2008－UBI－20 (2)
2008／11／13

サービス利用に必要なノードを自動的に追加する動的VPN管理機構

榎堀 優† 新井イスマイル†† 西尾 信彦†††

† 立命館大学大学院理工学研究科
†† 立命館大学総合理工学研究機構
††† 立命館大学情報理工学部

あらまし ノード単位にアクセス制御が可能なVPNは，複数のネットワークに分散する任意のノード群の間に接続性を与えたい場合などに有用であるが，必要なノードすべてに対して逐一手動で接続設定しなければならない煩雑さに課題がある．我々は，この種のVPNの一実装であるPeer Poolを拡張し，サービス利用に必要なノードを自動的に接続可能にするシステムを提案する．提案するシステムは，名前解決を契機として，当該名前によって参照されるノードと名前解決を要求したノードの間に接続性を与える．また，提案システムを実装し，実用に耐える応答時間で動作することを確認した．

キーワード アクセス制御，VPN, DNS

A Dynamic VPN Management System with Automatic Connectivity Configurations

ENOKIBORI Yu†, ARAI Ismail††, and NISHIO Nobuhiko†††

† Graduate School of Sicence and Engineering, Ritsumeikan University
†† The Research Organization of Science and Engineering, Ritsumeikan University
††† College of Information Science and Engineering, Ritsumeikan University

Abstract We have been developing Peer Pool, a VPN technology that links an arbitrary set of nodes placed in separate private IP networks. Such kind of VPNs are useful when the user wishes to set up per-node connectivities among those networks; however, they require manual per-node configurations. In this paper, we propose an extended Peer Pool that automatically links the nodes in need. The proposed system hooks DNS queries and configures connectivities between the requester and the node with the queried domain name. We confirmed that an implementation of the proposed system works with a reasonable response time.

Key words Access control, VPN, DNS

1. はじめに

家庭やオフィスに遍在したデバイスを用いてユーザに有益なサービスを提供するスマート環境が提唱されている．スマート環境の普及にともない，複数のスマート環境間でサービスを連携利用する需要が生まれると考えられる．多くのスマート環境が，外部からの通信が制限されたプライベートネットワークに構築されると考えられる．これらの間でサービスを連携利用するためには，何らかの技術を用いて通信経路を確保する必要がある．

現在は，遠隔のプライベートネットワーク間に通信経路を確保するために各種のVPN（Virtual Private Network）が利用されている．それらの中で，ノード単位にアクセス制御が可能なVPN（以下，ノード単位VPN）の利用が，セキュリティ確保などの面から注目されている[1], [2]．しかし，既存のノード単位VPNでは，接続性が必要なノードすべてについて逐一手動で接続性を与える設定をする必要があり，煩雑である．例えば，Webページ中に他のプライベートなノードへのハイパーリン

クが多数あり，リンク先のノードもVPNを介して利用したい場合に，クリックされるリンクを予期して接続設定することが困難である．

本研究は，ノード単位VPNにおいて，サービス利用に必要なノード群に自動的に接続性を与えることを課題とする．本稿では，我々が開発してきたノード単位VPN技術であるPeer Pool [1] を拡張することによって，この課題を解決するシステムを提案する．提案システムは，名前解決を契機として，当該名前によって参照されるノードと名前解決を要求したノードの間に接続性を与える．また，これを実装して稼動させ，実現可能性を確認した．

本稿は全7節で構成される．以下，2節で本研究を動機付ける問題意識について述べる．3節で我々が開発してきたVPN技術について説明し，4節でこれを拡張して問題解決を図るシステムを提案する．5節で提案システムの実装と評価について述べる．6節で本研究を関連研究と比較し，7節でまとめる．

2. ノード単位VPNの有用性と問題点

2.1 ノード単位VPNの有用性

VPN技術は，その接続形態により2種類に大別される．一つはあるネットワーク上の空間全体に対して一様に接続性を提供するものであり，また一つはノード単位に接続性を制御可能なもの（**ノード単位VPN**）である．

任意のノード群に動的に接続性を与える必要があり，且つ無関係なノードに及ぼす影響を最小限に抑えたい場合，ノード単位VPNが有用である [1], [2]．例えば，共同でプロジェクトを運営する複数の組織が定例で遠隔会議を開催するシナリオを想定する．各組織がそれぞれにスマート環境を持ち，それらに属するノード群の連携によって遠隔会議支援システムを構成できるとする．この場合，会議の時間中に限り，遠隔会議支援システムを構成するノード群のみに接続性を与えられればよい．ノード単位VPNを利用すれば，遠隔会議支援システムと無関係なノードへの影響を排除した安全な連携が可能になる．

2.2 ノード単位VPNの現状

ノード単位VPNにおける問題点は，接続設定の煩雑さである．既存のノード単位VPN技術では，接続性が必要なノードすべてについて逐一手動で接続性を与える設定をする必要がある．例えば，先の遠隔会議支援システムの例において，プロジェクトに参加する組織は互いに信頼関係を築いており，一組織のイントラネットのWebサイトを他組織のプロジェクトメンバーに公開したいとする．この場合，アクセスするユーザ端末とWebサーバをそれぞれノード単位VPNのネットワークに参加させればよい．しかし，このWebサーバが提供するHTML文書中にプライベートネットワーク内の他ノードへのリンクがある場合，それらを利用するためには個々のリンク先のノードについても事前に接続設定をしなければならない．HTML文書中でクリックされるリンクをすべて事前に把握するのは困難であり，このような設定をすべて手動で行うことは現実的でない．

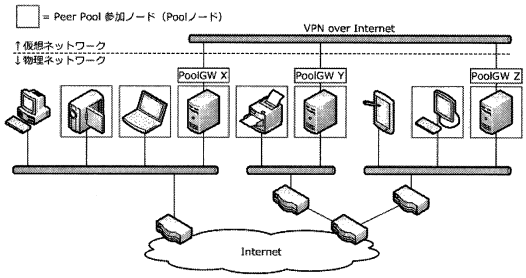

図1　Peer Poolの構成例

以上の問題意識により，本研究は，ノード単位VPNにおいてサービス利用に必要なノード群に自動的に接続性を与えることを課題とする．

3. Peer Pool

我々は，ノード単位VPNの一実装としてPeer Poolを開発してきた．本研究ではこれを拡張する．そのため，本節ではPeer Poolの動作を概説する．

3.1 Peer Poolの概要

Peer Poolは，DNSクエリをインタフェースとしてIPレイヤの接続性を適時構成・管理するシステムである．Peer Poolを利用するにあたっては，Peer Poolの機構を導入したホスト（**PoolGW**）を各ネットワークに1台ずつ設置し，それらをレイヤ3以下のVPNで接続する（図1）．PoolGWと同一ネットワークセグメントに存する任意のノードは，一定様式のDNSクエリをPoolGWに送ることによってPeer Poolに参加することができる．PoolGWは，Peer Poolに参加したノード（**Poolノード**）群に対し，VPNの通信路を介した接続性を提供する．

Poolノード間の通信手法は2種類あり，目的に応じて使い分ける．それぞれの詳細を以下に述べる．

3.2 Poolアドレスによる通信

各Poolノードには，仮想的なIPアドレス（**Poolアドレス**）と`.pool`で終わるホスト名（**Poolノード名**）が割り当てられる．PoolGWは，Poolアドレスと実アドレスの間に双方向NATを適用し，Poolアドレス宛のパケットを他のPoolGWに転送する．また，PoolGWは，Poolノード名からPoolアドレスを解決するDNSサーバの機能を持つ．

以上の設計により，Poolノードにおいて通信を利用

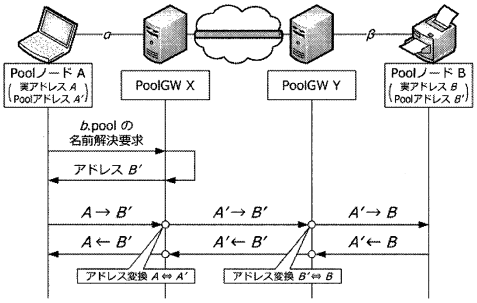

図 2　Pool アドレスによる Pool ノード間の通信

するアプリケーションの宛先に *a-pool-node-name*.pool なるホスト名を指定することによって当該ノードへのアクセスが可能である（図 2）．なお，pool ゾーンの名前解決要求と Pool アドレス宛のパケットがローカルネットワーク内の PoolGW に到達するよう，Pool ノードまたはルータ等において DNS サーバと経路表の設定を変更する必要がある．

3.3　仮想ローカルノードによる通信

Peer Pool では，Pool アドレスによる通信のほか，仮想ローカルノードを利用した通信（図 3）が可能である．仮想ローカルノードは，ローカルネットワークの空きアドレスを用いた ARP の代理応答による，遠隔の Pool ノードの Proxy である．仮想ローカルノードを利用すると，Pool アドレス宛の特別な経路設定が不要になる．

図 3 は，Pool ノード A が Pool ノード B に対し，仮想ローカルノードを利用した通信を開始するときの処理手順を示す．まず，Pool ノード A は，Pool ノード B の Pool ノード名のサフィックス.pool を.local.pool で置換した名前の解決を PoolGW X に要求する．この要求を受けた PoolGW X は，ネットワーク α の空きアドレス B'' を返答し，以後，B'' に対する ARP に代理応答することによって B'' 宛のパケットを受け取る．また，この要求は PoolGW Y にも通知され，以後 PoolGW Y は，ネットワーク β の空きアドレス A'' 宛のパケットを ARP の代理応答によって受け取る．Pool ノード A から B'' 宛のパケットは，2 度のアドレス変換を経て Pool ノード B に到達する．同様に，それに対する返答のパケットも 2 度のアドレス変換を経て Pool ノード B から Pool ノード A に到達する．

以上により，Pool ノードにおいて通信を利用するアプリケーションの宛先に *a-pool-node-name*.local.pool なるホスト名を指定することによって当該ノードへのアクセスが可能である．

3.4　Peer Pool の問題点

2.2 節で述べた問題を，先の遠隔会議のシナリオを Peer Pool で実現した場合について具体的に説明する．

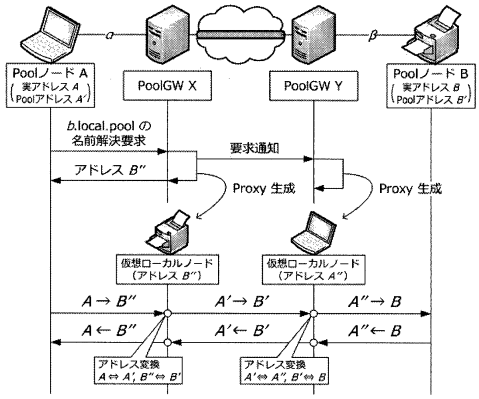

図 3　仮想ローカルノードを利用した Pool ノード間の通信

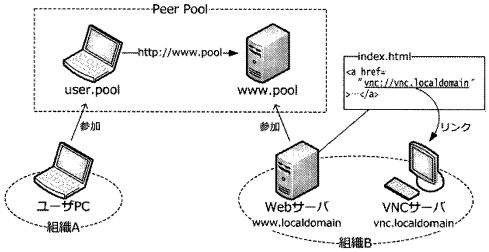

図 4　Peer Pool でリンク先にアクセスできない問題

図 4 において組織 B のスマート環境では，複数のノードからなる会議支援システムが稼動しているとする．会議支援システムを構成するノードは，議事録を共有するためのファイルサーバやプレゼンテーションを見るための VNC サーバなどを想定している．また，これらの個々のサービスへのリンクを含んだ会議支援システムポータルサイトを提供する Web サーバが存在する．組織 B 内のユーザは，このポータルサイトからリンクをたどって個々のサービスを利用する．

組織 A のユーザが Peer Pool を介して組織 B の会議支援システム利用し，両組織間で遠隔会議を開催する場面を考える．組織 B のユーザが Web サーバを Peer Pool に参加させると，組織 A 内の Pool ノードから会議支援システムポータルサイトを参照することができる．しかし，組織 A 内のユーザがポータルサイトからリンクされた個々のサービスを利用するためには，リンク先のノードについても逐一 Peer Pool に参加させなければならない．このようなサービス利用に必要なノードを自動的に参加させられれば，遠隔のスマート環境間でサービスを相互に利用しあうときの利便性が向上すると考えられる．

4.　名前解決を契機としたノードの自動追加

本節では，前節で述べた Peer Pool を拡張し，名前解

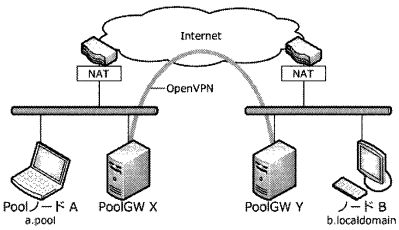

図 5 実験環境

決を契機としてノード間に接続性を与えるシステムを提案する．

我々は，ドメイン名を用いた典型的な IP ネットワークの利用例では，ノードにアクセスする直前にそのノードの名前解決要求が発生することに注目した．そこで，名前解決を要求したノードと解決された答のノードの間に自動的に接続性を与えることを試みる．また，PoolGW が DNS サーバ機能を含むので，Peer Pool はこのようなアプローチと相性がよい．

従来の PoolGW の DNS サーバ機能は，解決すべき名前が未知のとき，サフィックスが

（1） `.pool` ならば VPN 接続された PoolGW に，

（2） それ以外ならば上位 DNS サーバに

それぞれ問合せを転送することによって Pool アドレスおよび通常の IP アドレスへの解決を図っていた．

提案システムは，（2）のときに上位 DNS サーバだけでなく VPN 接続された PoolGW にも問合せを転送することによって，グローバルに公開されていない名前の解決を図る．転送先の PoolGW は，再び名前解決を試みて得られた答のノードが，

（1） グローバルに公開されていないノードであり，

（2） Peer Pool に参加していないが，

（3） 参加すれば要求元との接続性が得られる

ならば，当該ノードを自動的に Pool ノード化した上で成功を返答する．さらに，転送元と転送先の PoolGW 間で，前述した仮想ローカルノードによる通信の設定も行う．

5. 実装と評価

提案システムを，Linux（Ubuntu 8.04）を OS とするホストを動作環境として，Java 6.0 で実装した．PoolGW 間の VPN は，OpenVPN 2.1 [3] による L2VPN で構成した．

5.1 動作の検証

提案システムの現実性を検証するため，本実装を用いて図 5 に示すネットワークを構築した．この環境において，Pool ノード A 上で “`ping b.localdomain`” を実行し，本実装が期待通りに動作することを確認した．すなわち，`b.localdomain` の名前解決を契機としてノード A が Peer Pool のネットワークに追加され，これに続く A–B 間の通信が成立した．本実験時の処理の流れを図 6 に示す．

表 1 実験環境のネットワーク遅延 (ms)

	最小	最大	平均
PoolGW 間 RTT	31.56	38.08	33.05

表 2 `b.localdomain` の解決の応答時間 (ms)

	最小	最大	平均
自動追加を伴うとき	225	413	278
キャッシュ失効後	96	312	413

本実験では，Pool ノード A からの `b.localdomain` の解決要求を契機としてノード B が Peer Pool のネットワークに自動的に追加される処理を行った．まず，Pool ノード A から，ドメイン名 `b.localdomain` の A レコードを PoolGW X に問い合わせを行った．この問合せは，PoolGW X によって上位 DNS サーバおよび PoolGW Y に転送される．PoolGW X からの問合せを受けた PoolGW Y は，自分自身のレコードの参照または上位 DNS サーバへの問合せによって `b.localdomain` を解決し，ノード B のアドレス B を得る．B がプライベートアドレスであることを確認した PoolGW Y は，Pool アドレス用のアドレス空間から得たアドレス B' と B の間に双方向 NAT ルールを生成し，B' を返答する．なお，PoolGW X から PoolGW Y への問合せには OpenVPN のブロードキャストを利用しているので，3 箇所以上のネットワークで Peer Pool を構成しても同様に動作する．一方，PoolGW X の上位 DNS サーバは，`b.localdomain` がグローバルに公開されていない名前であるため解決に失敗する．以上の処理を待って PoolGW X は，成功を返答した PoolGW Y の答を採用する．最終的に PoolGW X が，追加された Pool ノード B を仮想ローカルノード化し，その仮想ローカルアドレス B'' を返答する．その返答を Pool ノード A が受け取り，そのアドレスへアクセスすることで通信が確立された．

5.2 応答速度の測定

DNS サーバは，クライアントがタイムアウト（例えば，代表的な DNS クライアントプログラムである nslookup はデフォルトで 2000 ms）する前に応答を返す必要がある．提案システムにおける名前解決は，追加処理を伴うため一般の DNS サーバより応答時間が長いと考えられる．そこで，これが実用上問題とならないかを検証するため，図 5 に示した環境で本実装の応答時間を測定した．なお，PoolGW X から PoolGW Y までの経路は，実験の前後でいずれも 19 ホップであった．また，PoolGW X から PoolGW Y に 64 bytes の ICMP Echo Request（ping）

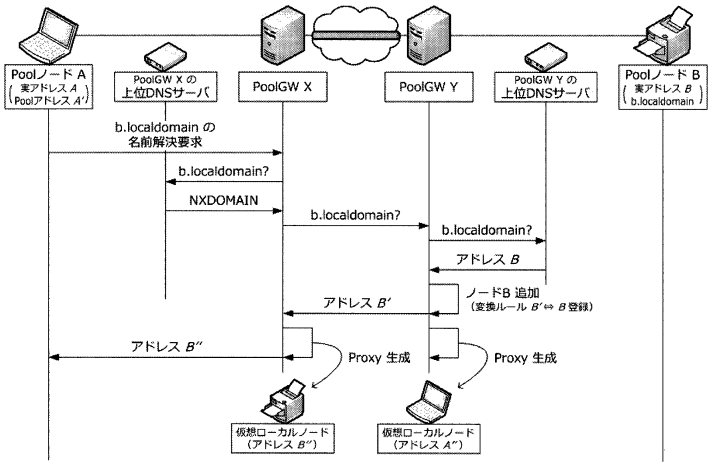

図 6　名前解決を契機としたノードの自動追加

を 30 回送信して測定した Round-Trip-Time（RTT）は，最小 31.56 ms，最大 38.08 ms，平均 33.05 ms であった（表 1）．

まず，名前解決時にノードの自動追加を伴う場合について応答時間を測定した．ノード B が Peer Pool に参加していない状態で Pool ノード A から PoolGW X に `b.localdomain` の解決要求を送る試行を 15 回繰り返したところ，応答時間は，最小 225 ms，最大 413 ms，平均 278 ms であった（表 2）．次に，ノード B が自動追加された後 PoolGW X のキャッシュが切れるまで待ってから再度同様に問い合わせる場合について測定した．このときの応答時間は，最小 96 ms，最大 321 ms，平均 127 ms であった（表 2）．以上の結果から，本実装は充分に実用に耐えると考えられる．

6. 関連研究

6.1 Shepherd

Shepherd [2] は，Proxy ARP を用いた Ethernet フレーム集約モデルに基づくノード単位 VPN 技術である．各プライベートネットワークに 1 台ずつ設置されたシェパードノードと呼ばれるノード（本システムにおける PoolGW に相当）が Proxy ARP を用いてパケットを集約し，他のプライベートネットワークのシェパードノードに転送する．したがって，提供される接続性や導入までの敷居は本研究の提案システムと同等である．しかし，Shepherd にノードを追加するには，ユーザがシェパードノード上の GUI を操作する必要があり，本研究のように自動的にノードを追加することはできない．

6.2 RNSplicer

RNSplicer は，Tunneling with Service Discovery [4] という，複数ネットワーク間のサービス連携手法の実装であり，Bonjour を対象としたアプリケーションゲートウェイ型の技術である．RNSplicer は，グローバルに公開されていない名前を用いた通信を可能にしている点で本研究と類似している．一方，本研究が通常のユニキャスト DNS による名前解決扱うのに対し，RNSplicer は，Bonjour を対象としているため Multicast DNS（mDNS）を扱う．本研究は，通常のユニキャスト DNS による名前解決を扱っている．また，接続性を提供するレイヤや利用可能なプロトコルも異なる．

7. まとめ

本稿では，ノード単位 VPN の一実装である Peer Pool を拡張し，名前解決を契機としてノードに自動的に接続性を与えるシステムを提案した．従来のノード単位 VPN は手動でノードを追加する煩雑さに課題があったが，提案システムはこれを解決した．また，提案システムを実装し，実用に耐える応答時間で動作することを確認した．

一方，本稿では，セキュリティについて考慮しなかった．無条件にノードに接続性を与えるとセキュリティ上の問題となるため，今後は，運用ポリシーに応じて自動追加されるノードの範囲に制限をかけられるようにするなどの検討が必要である．

謝　辞

本研究の一部は，総務省「戦略的情報通信研究開発推進制度（SCOPE）」の支援を受けて行われた．また，本研究に協力してくれた後輩の松井智紀君に感謝を表する．

文　献

[1] 中野悦史, 西尾信彦：“Peer Pool: DNS クエリによって構成されるオーバーレイネットワーク”, インターネットコンファレンス 2007 (IC2007), pp. 3–12 (2007).
[2] 青柳禎矩, 高橋ひとみ, 斉藤匡人, 間博人, 徳田英幸：

"Shepherd: 利用者主導型仮想プライベート・サブネットワーク構築機構", 電子情報通信学会技術研究報告. NS, ネットワークシステム, **106**, 577, pp. 227–232 (2007).
[3] "OpenVPN", `http://openvpn.net/`.
[4] Mizukami, T. and Cho, K: "Tunneling with service discovery: a remote access model and implementation", Consumer Communications and Networking Conference, 2006 3rd IEEE, **1**, pp. 223–227 (2006).