安全な多重帰属制御を実現するVPN分散管理プロトコルの提案

中村嘉隆[†] 木谷友哉[††] 木村旭[†]
山口弘純[†] 中田明夫[†] 東野輝夫[†]

Virtual Private Network（VPN）はサイトと呼ばれるネットワーク上の地点間にセキュアな接続を提供する．VPN を構成するサイト数の増加にともない，複数の VPN に同時に帰属したいというサイトも増加することが予想される．このような VPN の多重帰属を認めた場合，多重帰属をしているサイトを通じた情報流出の危険性が考えられる．そこで，各 VPN が定めるポリシと呼ばれるアクセス条件に各サイトが違反しないかを検証できることが望ましい．しかし，既存の VPN アーキテクチャでは，このような多重帰属制御は想定されていない．本論文では，既存の VPN プロトコルおよびアーキテクチャ上で，ポリシを用い，多重帰属を制御して VPN をセキュアに保つような VPN 管理手法の提案を行う．

Secure Multiple Association Control Protocol for VPNs

Yoshitaka Nakamura,[†] Tomoya Kitani,[††] Akira Kimura,[†]
Hirozumi Yamaguchi,[†] Akio Nakata[†] and Teruo Higashino[†]

Virtual Private Networks (VPNs) have been widely employed to establish secure connections among network sites. Due to the increase in the number of sites constructing VPNs, some sites may wish to associate with two or more VPNs simultaneously. However, there is danger of causing the information leakage in multiple associations of VPNs. Thus, it is desirable to verify that such multiple association does not violate access regulation called "policy" which is defined in each VPN. However, in the existing VPN architecture, it is not assumed to control such multiple association. In this paper, we propose a policy-based protocol that controls multiple association and keeps VPNs secure over existing VPN protocols and architectures.

1. はじめに

企業などにおいて，地理的に離れた部署間で安全に通信を行ったり，特定のビジネスパートナに対して安全に情報を提供したりする手段として，専用線の代わりに公衆回線を利用した VPN（Virtual Private Network）が普及してきている．公衆網を使う VPN では専用線を使った通信と比較してコストが小さく，公衆網に仮想的なリンクを張るために 1 対多の接続も容易に実現できる．そのようなメリットに加え，現在のサービスの多様化から，サイト（VPN を構成する最小単位）が複数の VPN に同時に帰属（多重帰属）したいという要求が起こっている．

通常，各 VPN には使用する暗号プロトコルや帰属を許すサイトの制限といった帰属受理に関する条件（ポリシ）がそれぞれ設定されており，VPN はそのポリシを充足するサイトの集合によって構築されている．あるサイトが複数の VPN に多重帰属する場合は，それらの VPN が持つポリシをすべて満たす必要がある．従来の VPN 構築プロトコルでも，帰属要求の受理に関する設定をあらかじめ静的に行っておくことや，帰属要求に対して手動で対応することで多重帰属の制御を実現することは可能である．しかし，前者はすべての可能性をあらかじめ設定しておく必要があり，動的な帰属要求に対応することは難しい．後者は応答速度や規模の面で非現実的である．また，文献 1)〜3) では，帰属要求に対するポリシの判定を自動で効率的に行うポリシベースの制御法について提案されているが多重帰属制御については考慮されていない．

従来型の多重帰属制御では，帰属要求を出すサイト

† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
†† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

が所属している VPN と帰属要求を受けた VPN のポリシのみをもとに要求受理の可否を判定する．しかし，サイトが複数の VPN に同時に帰属すると，そのサイトがゲートウェイと化することによって，異なる VPN 間が間接的に通信可能となるため，離れた VPN 間で情報漏洩が起こる可能性がある．営利面などで競合関係にある企業どうしでは間接的にでも VPN を接続したくないと考えられるため，このような間接的な接続関係を要求の受理の際に考慮する必要がある．そのため，要求の送信元・送信先の 2 つの VPN のポリシのみをもとにした帰属制御ではなく，各 VPN が別の VPN を介して通信可能となっている他の VPN の情報も考慮に入れ，帰属の可否を決定する．

そこで本論文では，既存の VPN アーキテクチャ上で，各 VPN の接続に関する情報と，各 VPN の規定しているポリシを効率性を考慮しながら収集し，それらの情報に応じて帰属の可否を判定することで，動的に多重帰属の制御を行う VPN 帰属制御プロトコルを提案する．提案手法では，VPN のアーキテクチャはサービスプロバイダが提供するネットワークにおいて，サイト側に用意されるカスタマエッジルータ（以後 CE と呼ぶ）と，プロバイダネットワークに接続するプロバイダエッジルータ（以後 PE と呼ぶ）から構成され，VPN の通信自体は既存のプロトコルを用いるものとする．ポリシとして従来の VPN 帰属ポリシと別に，間接的に通信可能な状態にある VPN に関するポリシを定義する．VPN の規模が大きくなった場合，接続関係にあるすべての VPN のポリシを効率的に判定しなければ，収集時間や帯域，制御情報を保持するための記憶領域が非常に大きくなり，スケーラビリティを達成できない．そこで，全 VPN から集めるポリシを VPN の接続関係のみに関するものに限定し，各 VPN が持つ VPN どうしの接続状態を PE に保持させ，間接的に接続している VPN の集合ごとに該当する VPN の情報を持つ PE のみからなる情報収集用のオーバレイネットワーク（以後 PE-グラフと呼ぶ）を構築する．これを用いて各 PE の分散処理で情報を収集することで，制御にスケーラビリティを持たせる．また，情報の一貫性を保つために，帰属要求の競合が発生するような場合も解決できるような情報収集法を提案する．

2. 多重帰属

本論文では VPN V にサイト S が含まれている場合，サイト S は VPN V に**帰属する**という．

VPN 間で情報交換を行うためには，1 つのサイトが複数の VPN へ帰属すること，すなわち，多重帰属が必要である．そこで，ある VPN に帰属しているサイトがその VPN とは異なる VPN にも帰属している，すなわち，あるサイトが複数の VPN に帰属しているとき，VPN に**多重帰属している**と定義する．

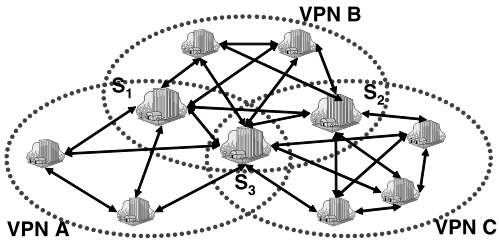

図 1 VPN の多重帰属
Fig. 1 Multiple association.

図 1 に多重帰属の例を示す．図 1 において，サイト S_1 は VPN A と VPN B に，サイト S_2 は VPN B と VPN C に，サイト S_3 は VPN A，VPN B，および VPN C に多重帰属しているという．

2.1 従来の VPN での多重帰属制御

一般的な VPN では，VPN の管理者の間で通信プロトコルの選定やアクセス制限などの事前調整を行い，ルータの設定を変更することで多重帰属を実現することができる[4]．しかし，これらのアプローチは手動による静的な設定を用いて実現されており，動的な帰属要求を想定した場合，すべての可能性をあらかじめ設定しておかなくてはならず，帰属制御が非効率的になる．

一方，MAVPN アーキテクチャ[5] を用いた場合，サイト内で VLAN などによるグルーピングを行い，各ホストが接続する VPN ごとに認証を受けることで VPN への多重帰属が可能となる．この方式では，グループごとにアクセス制御を行うことで，従来の VPN に比べ効率の良い多重帰属を実現できる．また，帰属制御を動的に行う VPN アーキテクチャ[1),3),6)] では，サイトの状況に応じてルーティングテーブルを構成する，というようなポリシを設定できるので，ポリシの設定次第で多重帰属の動的な制御も可能である．

2.2 多重帰属の問題点

前述した既存の VPN アーキテクチャでは，多重帰属自体の実現は可能であるが，次のような帰属の制御は想定されていない．

たとえば**図 2** のように，A，B，C の 3 つの組織があり，それぞれが独自の VPN を構成しているとする．A と B，B と C はそれぞれ友好関係にあるが，A と C は競合関係にあるとする．今，A が構成する VPN と B が構成する VPN が通信可能な状態にある（A と

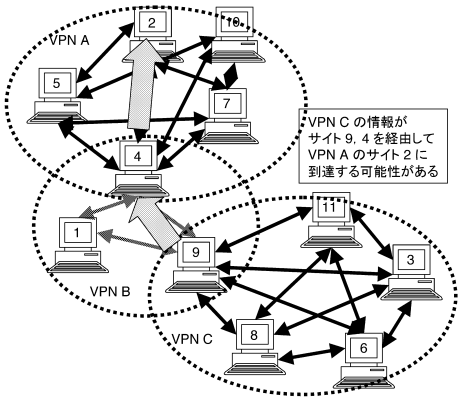

図 2 多重帰属サイトを介した情報の漏洩
Fig. 2 Information leakage.

B の両 VPN に多重帰属しているサイトが存在する）とする．このとき，C のサイトが B に対して帰属要求を発生させた場合，A は C への情報流出の可能性がある以上，友好関係にある B の帰属も認めたくないという立場をとることも考えられる．つまり，ここで A のポリシとして，B が自身（A）と通信可能な状態にあるときには B において C のサイトの帰属は認めないが，それ以外の状態では認める，といったものが考えられる．

また，このような形の情報流出は，中間 VPN のホストに悪意があって中継されてしまう場合だけではなく，スパイウェアなどによって，同時に帰属しているサイトに無意識のうちに中継される可能性があるため，十分な脅威となる．

既存の VPN においてこのような帰属制御を実現するには，VPN 管理者がつねにネットワーク状況を監視し，不都合な状況が発生した場合には，ポリシを変更する，といったことが必要となる．また，文献 1), 3) の手法では，ポリシの設定次第で自身の状態に応じた帰属制御が可能であるが，他のサイトの状態も考慮に入れたポリシの設定は想定されていない．したがって，上記のような帰属制御の実現は困難である．

3. 安全な多重帰属の実現

情報漏洩を完全に防ぐためには，VPN を介して接続している全ネットワークを調べ，情報を受け取られたくないサイトが存在しないことを確認する必要がある．しかし，帰属の確立時に，間接的に接続関係にあるすべてのサイトをたどって，そのようなサイトを発見するのは，スケーラビリティの点からも現実的ではない．そこで，効率良く VPN の接続関係に関するトポロジ情報を収集できる手法を考える必要がある．

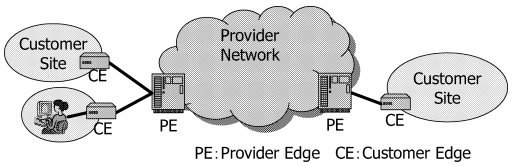

図 3 アーキテクチャの概観
Fig. 3 Overview of architecture.

3.1 対象アーキテクチャ

本論文は，**図 3** のような，プロバイダの用意する PE とサイト側で用意する CE で構成される一般的な PPVPN アーキテクチャ上での制御を対象とする．CE はサイトの VPN への帰属要求や離脱要求を PE に送信したり，自身の規定するポリシを管理したりするための機能を持つ．PE はサイトの要求に基づいて通信設定を変更したり，各 VPN が規定するポリシを管理したりするための機能と，PE 間でポリシなどの情報を交換する機能を持つ．また本プロトコルのために，PE には所属している CE の VPN 情報や，PE 間のルーティングテーブルを保持するためのメモリ領域が確保されており，メッセージ交換によってこれらの情報を更新する機能が拡張されているとする．

3.2 諸 定 義

多重帰属関係を通した情報漏洩を防止するトポロジ制御問題（以下，「安全な多重帰属制御問題」と呼ぶ），および，そのために必要な諸概念を定義する．

3.2.1 サイトの論理ネットワークトポロジの定義

まず，**サイト** s の集合を S とし，**VPN** v はサイトの任意の部分集合，すなわち $v \in 2^S$ であると定義する．VPN 全体の集合を V とすると，$V = 2^S$ である．任意の 2 つの VPN v_1，v_2 に対して，それらが**多重帰属関係にある**とは，あるサイト $s' \in S$ が存在して，$s' \in v_1 \cap v_2$ が成り立つことであると定義し，m-assoc(v_1, v_2) と書く．**多重帰属グラフ** $G_{ma} = (V, E_{ma})$ は，各頂点を VPN とし，多重帰属関係にある VPN どうしを枝で結んだ無向グラフ，すなわち，枝の集合が $E_{ma} \stackrel{\text{def}}{=} \{(v_1, v_2) | v_1, v_2 \in V, \text{m-assoc}(v_1, v_2)\}$ と定義されるグラフとする．VPN v_1，$v_2 \in V$ が**到達可能であるとは**，多重帰属グラフ $G_{ma} = (V, E_{ma})$ において v_1 から v_2 へのパスが存在することであると定義し，reachable(v_1, v_2) と書く．多重帰属グラフ $G_{ma} = (V, E_{ma})$ の連結成分の頂点集合 $V_1, \ldots, V_k$ をそれぞれ**到達可能 VPN 群**と呼ぶ．連結成分の定義より任意の i，$j \in \{1, \ldots, k\}$ に対して $i \neq j$ ならば $V_i \cap V_j = \emptyset$ である．また，容易に示されるように，任意のサイト $s \in S$ に対して，$s \in v$ かつ $v \in V_i$ であるような到達可能 VPN 群はちょうど 1 つ存在す

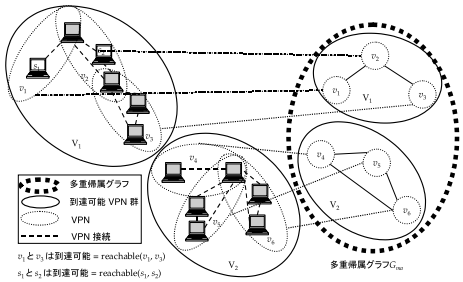

図 4 多重帰属グラフと到達可能性
Fig. 4 Multiple association graph and its relation to reachability.

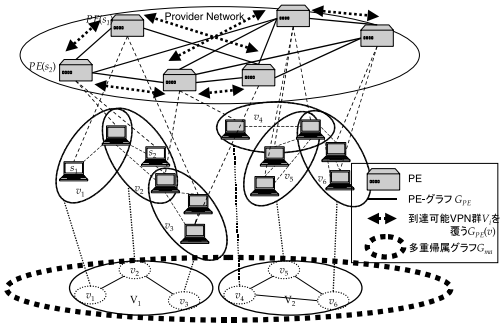

図 5 PE-グラフ
Fig. 5 PE-Graph.

る．サイト s_1，$s_2 \in S$ が**到達可能である**とは，ある VPN v_1，$v_2 \in V$ が存在して，$s_1 \in v_1$，$s_2 \in v_2$，かつ reachable(v_1, v_2) が成り立つことであると定義し，reachable(s_1, s_2) と書く．これらの例を**図 4** に示す．

3.2.2 サイトの多重帰属ポリシの定義

サイト $s \in S$ の**多重帰属ポリシ** $Policy(s)$ とは，到達可能であってはならないサイトの集合であると定義する☆．以後，$Policy(s)$ のことを s の**禁止リスト**，$s' \in Policy(s)$ を s の**禁止サイト**と呼ぶ．サイト s の多重帰属ポリシが**満たされている**とは，reachable(s, s') であるようなサイト $s' \in Policy(s)$ が存在しないことであると定義する．VPN v のポリシが**満たされている**とは，v に属する任意のサイト $s \in v$ のポリシが満たされていることであると定義する．VPN 群 V_i のポリシが**満たされている**とは，V_i に属する任意の VPN $v \in V_i$ のポリシが満たされていることであると定義する．

3.2.3 サイトの物理ネットワークトポロジの定義

サイト $s \in S$ に対して，s に物理的に最も近い s を管理する **PE** が 1 つずつ存在するとし，$PE(s)$ と書く．すべての PE の集合をそれぞれ $PEs \stackrel{\text{def}}{=} \{PE(s) | s \in S\}$ と書く．s と $PE(s)$ は CE を介して通信路で結ばれているとする．各 PE はいくつかの他の PE と直接通信路で結ばれているとする．$PE(s)$ と直接通信路で結ばれている他の PE の集合を Neighbor$(PE(s))$ で表す．任意の PE $p \in PEs$ を頂点集合とし，枝の集合が $E_{PE} \stackrel{\text{def}}{=} \{(p, p') | p \in PEs \wedge p' \in \text{Neighbor}(p)\}$ と定義される無向グラフ $G_{PE} = (V_{PE} = PEs, E_{PE})$ を **PE-グラフ**と呼ぶ．各 PE $p \in PEs$ に対して，それが管理するサイト s' の集合を $Sites(p)$，p が管理するサイトが属する VPN の集合を $VPN(p)$ として，$Sites(p) \stackrel{\text{def}}{=} \{s | s \in S \wedge p = PE(s)\}$，$VPN(p) \stackrel{\text{def}}{=} \{v | v \in V \wedge s \in Sites(p) \wedge s \in v\}$ と定義する．

PE-グラフ $G_{PE} = (V_{PE}, E_{PE})$ が，到達可能 VPN 群 $V_i \subseteq V$ を**覆う**とは，V_i に属する任意の 2 つの VPN v_1，v_2 に対して，v_1 に属する任意のサイト s_1 を管理する PE $PE(s_1)$ から v_2 に属する任意のサイト s_2 を管理する PE $PE(s_2)$ へのパスが G_{PE} 上に存在することであると定義する．多重帰属グラフ G_{ma} の任意の到達可能 VPN 群 V_i を覆う PE-グラフを，G_{ma} **を管理可能な** PE-グラフであると呼ぶ（**図 5**）．G_{PE} を，G_{ma} を管理可能な任意の PE-グラフとしたとき，任意の $p \in PEs$ および p が管理する任意の VPN $v \in VPN(p)$ に対して，$v \in V_i$ であるような到達可能 VPN 群 V_i 全体を覆う G_{PE} の部分グラフを $G_{PE}(v) = (V_i, E_{PE}(v))$ と書き，$G_{PE}(v)$ において，p と隣接関係にある PE，すなわち，$(p, p') \in E_{PE}(v)$ であるような p' の集合を Neighbor(p, v) と書く．

s の任意の VPN への帰属要求は CE を介して $PE(s)$ に対して送信され，それに対する許可処理は，$PE(s)$ が PE-グラフで示される通信トポロジに従って他の PE と通信を行った後に行うものとする．

3.2.4 問 題 定 義

安全な多重帰属制御問題とは，任意のサイト $s \in S$ が任意のタイミングで $s \notin v$ であるような任意の VPN $v \in V$ に対して帰属要求を出したとき，VPN v にサイト s を追加しても $v \in V_i$ であるような，唯一存在する到達可能 VPN 群 V_i のポリシが満たされるときのみ，s の v への帰属を許可する問題であると定義する☆☆．また，帰属を許可する過程で，複数の帰属要求

☆ 一般には，他のポリシも考えられるが，簡単のため本論文では，このような狭義のポリシのみを扱う．

☆☆ ここでは，サイトの VPN への帰属要求がポリシを満たす場合に必ず帰属が許可されること（活性）は保証しなくてもよいとしている（別のサイトの帰属要求と競合すれば却下される可能性があるため）．しかし，十分な時間継続的に帰属要求を繰り返せば，いつかは十分高い確率で帰属が許可されることは，現実的には保証できると考えられる．本論文では紙面の制限により，本文で述べたような安全性の保証のみに焦点を絞る．

が同時並行的に発生した場合にも，要求の競合を正しく解決しながら情報の収集を行えているものとする．

4. 提案手法

4.1 基本方針

安全な多重帰属を実現するためには，ポリシをサーバなどで集中管理し，それに基づいて帰属制御を行う手法が考えられる．しかし，サーバを用いた場合，VPN の構成変更が頻繁に行われる状況ではスケールしない．このような状況は，たとえば VPN の構成単位が個人のような小さい単位であり，帰属先を頻繁に変更するようなときに起こりうる．一方，各 PE において分散で判定を行う場合，PE 上に収集したポリシをキャッシュしておき，それをもとに判定するという手法はポリシの収集を最も効率良くできると考えられるが，キャッシュを用いた場合，情報の一貫性を保証できないため，帰属制御に必要な安全性の面から適当ではない．そこで，提案手法では，各 PE 上でポリシ判定を行いながら，効率性を考慮しつつポリシの収集を行う．

多重帰属ポリシはネットワークにおけるサイトの存在情報をもとに判定するため，ネットワーク全体のサイト情報を収集する必要がある．すべてのサイトをたどってこの情報を収集することは非効率的であり，また，すべてのサイトをたどったという保証ができないため，該当サイトが存在しないことを必ず保証することはできない．そこで，ある到達可能 VPN 群を覆う PE-グラフ上で各サイト情報を持っている PE をたどることで，各 PE 上で分散的にポリシ判定を行い，その結果を収集することを考える．また，同時並行的に帰属要求が発生するような状況においても，これらの情報に一貫性を保つことができるよう，収集時に要求の競合を解決させることが必要である．

4.2 制御の流れ

図 6 を用いて制御の流れを説明する．サイト s が $PE(s)$ に VPN v への帰属要求を出したとき（処理 (1)），まず，$PE(s)$ は $v \in V_i$ であるような到達可能 VPN 群 V_i に属するサイトで，$PE(s)$ 自身が直接ポリシ管理をしている任意のサイト s' が s のポリシに反しないか，また，逆に s が s' のポリシに反しないかローカルに調べる（処理 (2)）．もし反していたら，s に対して「no」を返答する．反していないならば，$PE(s)$ は他の PE を介して $v \in V_i$ であるような任意の到達可能 VPN 群に属する全サイトに要求を転送し（処理 (3)），それに対する返答を集積する（処理 (4)）．転送先は $v \in V_i$ であるような V_i に属するサイト全体

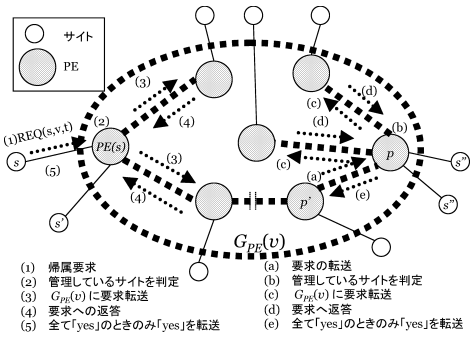

図 6 制御の流れ
Fig. 6 Overview of proposed protocol.

に情報を転送するのに十分な物理ネットワークトポロジ $G_{PE}(v)$ を用いればよい．すなわち，隣接 PE 集合 $\mathrm{Neighbor}(PE(s), v)$ に属する PE にのみ要求を転送し，$\mathrm{Neighbor}(PE(s), v)$ に属するすべての PE から「yes」を受信したときにのみ，s に対して「yes」を返答する（処理 (5)）．

PE p において，$p' \in \mathrm{Neighbor}(p, v)$ であるような他の PE p' から，s の v への帰属要求が転送されてきた（処理 (a)）場合，上と同様に s および任意の $s' \in \mathit{Sites}(p)$ のポリシに反しないか否かをローカルに判定し（処理 (b)），もし反していれば p' に「no」を返答する．反していなければ $\mathrm{Neighbor}(p, v) \setminus \{p'\}$ に属する PE に要求を転送し（処理 (c)），返答を集積する（処理 (d)）．この場合も，$\mathrm{Neighbor}(PE(s), v) \setminus \{p'\}$ に属するすべての PE から「yes」を受信したときのみ，p' に対して「yes」を返答する（処理 (e)）．

この方針により，s の v への帰属要求は $G_{PE}(v)$ のトポロジに沿って v が属する到達可能 VPN 群 V_i に属する全サイトに転送され，それに対する返答がすべて「yes」の場合にのみ，最初の帰属要求に対して「yes」が返答されることが保証される．

4.3 帰属先 VPN と代表 PE の対応の管理

帰属要求を最初に受信した PE（要求元 PE）は帰属要求の送信先の PE を知らない場合がある．たとえば，PE p が VPN v への帰属要求を初めて受信したときは，$\mathrm{Neighbor}\ (p, v)$ が空集合であるため，v への帰属サイトを管理する適当な代表 PE p' を選択して v への帰属要求を転送し，PE グラフにおいて p' と連結する．この処理のためには，v への帰属サイトを管理する既存 PE の有無の判定およびそれらのうち代表 PE の検索を可能にしておく必要がある．そこで，VPN v および v に帰属する代表 PE の情報を持つ検索サーバの存在を仮定する．この検索サーバには，各 VPN についてその VPN に最初に帰属した PE が代

表 PE となり，自分自身を登録しておく．これによって，検索サーバに VPN の ID と PE のアドレスの組が登録され，他の PE はこの情報を参照することができる．要求元 PE が要求の宛先を知らない場合はこの検索サーバから宛先を知り，要求を送信する．また，通常代表 PE はその VPN に最初に帰属した PE とするが，なんらかの理由でその PE が該当 VPN から離脱する場合は，同一 VPN の隣接 PE と交代し，検索サーバに通知する．

4.4 競合の解決

PE p がサイト s の VPN v への帰属要求を隣接 PE に転送して返答を待っている間に，他の隣接 PE p'' から同じ帰属要求あるいは別のサイト s' の VPN v' への帰属要求を受信する可能性がある．同じ要求が自分宛に転送されるのは PE-グラフに閉路がある場合なので，単純に無視すればよいが，異なる要求に対しては適切に処理しなければ，競合する 2 つの帰属要求を同時に許可してしまう可能性がある．これを回避する手法を以下に述べる．

4.4.1 基本アイディア

任意の帰属要求メッセージにはそれが送信された時刻に関する論理時刻印（Lamport のタイムスタンプ[7]）を記録するものとする．論理時刻印とは，分散システム全体で起こるイベントに対して，値の大小関係がイベントの因果関係の順序に矛盾せず，かつ，システム全体で一意な整数値を付与したものである．もし PE p_1 が，論理時刻印が t_1 である，あるサイト s_1 の v_1 への帰属要求（$REQ(s_1, v_1, t_1)$ とする）をすでに転送し終わってから，論理時刻印が t_2 である別のサイト s_2 の v_2 への帰属要求（$REQ(s_2, v_2, t_2)$ とする）を隣接 PE $p'' \in \mathrm{Neighbor}(p_1, v_2)$ から受信したとき，多重帰属グラフ G_{ma} の連結成分の性質により，帰属要求に含まれる VPN v_1 と v_2 はともに同じ到達可能 VPN 群に属している場合がある．この場合は関係するサイトのポリシによっては競合する可能性がある．

先着の $REQ(s_1, v_1, t_1)$ を受け入れたと仮定した場合，すなわち v_1 を $v_1 \cup \{s_1\}$ に更新したと仮定した場合，$REQ(s_2, v_2, t_2)$ が自身の管理するサイト群 $Sites(p_1)$ のポリシに反しないかローカルに判定する．もし反しないならば，$REQ(s_2, v_2, t_2)$ は，PE p_1 が知る限り以前の $REQ(s_1, v_1, t_1)$ と競合しないと判定されて以前の要求と同時並行処理され，もし反すると判定されたならば，この $REQ(s_2, v_2, t_2)$ は，$REQ(s_1, v_1, t_1)$ と競合すると判定される．この場合，次に論理時刻印を比較し，時刻印が早いほうの帰属要求のみ受け入れる．すなわち，$t_1 < t_2$ ならば $REQ(s_2, v_2, t_2)$ を受信した p'' に対して即座に「no」を返答する．$t_2 > t_1$ ならば（等しい論理時刻印は存在しないため $t_1 = t_2$ にはならない[7]）先着の $REQ(s_1, v_1, t_1)$ を受け入れなかったと仮定した場合に，$REQ(s_2, v_2, t_2)$ がポリシに反しないかローカルに判定する．もし反しないならば，以前の帰属要求の代わりにこの帰属要求を処理する．この場合，もし以前の帰属要求 $REQ(s_1, v_1, t_1)$ に対する返答がまだであれば，こちらには「no」を回答する．すでに回答済みの場合は何もしない．この間にさらに別の帰属要求があれば，論理時刻印を比較して同様の処理を行う．この方針により，異なるサイトからほぼ同時に生じた競合する帰属要求に対しては，それらの論理時刻印において最も早い帰属要求のみが，そのサイトから到達可能な全サイトを管理する PE から「yes」の返答を受信することができ，それ以外の帰属要求に対しては，必ずある PE から「no」の返答が受信される．

たとえば，$REQ(s_1, v_1, t_1)$ と $REQ(s_2, v_2, t_2)$ をそれぞれ最初に受信した PE をそれぞれ p_1，p_2 とすると，$p_1 \neq p_2$ ならば p_1 では $REQ(s_1, v_1, t_1)$ が先に受信され，p_2 では $REQ(s_2, v_2, t_2)$ が先に受信されているはずである．ここで $t_1 < t_2$ と仮定すると，p_1 が必ず $REQ(s_1, v_1, t_1)$ に対して「no」を返答し，$t_1 > t_2$ ならば，p_2 が必ず $REQ(s_2, v_2, t_2)$ に対して「no」を返答することが保証される．また，$p_1 = p_2$ ならば p_1 の受信順で早いほうに「yes」，遅いほうには「no」が返答されることが同様に保証される．このことによって競合が回避される．

4.4.2 未処理要求が複数ある場合への拡張

PE に未処理の帰属要求が複数ある状態では，1 つ 1 つの要求は競合していなくても，未処理の要求がいくつか処理された時点で競合が発生する可能性がある．たとえば，VPN が現在 $v_1 = \{s_1\}$，$v_2 = \{s_2\}$，$v_3 = \{s_3\}$ という状況であるとし，今，PE p は $REQ(s_1, v_2, t_1)$ および $REQ(s_2, v_3, t_2)$ を受信して他の PE にそれを転送して返答を待っている状態であるとする．このとき，p がさらに $REQ(s_3, v_1, t_3)$ を受信したとする．ここで s_3 のポリシとして，s_1 と到達可能であってはならないとすると，この帰属要求は前の 2 つの帰属要求 1 つ 1 つとは競合しないが，両方が許可されたと仮定すると競合する（v_1 から v_3 へ到達可能となるため）．この場合は，受信した要求の時刻印が未処理の要求のいずれの時刻印よりも早かった場合のみ，未処理の要求すべてに「no」を回答し，受信した要求の処理を開始する．未処理の要求のうち 1 つでも時刻印が

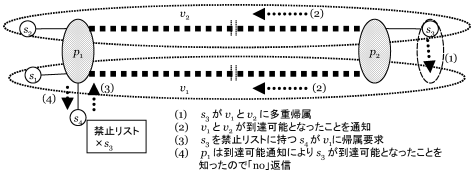

図 7 到達可能通知
Fig. 7 Notification of reachability.

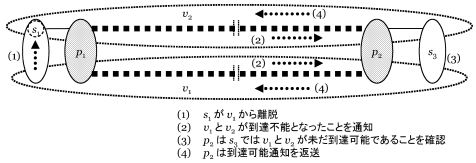

図 8 到達不能通知の訂正
Fig. 8 Correction of unreachability notification.

早いものがあった場合は，受信した要求に「no」を回答し，未処理の既存の要求の処理を継続する．

4.5 多重帰属関係の通知

多重帰属はサイト単位で発生するため，離れた別の地点で同じ VPN への多重帰属が存在している可能性がある．たとえば，**図 7** のように，PE p_1 で $v_1 = \{s_1\}$，$v_2 = \{s_2\}$ を管理していて，p_1 は v_1 と v_2 は互いに到達可能でないと思っていたとする．p_1 から遠く離れたある PE p_2 において，サイト s_3 が v_1 と v_2 に多重帰属したとき，v_1 と v_2 は新たに到達可能となったので，今後 p_1 が新たな帰属要求を受けたとき，たとえば s_4 が v_1 に帰属したいが s_3 とは到達可能であってはならないという帰属要求があった場合（処理 (4)）などは却下する必要がある．この例のように，各 PE でローカルにポリシ判定するときや，VPN 群の統合を行うときに，このような別の多重帰属の存在が問題となる．そこで，多重帰属が発生した場合は到達可能になったことを，多重帰属が終了した場合は到達不能になったことを VPN 群内の各 PE に通知する．

ただし，ある PE で 2 つの VPN が到達不能になったと判断して通知を出しても，別の PE でそれらの VPN に多重帰属するサイトが存在していて，実はまだ到達可能であったという場合が考えられる．たとえば，**図 8** のように，PE p_1 で $v_1 = \{s_1\}$，$v_2 = \{s_1\}$ という状況から s_1 が v_1 を離脱したとする．p_1 はこのことで v_1 と v_2 は到達不能になったと判断し，そのことを他の PE に通知する．しかし，p_1 から遠く離れた PE p_2 では $v_1 = \{s_3\}$，$v_2 = \{s_3\}$ という状況であり，v_1 と v_2 はまだ到達可能であることを知っている．この場合，p_2 は隣接 PE から，p_1 からの到達不能通知が転送されてきた時点で，それを取り消すため v_1 と v_2 がまだ到達可能である通知を，到達不能通知が送られてきた方向に向かって返送する．なお，これらの通知を受信した PE は，一貫性を維持するために，未処理のすべての帰属要求に「no」を返答することにより，それらを却下する．

4.6 各 PE の状態の更新

各 PE $p \in PEs$ が保持する必要がある状態変数は，受信してまだ返答していない帰属要求 $REQ(s, v, t)$，その要求の転送元 PE p'（自分が管理するサイトから直接受信したならば，$p' = p$（自分自身）とする）および，その要求の転送先隣接 PE 群とそれらからの返答の状態（未受信/「yes」を受信/「no」を受信）の組の**未処理帰属要求リスト**，各 $v \in V$ に対して，Neighbor(p, v) で表す **VPN ごとの隣接 PE 集合**，そして，各サイト s に対し，（サイト ID，帰属する VPN リスト，帰属する VPN 群 ID）で表す**配下のサイトの VPN 帰属状況**である．

各 PE の状態が更新されるのは，(1) 帰属要求を受けたとき，(2) 帰属要求に対する返答を受信したとき，(3) 帰属要求が最終的に許可されてサイトが VPN に加わったとき，そして (4) サイトが VPN から離脱したときの 4 つの時点である．このとき，各状態変数をそれぞれ以下のように更新する．

(1)，(2) の場合 $REQ(s, v, t)$ を受信し，ローカルなポリシ判定により即座に「no」を返答しなかった場合，未処理帰属要求リストにこの帰属要求のエントリを追加する．このとき，隣接 PE からの返答状態はすべて「未受信」に初期化する．隣接 PE 群に要求を転送した後，転送先隣接 PE から返答を受信するたびに，対応する返答状態を更新する．ある転送先隣接 PE から「no」を受信，あるいは，すべての転送先隣接 PE から「yes」を受信したならば，この帰属要求に対する返答を転送元 PE に送信した後，対応する帰属要求のエントリを削除する．

(3)，(4) の場合 $REQ(s, v, t)$ が最終的に許可されたとき，s を管理する $PE(s)$ は，v に s を追加し，s から到達可能な適当なサイト s' を選び，隣接 PE 集合 Neighbor$(PE(s'), v)$ に $PE(s)$ を追加する．一方，s の v からの離脱要求を $PE(s)$ が受信した場合は，$PE(s)$ は v から s を削除し，もし，$PE(s)$ が管理するサイトに v および v から到達可能なすべての VPN v' に帰属するサイトがなくなったならば，$PE(s)$ は v に関する PE-グラフ $G_{PE}(v)$ から離脱する．具体的には，$PE(s)$ は Neighbor$(PE(s), v)$ の元から適当な p' を選択し，残りの PE 群 $pp \in$ Neighbor$(PE(s), v) \setminus \{p'\}$

に対して，Neighbor(pp, v) から $PE(s)$ を削除して p' を追加し Neighbor(p', v) から $PE(s)$ を削除して，Neighbor$(PE(s), v) \setminus \{p'\}$ の元を追加する．また，v_1，v_2 が新たに到達可能になったとの通知を p が受信した場合，Neighbor(p, v_1) と Neighbor(p, v_2) の和集合をとり，それを新たな Neighbor(p, v_1) および Neighbor(p, v_2) の値とする．

4.7 PE-グラフの再構築

正確な情報の収集の観点のみからは，PE-グラフの再構築を行う必要はない．頻繁な木の再構築は，情報の誤りや，古い情報の伝達を引き起こす反面，再構築をしない場合は，必要のない PE も接続している PE-グラフ上で，すでに情報を持っていないより多くの PE を探索することになるのみである．

しかし，PE-グラフのサイズの増大は帰属要求の処理時間増大を招くので，できるだけ PE-グラフを最適なものに再構築したい．そこで，分散スナップショット[8] を用いて定期的に PE-グラフの状態を収集し，PE の総数から決定した適当な間隔に基づいて全 PE の帰属処理をロックして再構築を行うこととする．

5. 性能評価

提案手法に対して，シミュレータを実装し，性能評価を行った．

5.1 想定環境

本論文はプロバイダが用意する数百の PE のもとで，数千のサイトが数十分単位で頻繁にポリシを更新，帰属先を変更・追加するような環境を想定している．

シミュレーションでは，10〜100 個の PE を置き，全体で 100〜1,000 個程度のサイトがランダムに帰属要求を発生させている．PE 間の遅延は平均 0.05 秒となるような正規分布で与え，PE での判定時間は，管理サイト数 × 各ポリシのサイズ（禁止リストのサイズ）× 0.001 秒とする．また，帰属要求・ポリシ変更の発生間隔を短くとり，数分間に集中的に発生させることで，要求の競合が起こりやすい状況をつくり，このときにも正しく動作するかどうかを確認している．

5.2 評価項目

本論文で評価する項目は以下のとおりである．

(1) 帰属要求に対する返答までの時間
(2) PE に追加が必要なメモリ領域量

帰属要求の判定には，PE をたどってポリシを収集する時間，および，各 PE で判定する時間が必要となる．また，競合する帰属要求が同時に発生した場合には，1 つの要求の判定が終わるまで待たされる要求が存在する．そのため，これらをあわせた応答時間が妥当な時間で収まっていることが必要である．そこで，要求への応答時間を評価する．

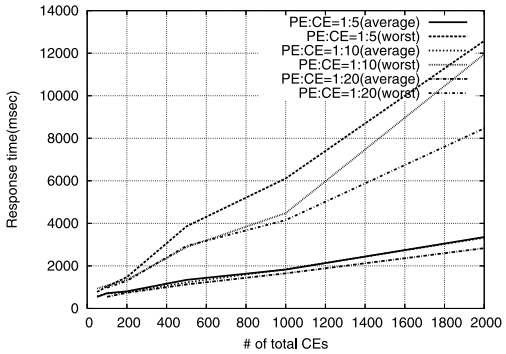

図 9 要求の応答時間
Fig. 9 Response time for requests.

また，本手法を実現するためには，既存のアーキテクチャに加え，4.6 節で述べたような独自のデータが必要となる．したがって，各 PE にこれらの情報を格納するための記憶領域を追加しなくてはならない．この記憶領域量が全体のサイト数の増加に対して妥当な値に収まり，爆発的に増加しないことを確認する．

5.3 要求の応答時間

総 CE 数が 100〜1,000 までの場合について，PE 数と CE 数の比をそれぞれ 1 : 5，1 : 10，1 : 20 とした状況下での要求の応答時間を評価した．**図 9** に結果を示す．ここでは．x 軸に総 CE 数（=総サイト数），y 軸に要求発生から回答受信までの時間をとっている．PE 数の割合によらず，総 CE 数の増加に対して平均応答時間の増加は低く抑えられており，総 CE 数 2,000 のときでも 3 秒程度で判定できている．これは，ポリシの判定のための PE 巡回時に，PE-グラフを利用することで，不必要な PE の巡回を回避しているためである．これにより，本手法は実用上十分な応答時間が実現できているといえる．

5.4 PE のメモリ領域

本手法を実現するために PE に追加することが必要なメモリ領域量を評価する．ここでは，(1) PE 数を 10 に固定し，各 PE の管理する CE 数を 10 から 100 まで変動させた場合，(2) 各 PE の管理する CE 数を 10 に固定し，PE 数を 10 から 100 まで変動させた場合，の 2 通りについて，各 PE において本プロトコルが使用するメモリ領域の量を調べた．禁止 CE として全 CE から 10% の CE をランダムに選択し，ポリシ（禁止リスト）に記述している．この結果を**図 10** に示す．x 軸に全体の CE 数，y 軸に必要なメモリ領域量をとっている．1,000 個の CE を 10 個の PE で管

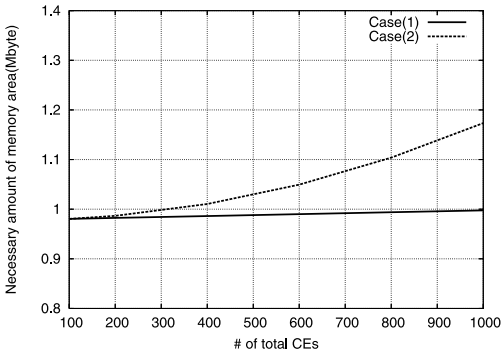

図 10 PE に必要な記憶領域量
Fig. 10 Necessary amount of memory area.

理したとき，すなわち 1 つの PE が 100 個の CE を管理しているときにも 1.2 Mbyte 程度の領域拡張ですんでいる．拡張されたメモリ領域はほとんどがポリシの格納に使われており，このポリシは管理する各 CE ごとに別々に保持するため，全 CE 数の増加よりも各 PE の管理する CE 数の増加が使用するメモリ領域量に大きな影響を与えている．実験ではポリシとして全 CE の 10%にあたる CE の ID を禁止リストに記述しているが，1PE あたりの管理 CE 数が多くなった場合は，禁止 CE 数やリストの構造を工夫することで，メモリ領域量をより削減することが可能である．したがって，本手法は既存のアーキテクチャを拡張して実現することが十分可能であると考えられる．

6. ま と め

本論文では，他サイトの帰属状況に応じて VPN の多重帰属制御を効率性を考慮しながら実現するプロトコルの提案を行った．サイトが規定するセキュリティポリシと現在の VPN 構成情報に基づいて，動的な多重帰属を分散制御で実現する．このとき，プロバイダ側で管理する PE によって現在の VPN 構成情報を管理し，多重帰属に関係する必要な PE 間のみで情報交換を行うことで，判定のための情報を正確かつ効率良く収集できる．

今後の課題としては，到達可能 VPN 群が分割された際の PE-グラフの分割法や，PE-グラフトポロジの最適化問題などがあげられる．

謝辞 本研究は一部，文部科学省科学振興調整費の援助を受けて行った．

参 考 文 献

1) Beak, S.J., Jeong, M.S. and Park, J.T.: Policy-based Hybrid Management Architecture for IP-based VPN, *Proc. 2000 IEEE/IFIP Network Operations and Management Symposium* (*NOMS 2000*), pp.987–988 (2000).
2) Kindred, D. and Sterne, D.: Dynamic VPN Communities: Implementation and Experience, *Proc. 2nd DARPA Information Survivability Conference and Exposition* (*DISCEX II*), pp.254–263 (2001).
3) Barrere, F., Benzekri, A., Grasset, F. and Laborde, R.: A Multi-domain Security Policy Distribution Architecture for Dynamic IP Based VPN Management, *Proc. 3rd International Workshop on Policies for Distributed Systems and Networks* (*POLICY 2002*), pp.5–7 (2002).
4) Hamed, H., Al-Shaer, E. and Marrero, W.: Modeling and Verification of IPSec and VPN Security Policies, *Proc. 13th IEEE International Conference on Network Protocols* (*ICNP 2005*), pp.259–278 (2005).
5) Honda, O., Ohsaki, H., Imase, M., Murayama, J. and Matsuda, K.: A Prototype Implementation of VPN Enabling User-Based Multiple Association, *Proc. 9th IASTED International Conference on Internet & Multimedia Systems & Applications* (*IMSA 2005*), pp.59–64 (2005).
6) Beigi, M., Calo, S. and Verma, D.: Policy Transformation Techniques in Policy-based Systems Management, *Proc. 5th IEEE International Workshop on Policies for Distributed Systems and Networks* (*POLICY 2004*), pp.13–22 (2004).
7) Lamport, L.: Time, Clocks, and the Ordering of Events in Distributed Systems, *Comm.ACM*, Vol.21, No.7, pp.558–565 (1978).
8) Chandy, K. and Lamport, L.: Distributed Snapshots: Determining Global States of Distributed Systems, *ACM Trans. Computer Systems*, Vol.3, No.1, pp.63–75 (1985).

(平成 18 年 5 月 21 日受付)
(平成 18 年 11 月 2 日採録)

中村 嘉隆（学生会員）
平成 14 年大阪大学基礎工学部情報科学科卒業．平成 16 年同大学大学院情報科学研究科博士前期課程修了．現在，同大学院情報科学研究科博士後期課程在学中．分散協調アプリケーションの実現に関する研究に従事．平成 18 年より日本学術振興会特別研究員．

木谷 友哉（正会員）

平成 14 年大阪大学基礎工学部情報科学科卒業．平成 18 年同大学大学院情報科学研究科博士後期課程修了．博士（情報科学）．平成 17 年より奈良先端科学技術大学院大学情報科学研究科助手．リアルタイム組み込みシステムの設計手法，組合せ最適化問題に対する近似アルゴリズム等の研究に従事．電子情報通信学会会員．IEEE Member.

木村 旭

平成 16 年大阪大学基礎工学部情報科学科卒業．平成 18 年同大学大学院情報科学研究科博士前期課程修了．現在，シャープ株式会社勤務．在学中，VPN の帰属制御に関する研究に従事.

山口 弘純（正会員）

平成 6 年大阪大学基礎工学部情報工学科卒業．平成 10 年同大学大学院基礎工学研究科博士後期課程修了．同年オタワ大学客員研究員．平成 11 年大阪大学大学院基礎工学研究科助手．平成 14 年より同大学院情報科学研究科助手．博士（工学）．分散システムや通信プロトコルの設計および実装に関する研究に従事.

中田 明夫（正会員）

平成 4 年大阪大学基礎工学部情報工学科卒業．平成 9 年同大学大学院基礎工学研究科博士後期課程修了．現在，同大学院情報科学研究科助教授．博士（工学）．実時間システムや分散システムの仕様記述と検証法，プロセス代数，時相論理等の研究に従事.

東野 輝夫（正会員）

昭和 54 年大阪大学基礎工学部情報工学科卒業．昭和 59 年同大学大学院基礎工学研究科博士後期課程修了．同年同大学助手．平成 2 年，6 年モントリオール大学客員研究員．現在，大阪大学大学院情報科学研究科教授．博士（工学）．分散システム，通信プロトコル等の研究に従事．電子情報通信学会，ACM 各会員．IEEE Senior Member.