクトリサーバの機能を各端末に組み込む
- ディレクトリサーバの機能を各 SIP プロキシサーバに設置する
- ディレクトリサーバの機能を DNS サーバに設置する
- ディレクトリサーバをネットワークに独立して設置する

の 4 種類が考えられる．

各 UA が各人の電話帳ファイルを保持している場合，親密度の計算のために各端末に親密度の問い合わせを行う必要がある．ネットワーク上に親密度に関する情報が飛び交うため，容易に親密度を知ることが可能である．各人の電話帳ファイルを各人の SIP プロキシサーバに設置する場合も，上記の場合と同様にネットワーク上に親密度がネットワーク上を流れることになるが，SIP プロキシサーバの運用や設置形態によっては安全な場合もある．電話帳ファイルを DNS サーバが保持している場合，電話帳が分散しているため，自分や相手の DNS サーバで他人から他人への親密度が露見する．SIP ネットワークにディレクトリサーバを独立して設置する場合，ディレクトリサーバにアクセスする処理の段階により，既存の SIP アーキテクチャを大幅に変更する必要が生じる．自分の電話番号を電話の発信者に伝えた人の情報および自分の電話番号を伝えた人と発信者との間の親密度を呼に付加しなければならないため，さまざまな場所で親密度が露見する．ロケーションサーバの一部の機能としてディレクトリサーバの機能を提供する場合，着信側の SIP プロキシサーバはロケーションサーバに UA の IP アドレスを問い合わせ，問い合わせと同時に発信者および着信者の情報をディレクトリサーバに送信する．ディレクトリサーバ内部で呼の親密度を計算するため，SIP ネットワークの大きな変更を必要としない．また，ディレクトリサーバ内部のみで親密度の計算をするため，親密度は露見しない．

これらの考察より，ディレクトリサーバをロケーションサーバに設置するのが適切だと判断した．考察の結果を表 1 にまと

表1　ディレクトリサーバ設置場所の一覧

ディレクトリサーバ設置場所	親密度が露見しない
各端末に組み込む	×
各 SIP プロキシに設置	△
DNS サーバに設置	△
ネットワークに独立して設置	△
着信側のロケーションサーバに設置	◎

める．

以上の結果より，本研究では各人の電話帳ファイルは分散管理型ではなく，ディレクトリサーバにすべての人の電話帳ファイルを保存する集中管理型が適していると判断した．集中管理することにより，親密度の計算をディレクトリサーバ内部のみで処理することが可能となり，親密度の露見を防止することが可能となる．また，ディレクトリサーバに着信側のロケーションサーバがアクセスすることで，着信側の SIP プロキシサーバに所属しているすべての IP 電話機がこの提案システムを利用可能になる．

このディレクトリサーバには SIP メッセージに含まれている送信先と送信元の情報のみを送信し，UA の情報や SIP プロキシの情報などは送信しない．このことにより，一般的な SIP ネットワークアーキテクチャに変更を加えることなく，ディレクトリサーバにデータを受け渡すことが可能である．本システムを適用した SIP ネットワークは図 4 のようになる．

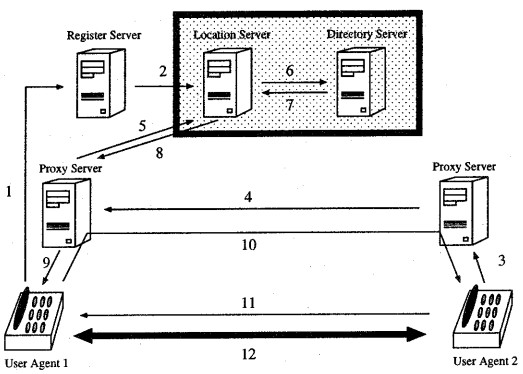

図 4　SIP ネットワークにおける呼の流れ

本システムでの呼の処理は以下となる．

（1）　レジスタサーバに UA1 の端末情報を送信

（2）　レジスタサーバからロケーションサーバに UA1 の端末情報を登録

（3）　UA2 が UA1 に接続するために UA2 の SIP プロキシサーバに接続

（4）　UA2 の SIP プロキシサーバは UA1 の SIP プロキシサーバにメッセージを転送

（5）　UA1 の SIP プロキシサーバはロケーションサーバに UA1 の IP アドレスを問い合わせる

（6）　ディレクトリサーバに発信者と着信者の情報のみ送信し，ディレクトリサーバ内部で呼の親密度を計算

```
A, A, A;B;C, 100, 100
B, B, , 80, 50
C, B, D, 60, 50
```

図 5　電話帳パラメータの例

（7）　ディレクトリサーバからの応答メッセージ

（8）　ロケーションサーバからの応答メッセージ

（9）　UA1 に UA2 からのメッセージが到着

（10）　UA1 から UA2 に確認応答メッセージを送信する

（11）　UA2 から UA1 に接続要求メッセージを送信する

（12）　音声による通話の開始

ディレクトリサーバは SIP プロキシサーバから送信される発信者と着信者の情報から各人の電話帳ファイルを参照し，親密度を計算する．親密度が十分に高ければ，ディレクトリサーバはロケーションサーバに呼を送信する．(7 番目の処理) ロケーションサーバは SIP プロキシサーバに UA の IP アドレスを送信する．(8 番目の処理) もし，呼の親密度が低ければ，ディレクトリサーバはロケーションサーバに呼を送信せず，ディレクトリサーバ内部で呼を破棄する．

3.2　提案システムにおける実装の詳細

以上の設計に基づき，プロトタイプを実装し，動作の確認を行った．本研究では，IP 電話機として Linphone [6] を用いた．また，レジスタサーバ，SIP プロキシサーバ，SIP ロケーションサーバとして，WellX Telecom 社 [7] が作成した party sip [8] を用いた．

ディレクトリサーバ内部で保存している各人の電話帳ファイルは以下の 5 つのパラメータを保持している．

- 相手の電話番号
- 相手の電話番号を他人へ通知した人
- 相手の電話番号を他人から通知された人
- 相手への親密度
- 相手の電話を着信許可する閾値

例えば，User Agent である A の電話帳パラメータを図 5 に示す．

1 番目のパラメータは「相手の電話番号」であり，最初の行は自分の電話番号が記述されている．自分の電話番号の記述は他人に自分の電話番号を教える際に必要である．2 番目のパラメータは「相手の電話番号を他人へ通知した人」が記述してあり，自分の電話番号は初期値として，自分自身が記述してある．3 番目のパラメータは「相手の電話番号を他人から通知された人」が記述してあり，自分の電話番号を得た人は，A, B, C の 3 人である．4 番目のパラメータは「相手への親密度」を表しており，自分から自分への親密度は初期値として 100 が設定される．5 番目のパラメータは「相手の電話を着信許可する閾値」であり，自分から自分への着信許可する閾値は初期値として 100 が設定される．

ロケーションサーバが SIP プロキシサーバからメッセージを受信すると，cb_rcv_udp_message() 関数が呼び出される．この

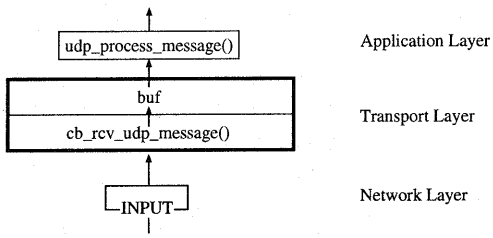

図 6　通常のロケーションサーバの処理

表 2　開発環境

OS	RedHat Linux 8.0 (kernel 2.4.18-14)
言語	C 言語
コンパイラ	gcc version 3.2
ライブラリ	oSIP

表 3　作成した関数一覧

関数名	内容
check_invite()	SIP メッセージが INVITE メソッドを含んでいるかチェックする
get_to_from()	SIP メッセージから To および From の SIP URI を取り出す
check_from()	To のアドレスを誰から教えてもらったかチェックする
check_to()	To の電話帳に From が登録されているかチェックする
calc_to_from()	From から To への親密度を計算し，To の閾値と比較する

関数は受信した UDP メッセージを受信バッファに格納し，SIP メソッドを理解させるためにアプリケーション層にデータを受け渡す処理を行う．通常のロケーションサーバの処理を図 6 に示す．

party sip は，SIP プロキシサーバ，レジスタサーバおよびロケーションサーバの機能を提供することができる．本研究では，party sip のロケーションサーバを改造し，INVITE メソッドを判別する関数を追加した．本研究では表 2 に示す開発環境でディレクトリサーバの実装を行った．

親密度を用いた IP 電話システムを実現するために，いくつかの関数を作成する．作成した関数の一覧および関数の説明を表 3 に、関数を追加したロケーションサーバの処理を図 7 に示す．

ロケーションサーバで cb_rcv_udp_message() 関数が呼び出された後に作成した check_invite() 関数により受信バッファに格納されている SIP メッセージが INVITE メソッドであるかどうかを調査する．もし，INVITE メソッドでない場合はそのままメッセージを通過させ，アプリケーション層にデータを受け渡す．

受信した SIP メッセージが INVITE メソッドの場合は SIP メッセージを受信バッファに格納し，次の処理に移行する．INVITE メソッド以外の SIP メッセージに手を加えないことで INVITE メソッド以外の通常の SIP メッセージをロケーションサーバに反映させることが可能である．

ロケーションサーバが INVITE メソッドを受信した場合，ディレクトリサーバの get_to_from() 関数を呼び出す．get_to_from() 関数はバッファに格納されている SIP メッセージの To と From の SIP アドレスを抽出する関数である．

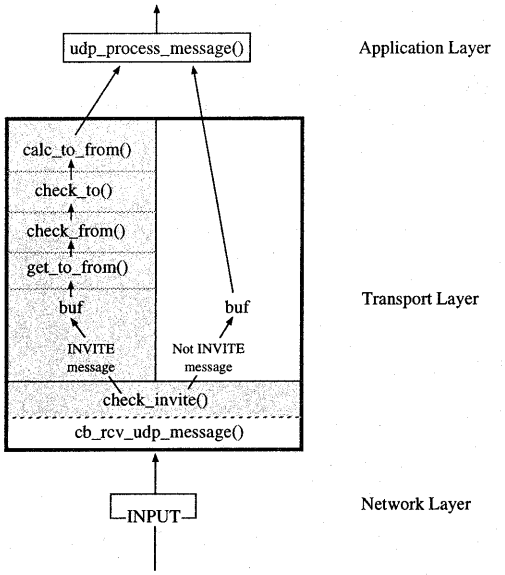

図 7　関数を追加したロケーションサーバの処理

To と From の SIP アドレスの抽出に成功した場合，次に check_from() 関数および check_to() 関数を実行する．check_from() 関数は誰から着信者の電話番号を得たかという情報を着信者に到着するまで再帰的に検査する．また，check_to() 関数は着信者の電話帳に電話の発信者が登録されているか検査する．

check_from() 関数により正しく着信者の本人から伝播した電話番号を得たことを確認できた場合，次に calc_to_from() 関数が実行される．calc_to_from() 関数は着信者から発信者への親密度を計算する．親密度の計算には式 (1) を用いる．

自分に着信する電話の呼の親密度が自分の設定した閾値より高い場合，ディレクトリサーバで受信した SIP メッセージをロケーションサーバに受け渡す．逆に，自分に着信する電話の呼の親密度が自分の設定した閾値よりも低い場合，ディレクトリサーバの SIP メッセージをロケーションサーバに送信せず，ディレクトリサーバ内部で SIP メッセージを破棄する．この処理により，着信者の電話機に呼を着信させる前に呼の処理をすることが可能となる．

4.　提案システムの検証

本研究の提案システムを用いることで，着信許可リストに登録されていない相手からの電話を着信することが可能になる．このときの提案システムの動作を検証する．

電話の着信者として自分がいる．電話の発信者と他に 3 人の知り合いがおり自分の電話番号を連鎖して知り，自分に電話をかける形態を考える．隣り合った人は互いに知り合いで電話の

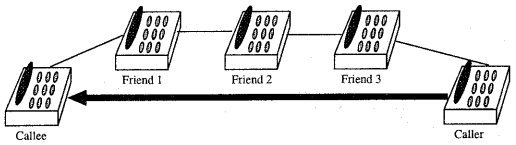

図 8　5 人での通話形態の例

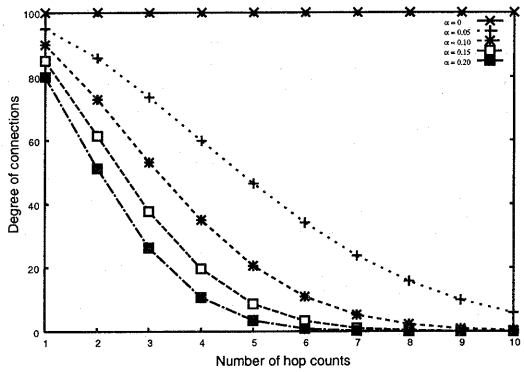

図 9　親密度 100 における親密度の推移

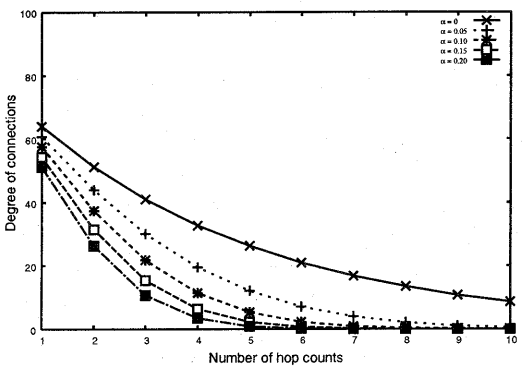

図 10　親密度 80 における親密度の推移

発信者は自分の電話帳に登録されていない（図 8）.

隣接間の親密度を 80，希釈度を 0.1 に設定する．その場合の発信者から着信者への呼の親密度は式（1）より約 21.8 と計算される．発信者からの電話を着信させるためには，着信者は 22 未満に閾値を設定する必要がある．隣接間の親密度を 100 に設定し，希釈度を 0 に設定した場合，発信者から着信者への呼の親密度は式（1）より 100 と計算される．この場合，呼の親密度の計算に希釈度は全く影響しないため，何人の又聞きを経ても親密度は 100 のままとなる．

隣接間の親密度を 100 に設定し，希釈度をそれぞれ (0, 0.05, 0.1, 0.15, 0.2) に設定した場合，発信者から着信者への呼の親密度の推移を図 9 に示す．隣接間の親密度を 80 に設定し，希釈度をそれぞれ (0, 0.05, 0.1, 0.15, 0.2) に設定した場合，発信者から着信者への呼の親密度の推移を図 10 に示す．

これらのグラフより，人の経由数で親密度は大きく希釈する事が分かる．これは，人を経由する初期段階での希釈度が大きく影響し，親密度が大幅に減少することを表している．希釈度を低く設定するほど親密度の減衰は小さくなり，希釈度を高く設定するほど親密度はより減衰する．希釈度を 0 に設定した場合は高い親密度のまま遠くの人に伝播しているのが分かる．希釈度を最高の 1 に設定することにより，自分が電話番号を伝えた人のみから着信するようになり，自分の電話番号を又聞きした人から着信しないようになる．

このように希釈度のパラメータによって、親密度をどれだけ遠くまで高く保つかを利用者自らが制御することができる．

5.　まとめと今後の課題

本研究では，VoIP 環境では迷惑電話が増加することを予想し，迷惑電話対策手法の一つである「着信許可機能」に注目し，「着信許可機能」の問題点である，自分に着信が必要な呼までも排除してしまう問題を Web of trust の技術を用いて解決する手法を提案した．

しかし，提案システムにはいくつかの欠点が存在する．例えば，ディレクトリサーバ上で呼の親密度を計算し，呼の親密度が低かった場合，ディレクトリサーバ上で呼を破棄するため，相手の端末に発信者からの着信履歴が残らないという欠点が存在する．この欠点は，自分の着信許可リストに登録されていない相手や親密度の低い相手からの呼を自分の電話機に着信させないほうが望ましいという点を優先させた為に起きる問題である．ディレクトリサーバ上で呼を破棄した場合，留守番電話サービスに転送するなどの運用で解決が可能である．また本システムを実際に運用するには、サーバの信頼性の向上や、サーバ内での呼の処理負荷への対応などの課題が考えられる．

文　献

[1] Voice On the Net Japan, Conference Note P.27, December 4, 2002
[2] J. Rosenberg, H. Schulzrinne. “SIP: Session Initiation Protocol”. RFC 3261, June 2002
[3] J. Moy. “OSPF Version 2”. RFC 2328, April 1998.
[4] Welcome to the ORDB.org - the Open Relay DataBase. http://www.ordb.org/
[5] Trust models - Web of trust. http://www.pgpi.org/doc/pgpintro/#p20
[6] Welcome to Linphone.org, Telephony on Linux. http://www.linphone.org/
[7] WellX Telecom S. A. http://www.wellxusa.com
[8] The partysip SIP proxy server. http://www.partysip.org/