

エージェントによるアプリケーション間コネクションの動的構成手法

今野 将[†] 北形 元[†] 原 英樹^{††}
菅沼 拓夫[†] 菅原 研次^{††} 木下 哲男^{†††}

エクストラネットのような柔構造ネットワークにおいて、その利用者やアプリケーションからの様々な要求に応じて、運用中のネットワークの接続形態や利用可能な帯域を調整することは、ネットワーク運用者にとって大きな負担となる。本論文では、この問題に対して、アプリケーションから与えられる要求に基づいて、アプリケーション相互間のコネクションを動的に構成する手法を提案する。そして、エージェント技術を利用して本手法に基づくミドルウェア、および、これを組み込んだ実験システムを構成し、これらを用いた実験を通して、本手法の有効性を示す。

Agent-based Dynamical Construction of Inter-application Connections

SUSUMU KONNO,[†] GEN KITAGATA,[†] HIDEKI HARA,^{††}
TAKUO SUGANUMA,[†] KENJI SUGAWARA^{††} and TETSUO KINOSHITA^{†††}

In a pliable network such an extra-network, it is difficult for the network operators to construct and tune the structure and functions of the network based on various requests from users and applications. In this paper, we propose a new method to construct the connections between applications dynamically with respect to the requests given by the applications. Then, we design and implement a middleware and an experimentation system by using agent technologies, and demonstrate the effectiveness of the proposed method.

1. はじめに

近年、エクストラネットをはじめとする、インターネットやテレコムネットワークなどの広域網を利用した分散コンピューティング環境が多くの企業や自治体などで利用されている^{1)~4)}。このうち、エクストラネットは、いくつかの組織を横断するプロジェクトの発生・終了とともに形成・解消される柔構造を持ち、その利用形態も様々である。そのため、プロジェクトごとに処理するタスクの内容が異なり、エクストラネットを実現するネットワークの構成や機能はプロジェクトごとに異なる。

すなわち、インターネットでは汎用のネットワーク

機能を効率良く実現することが課題であることに對して、インターネット上に構築される論理ネットワークであるエクストラネットは、そのネットワーク機能を業務ごとにカスタマイズして実現することが要求される。

しかしながら、業務ごとにエクストラネットの機能・性能を実現することは、エクストラネットの設計者の負担になる。さらに、プロジェクトの進行の過程でネットワーク機能に対する利用者要求の変更や運用規則の変更が頻繁に行われる場合には、そのエクストラネットを運用する技術者の負担が増大する。つまり、利用者要求や運用規則の変化に柔軟に対応できるネットワークが必要とされている^{5),6)}。

そこで本論文では、このようなエクストラネットを構築・運用するうえで、利用者要求や運用規則の変化に対応したネットワークを実現するという観点から、エクストラネット上で定義される分散アプリケーション間の論理コネクション(アプリケーション間コネクション)とその帯域に着目し、以下の課題を解決することを目的とする。

[†] 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University

^{††} 千葉工業大学情報ネットワーク学科
Department of Information and Network Science, Chiba Institute of Technology

^{†††} 東北大学情報シナジーセンター
Information Synergy Center, Tohoku University

(1) 分散アプリケーション間に提供される論理コネクションが必要とする帯域を提供するためには、ネットワークに関する様々な設定を行うことが必要である。この作業が利用者あるいはエクストラネットの運用者の負担になる。

(2) 多数の分散アプリケーションが有限のネットワークの帯域を共有して、並行に動作するとき、これらの中でネットワークの帯域の割当てに関する競合が発生する。このような競合を解消し、多数の利用者にサービスを提供するためには、ネットワークの設定を変更することが必要になる。この作業が利用者あるいはエクストラネットの運用者の負担になる。

本論文では、上記の課題を解決するために、TCP/IP 層とアプリケーション層の間にエージェント型ミドルウェアを導入し、これによりエクストラネットの運用規則や利用者要求を反映した帯域を持つ分散アプリケーション間の論理コネクションを動的に構成し調整するための仕組みの実現を目的とする。

現在、エージェント概念を導入したネットワーク管理システムとしては、ユーザアダプティブエージェント⁷⁾、ENCORE⁸⁾、移動プロクシエージェント⁹⁾などの研究がある。これらはいずれも「ネットワークのQoSを維持する」ことを目的としている。これに対して、本論文では、エージェントの自律性や社会性に基いて「アプリケーション(利用者)の要求を満たすようにネットワークを動的に制御する」ための新たな仕組みを目指しており、従来の研究とは異なる立場からのアプローチといえる。

本論文の構成は、2章で上記課題を解決するためのアプリケーション間コネクションの動的構成機能を提案する。3章では、上記機能を実現するためのミドルウェアであるゲートウェイ層のモデルを提案し、4章で、モデルに基づくゲートウェイ層の機能設計を行う。5章では、各機能を実現するエージェントを設計し、6章において、提案手法に基づく実験とその結果を述べ、本機能の有効性を示す。

2. ゲートウェイ層によるアプリケーション間コネクションの動的構成機能

2.1 エクストラネットとゲートウェイ層

本論文で対象とするエクストラネットは、図1に示すように、複数のイントラネットがゲートウェイ機能を介してインターネットあるいはテレコムネットワークによって接続された論理的なネットワークである¹⁰⁾。

エクストラネットは、これまで図2のような階層構造で構成されていた。下層は、分散コンピューティング

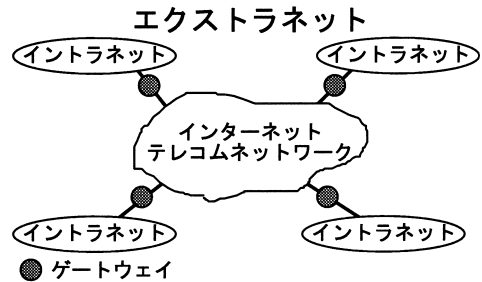


図1 エクストラネット
Fig.1 Extranetwork.

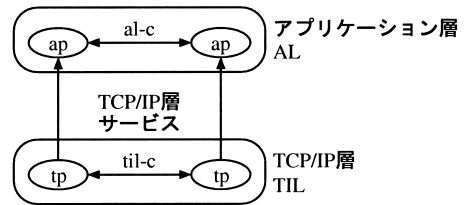


図2 従来型ネットワークサービス階層
Fig.2 Normal network service layer model.

基盤と呼ばれるネットワークサービス階層であり、コンピューティングプラットフォーム、インターネットやテレコムネットワークを含むデジタル通信基盤、ネットワークサービスソフトウェアなどの資源を提供する。この階層は上記のネットワーク資源に基づくTCP/IPプロトコルのサービスを提供する機能から構成され、これをTCP/IP層と呼び、TIL(TCP/IP Layer)と表記する。また、これらのサービスを実現するプロセスをtpで表し、tp間のコネクションをtil-cで表す。一方、上位の階層は利用者業務を遂行するアプリケーションシステムの階層であり、アプリケーション層と呼び、AL(Application Layer)と表記する。ALのプロセスはapで表し、ap間で設定される論理コネクションをアプリケーション間コネクションと呼び、al-cで表す。

本論文では、1章で述べた課題を解決するために、図3に示す中間層を導入し、この階層をゲートウェイ層と呼び、GWL(Gate-Way Layer)と表記する¹¹⁾。GWLは、TCP/IP層サービスを利用して、GWLの論理コネクション gwl-cを実現する。このgwl-cを利用することにより、ALのプロセスapの間にal-cを提供する。

2.2 アプリケーション間コネクションの動的構成機能

本論文では、GWLが以下の機能に基づくアプリケーション間コネクション al-cを提供するとき、これをア

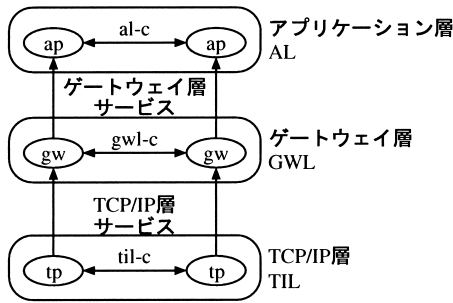


図3 ゲートウェイ層モデル
Fig. 3 Gateway layer model.

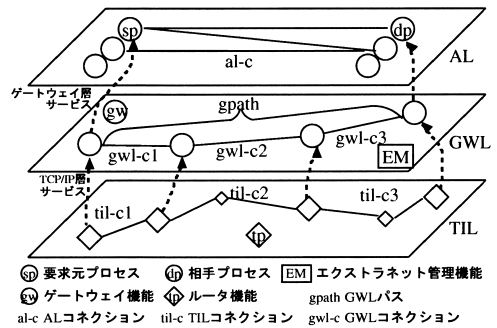


図4 階層とコネクション
Fig. 4 Layer model and connections.

アプリケーション間コネクションの動的構成機能と呼ぶ。

(1) 要求コネクション提供機能

アプリケーションプロセス ap から、所定の帯域を持つ論理コネクション al-c が要求されたとき、エクストラネットが使用できる TIL のコネクションから要求を満たす帯域を持つ論理コネクションを探索し、al-c を提供する。

(2) コネクション再割当て機能

アプリケーションプロセス ap から要求された論理コネクション al-c の帯域が提供できないときは、エクストラネットの運用規則に基づいて、他のアプリケーションプロセスとの帯域使用に関する競合を解消するために、論理コネクションの再割当てを行う。

ここで、エクストラネットの運用規則とは、そのエクストラネットを利用している組織において、目的とする業務を遂行するために定められたサービス提供の優先順位などの規則である。有限なネットワーク資源をエクストラネットに属する多数の利用者が使用する時、たとえば、利用者からの帯域要求などで競合が発生した場合、al-c の帯域の割当てを運用規則に基づいて行うことが、業務全体を円滑に行ううえで有効である。しかしながら、時々刻々発生する ap からのコネクション要求に対して、al-c の帯域の割当てを運用規則に基づいて動的に行う作業は、利用者あるいはエクストラネットワーク管理者の負担となっていた。本論文では、このようなアプリケーション間コネクションの動的構成を支援するミドルウェアとして GWL を設計する。

3. ゲートウェイ層のモデル

エクストラネットは、図4に示すように、アプリケーション層 AL、ゲートウェイ層 GWL および TCP/IP 層 TIL から構成されるものとする。GWL は、TCP/IP 層サービスを利用して、AL のプロセスに対して、アプリケーション間コネクション al-c を動的に提供する

ミドルウェアである。以下、AL、TIL、GWL の順に構成要素を説明する。

3.1 AL: アプリケーション層

AL は、プロセス ap の集合 AP と、その間のアプリケーション間コネクション al-c の集合 ALC から構成される。

$$AL = \langle AP, ALC \rangle$$

$$AP = \{ap \mid ap = \langle pid, type, user, req, state \rangle\}$$

ただし、pid はプロセス識別子、type はプロセスの種類、user は利用者識別子、req は利用者要求を実現するために ap から GWL に与えられるコネクション要求、state は ap の動作状況である。また、ap はさらに要求元プロセス sp と相手プロセス dp に分類され、各 ap は、その利用者インタフェースを介して利用者要求を受理し、これを形式的な要求記述に変換するものとする。アプリケーションプロセス ap の要求表現形式 req(ap) は以下のように定義される。

$$req(ap) = \langle dp, tlimit, bw, user \rangle$$

これは、相手プロセス dp との間に、tlimit で指定された制限時間内に、bw で指定された帯域を持つコネクション al-c を実現することを利用者 user が要求していることを意味する。

また、アプリケーションプロセス ap の動作状況表現形式 state(ap) は以下のように定義される。

$$state(ap) = \langle data, connection, bw \rangle$$

これは、本論文で用いた例題である DB における動作状況表現形式であり、data は格納されているデータ、connection は現在のコネクション数、bw は現在使用している帯域幅を意味する。

また、ALC は AP の要素の間で利用されるアプリケーション間の論理コネクションの集合であり

$$ALC = \{al-c \mid al-c = \langle cid, pid1, pid2, bw \rangle\}$$

ただし、cid は al-c の識別子、pid1、pid2 は al-c を利用する ap の識別子、bw は al-c に与えられた帯域を

表す .

3.2 TIL : TCP/IP 層

TIL は, ルータ機能 tp の集合 TP と, tp 間の TIL 層コネクション til-c の集合 TILC により構成される . すなわち,

$$TIL = \langle TP, TILC \rangle$$

$$TILC = \{ \text{til-c} \mid \text{til-c} = \langle \text{tilc-id}, \text{tp1}, \text{tp2}, \text{tilc-bw} \rangle \}$$

ただし, tilc-id は til-c 識別子, tp1, tp2 は til-c を接続する両端のルータ機能の識別子, tilc-bw は, til-c の帯域である .

3.3 GWL : ゲートウェイ層

GWL は, ゲートウェイ機能 gw の集合 GW, エクストラネット管理機能 EM と, gw 間に構成されるゲートウェイ層コネクション gw-l-c の集合 GWLC から構成されるミドルウェアである . すなわち,

$$GWL = \langle GW, EM, GWLC \rangle$$

$$GW = \{ \text{gw}(i) \mid \text{gw}(i) = \langle \text{gwm}(i), \text{GC}(i) \rangle \}$$

ただし, gw(i) は図 1 に示すように各ゲートウェイに 1 つ存在するものとする . また, gw(i) の構成は図 5 に示すようにゲートウェイマネージャ gwm(i) とゲートウェイコネクタ gc(i, p) の集合 GC(i) により構成される . ただし, i はイントラネット識別子, p は GC(i) における gc の識別子である . また, gc は 1 つの til-c に対して両端で 1 つずつ生成され, gc の生成・消滅は gwm によって管理される . 一方, EM は, エクストラネット全体のネットワーク情報と運用規則を管理するプロセスである . EM, gwm(i), および gc は, 2.2 節で定義された要求コネクション提供機能とコネクション再割当て機能を協調的に実現する機能要素であり, 5 章で示すようにエージェントにより実装される . その機能については 4 章で述べる .

GWLC はゲートウェイ層上のすべてのゲートウェイ層コネクションの集合であり

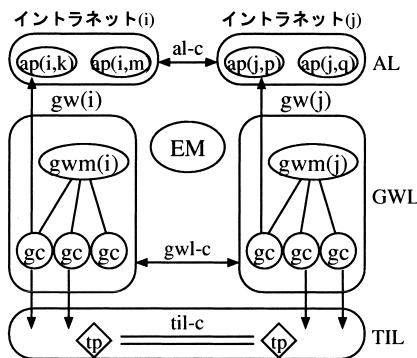


図 5 ゲートウェイ層プロセス gw の構成

Fig. 5 Configuration of gateway layer process: gw.

$GWLC = \{ \text{gw-l-c} \mid \text{gw-l-c} = \langle \text{gwlc-id}, i, j, \text{gwlc-bw} \rangle \}$ と表される . ここで, gwlc-id はコネクション識別子, i, j は両端のゲートウェイの識別子, gwlc-bw はコネクションの帯域の記述である .

AL における sp からの al-c の要求に対して, GWL では, 図 4 に示すように, 複数のゲートウェイ機能 gw が協調して gw-l-c を生成し, それを接続して al-c が実現される . この al-c を実現する GWL における経路を GWL 経路と呼び, 以下のリスト型データ gpath(al-c) で表す .

$$\text{gpath}(\text{al-c}) = (i, k_1, k_2, \dots, k_m, j)$$

ただし, i と j は al-c を要求したプロセスが属するイントラネットの識別子であり, ki は経由したゲートウェイが属するのイントラネットの識別子を順に並べたものである . 次章で, GWL の機能の詳細を定義する .

4. ゲートウェイ層の機能

3 章で定義した構成モデルに基づいて, 2.2 節で提案した 2 つの機能を設計する .

4.1 要求コネクション提供機能

本機能は, 図 6 に示すように, AL の要求元プロセス sp の al-c 要求 req(sp) を受理して, req(sp) で与えられる接続相手のプロセス dp との間で, 指定された帯域を持つ al-c を提供する .

ゲートウェイマネージャ gwm(i) は, プロダクションモデルで記述されたイントラネット i に関する管理・運用規則知識 NMOR(i) を持つ . NMOR(i) の記述形式については 5 章で述べる . gwm(i) は, NMOR(i) と要求コネクション提供手続き (図 7), および gpath 探索手続き (図 8) を用いて, 要求コネクション提供機能を実現する . 以下に各手続きを説明する .

[要求コネクション提供手続き]

[step1] gwm(i) は, 相手プロセス dp が属するイントラネット j の gwm(j) に対して, req(sp) の情報を含んだ接続要求メッセージ request を送信する .

[step2] gwm(j) は, ask-state メッセージを dp に

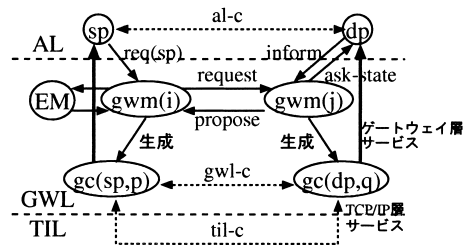


図 6 要求コネクション提供機能

Fig. 6 Function of required connection service.

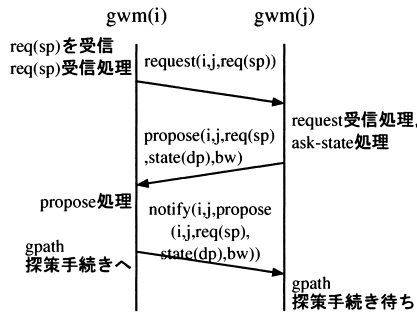


図7 要求コネクション提供手続き

Fig. 7 Procedure for required connection service.

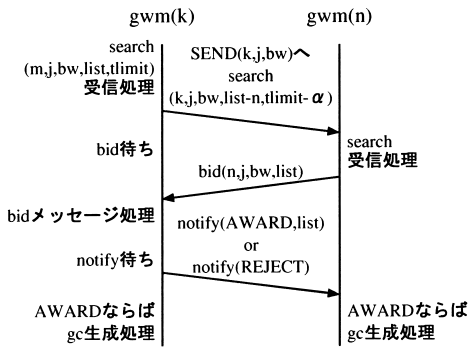


図8 gpath探索手続き

Fig. 8 Procedure for searching gpath.

送信し，inform メッセージにより，dp の動作状況 state(dp) を取得する。

[step3] gwm(j) は，NMOR(j) を用いて，req(sp) と state(dp) の情報から，al-c に対する妥当な帯域を計算し，結果を帯域提案メッセージ propose によって gwm(i) に提案する。

[step4] gwm(i) は，NMOR(i) を用いて，propose メッセージ中の state(dp) と req(sp) から，al-c に対する妥当な帯域を計算し，gwm(j) が提案した bw の受諾の可否を判定する。受諾しない場合には，gwm(i) と gwm(j) は帯域の再計算要求や req(sp) の要求帯域を減少させるために交渉モードに入る。交渉が成功した場合は，gpath 探索手続きを実行し，失敗の場合は，sp への報告後，sp からの再要求待ちとなる。その詳細は煩雑となるので本論文では割愛する。一方，受諾する場合は，gwm(j) へ notify メッセージを送信し，GWL における経路を探索する gpath 探索手続きを実行する。notify メッセージを受信した gwm(j) は受諾されたこと確認し，gpath 探索手続き待ちとなる。gpath 探索手続きが失敗した場合には，gwm(i) は EM に対してコネクション再割当て要求メッセージ reallocation を送信する。コネクション再割当て機能については 4.2 節で述べる。また，コネクション再

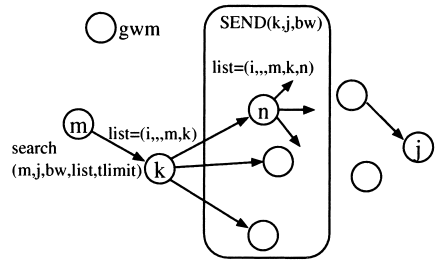


図9 searchメッセージ伝播集合：SEND

Fig. 9 Propagation set of a search message: SEND.

割当て処理も失敗した場合には，gwm(i) は接続要求 req(sp) に含まれる要求帯域を下げ，再び gpath 探索手続きを実行する。

[step5] gpath 探索手続きにより得られた gpath(i, j) を構成するすべての gwm(k) は，al-c を構成する経路 gw-l-c を実現するために，ゲートウェイコネクタ gc(k, p) を生成する。gwm(k) は，gc(k, p) 生成時に，待ち時間 γ を設定する。

[step6] gc(k, p) は，gwm(k) の指示に従って他の gc と協調して sp と dp の間に al-c を提供する。このとき，待ち時間 γ が負になっても gc(k, p) がコネクションを生成できなかった場合，gc(k, p) は消滅する。

[手続き終了]

[gpath 探索手続き]

図9に示すように，gwm(m) から gwm(k) に対して送信される経路探索要求メッセージを search(m, j, bw, list, tlimit) と表す。j は相手プロセス dp の存在するイントラネットの識別子，bw は要求する帯域，list は要求元プロセス sp のイントラネットの識別子から gwm(k) まで search メッセージが送られてきた経路のリスト，さらに，tlimit は gwm(k) が gwm(m) に対して，探索結果の list を bid メッセージにより返送するまでの制限時間であり，この時刻までに bid メッセージを返送できない場合は，gwm(k) は経路探索処理を終了する。また，search(m, j, bw, list, tlimit) を受信した gwm(k) が，gwm(j) までの帯域 bw の経路を探索するために，search メッセージを送信するゲートウェイの識別子のリストを SEND(k, j, bw) で表す。gpath 探索手続きは gwm の協調に基づく以下の3種類の手続きからなる。

[gpath 探索手続き (要求元 gwm(i))]

[step1] 要求元プロセス sp が属するイントラネット i の gwm(i) は，SEND(i, j, bw) に含まれるすべてのゲートウェイマネージャ gwm(k) に対して，経路探索要求メッセージ search(i, j, bw, list, tlimit) を送信する。

[step2] SEND(i, j, bw) に含まれる $gwm(k)$ からの返答メッセージ $bid(k, j, bw, list)$ を待つ .

[step3] 1つ以上の bid が帰ってきた場合, NMOR(i) に含まれる TIL-table (5.3 節) を用いて bid メッセージ中の bw を評価して, 最良の経路を返した $gwm(k)$ を選択し, 採用メッセージ $notify(AWARD, list)$ を送信する . 同時に, 他の $gwm(n)$ には不採用メッセージ $notify(REJECT)$ を送信する . そして, 得られた経路情報を $list$ に代入して $gpath$ を帯域要求手続きに戻す . なお, $tlimit$ 内に bid メッセージがこなければ手続きは失敗となり, $gpath$ は空リストで戻され, reallocation 処理手続き (4.2 節) を実行する [手続き終了]

[$gpath$ 探索手続き (探索経路 $gwm(k)$)]

[step1] $gwm(k)$ が (ただし, $k \neq i, j$ とする) $gwm(m)$ から, 経路探索要求メッセージ $search(m, j, bw, list, tlimit)$ を受信する .

[step2] $search$ メッセージの伝播時間を表す変数 α を参照し, $tlimit - \alpha$ が負でなければ, SEND(k, j, bw) に含まれるすべてのゲートウェイマネージャ $gwm(n)$ に対して, 経路探索要求メッセージ $search(k, j, bw, append(list, n), tlimit - \alpha)$ を送信する . ただし, $append(list, n)$ は, $list$ の末尾にメッセージ通信先の識別子 n を加えたりストである . なお, $tlimit - \alpha$ が負であれば, この経路探索は失敗し, $gwm(k)$ の経路探索は失敗する .

[step3] SEND(k, j, bw) に含まれる $gwm(n)$ からの返答メッセージ $bid(n, j, bw, list)$ を待つ .

[step4] 1つ以上の $bid(n, j, bw, list)$ が帰ってきた場合, NMOR(k) に含まれる TIL-table を用いて bid メッセージ中の bw を評価して, 最良の経路を返した $gwm(n)$ を選択し, $gwm(m)$ に返答メッセージ $bid(k, j, bw, list)$ を送信する . bid 送信後, $gwm(k)$ はタイムアウトのための待ち時間 β を設定する .

[step5] $notify$ メッセージを待つ . そして, 待ち時間 β が負になると, タイムアウトとなり手続き終了となる . また, $notify(AWARD, list)$ を受信した場合, $gwm(n)$ に対して採用メッセージ $notify(AWARD, list)$ を送信し, ゲートウェイコネクタ $gc(k, p)$ を生成する . 一方, 不採用メッセージ $notify(REJECT)$ が受信された場合には, bid メッセージを返送した $gwm(n)$ に対して, 不採用メッセージ $notify(REJECT)$ を送信する . また, $gwm(k)$ は, $gc(k, p)$ 生成時に待ち時間 γ を設定する . この待ち時間 γ が負になっても $gc(k, p)$ がコネクションを生成できなかった場合, $gc(k, p)$ は消滅する [手続き終了]

[$gpath$ 探索手続き (相手 $gwm(j)$)]

[step1] $gwm(j)$ が $gwm(m)$ から, 経路探索要求メッセージ $search(m, j, bw, list, tlimit)$ を受信する .

[step2] $gwm(m)$ に $bid(j, j, bw, list)$ を返送する . bid 送信後, $gwm(j)$ はタイムアウトのための待ち時間 β を設定する .

[step3] $notify(AWARD, list)$ を受信すれば, ゲートウェイコネクタ $gc(j, l)$ を生成する . このとき, $notify(REJECT)$ は無視される . また, 待ち時間 β が負になれば, タイムアウトとなり, 手続きは終了する . [手続き終了]

4.2 コネクション再割当て機能

本機能は, EM に gwm から以下のメッセージが到着したとき, gwm と EM の協調処理に基づいて, エクストラネットのアプリケーション間コネクション集合 ALC に対するコネクション再割当てを行う .

- コネクション再割当て要求メッセージが到着した場合

- コネクション終了メッセージが到着した場合

以下, 各々に対する処理手順を述べる .

(1) コネクション再割当て要求に対する処理

$gwm(i)$ で, $gpath$ 探索手続きが失敗した場合には, $gwm(i)$ から EM に対してコネクション再割当て要求メッセージ $reallocation(i, j, req(sp))$ が送信される . 以下, 本メッセージが到着したときの EM の処理を述べる . ただし, 手続き中のデータの表現形式, ALC 管理用テーブル型データ TALC, プラン P は 5 章で説明する .

[reallocation 処理手続き]

[step1] ALC 管理用テーブル型データ TALC から, 現在使用されている $al-c$ の優先度と使用帯域 bw と $gpath$ を検索する .

[step2] エクストラネット運用知識を使って, $req(sp)$ に通常用いられる $gpath'$ を獲得し, $gpath'$ の生成を目標とするコネクション再割当て目標 ($realloc-goal$) を作成する .

[step3] [step1] で獲得した情報から, 再割当て対象として $gpath'$ と構成 gwm が重なり, 優先度が低く, 使用帯域の近い $gpath$ を選択し, 再割当て目標を達成するプラン P を生成する . P が生成できない場合は [step4] へ進む . P が生成できた場合は [step5] へ進む .

[step4] $gwm(i)$ にコネクション再割当てが不可能であることを通知するメッセージ $reject$ を送信して手続きを終了する .

[step5] プラン P で指定された $gpath$ および $gpath'$

に含まれる各 gwm に対して、図 10 に示す提案メッセージ propose を送信する。

[step6] もし逆提案メッセージが受信されれば、図 11 に示す逆提案プロトコルを実行する。

[step7] [step5] と [step6] の結果、すべての受信メッセージが受諾を表す accept であれば [step8] に進む。ただし、拒否を表す refuse メッセージが 1 つでも含まれる場合には、各 gwm に提案をキャンセルするメッセージ cancel を送信し [step4] に戻る。

[step8] gpath および gpath' に含まれる gwm に対して、プラン P を実行するために、メッセージ generate -gc を送信し、TALC を書き換える。

[step9] generate-gc を受信した gpath に含まれる各 gwm は、al-c から al-c' への切替え処理を実行する。gpath' に含まれる各 gwm は、gc を生成して、al-c の構成処理を実行する [手続き終了]

gwm は、EM からのメッセージを処理する手続きを持つ。その概要が、図 10, 11 に示されている。この手続きは、プランを 1 回のみ生成し、そのプランが関係する gwm に受諾されなかった場合には、コネクション再割当て処理は失敗となる。

(2) コネクション終了に対する処理

アプリケーション間コネクションの生成と終了は、gwm から EM への通知により行われ、その情報は EM で管理される。また、ALC 管理テーブル TALC には、各コネクション al-c に対するアプリケーションプロセス sp からのコネクション要求 req(sp) と、実際に割り当てられた帯域 bw が管理されている。EM は、これらの情報に基づいて、再割当て優先順位のテーブル realloc-table を作成する。TALC と realloc-table の内容は 5.4 節で述べる。以下、アプリケーション間コネクション終了メッセージ release(i, j, req(sp), al-c) の到着に対する処理を示す。

[release 処理手続き]

[step1] TALC から al-c の帯域 bw(al-c) を検索する。

[step2] realloc-table を優先順位の高い順に検索し、帯域 bw(al-c) より狭い帯域を要求しているコネクション al-c' を探す。これを利用しているプロセスを sp とする。コネクションがあれば [step3] へ進む。なければこの手続きを終了する。

[step3] 終了したコネクション al-c を提供していた gwm に、sp に対する al-c の提供を提案するメッセージ propose を送る。続いて、TALC から al-c' を提供している gwm に対して、コネクションの切替えを提案するメッセージ propose を送る。

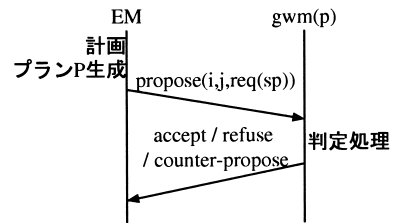


図 10 提案メッセージとレスポンス
Fig. 10 Proposal message and response.

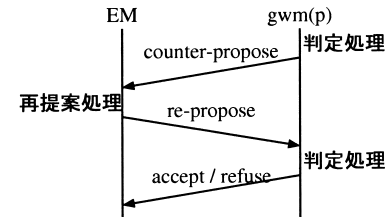


図 11 逆提案メッセージと処理
Fig. 11 Counteroffer message and process.

[step4] 提案に対するすべての応答が受諾を表す accept であれば [step5] に進む。もしも拒否を表す refuse メッセージが 1 つでも含まれれば、propose を送信した gwm に対して提案をキャンセルするメッセージ cancel を送信し、この手続きを終了する。

[step5] sp に対するコネクションを al-c' から al-c に切り替える処理を実行する [手続き終了]

5. エージェント型ゲートウェイ層の設計

5.1 エージェントのアーキテクチャ

ゲートウェイ層の実装においては、エージェント型分散処理システム的设计開発用フレームワークである ADIPS フレームワーク¹²⁾を用いる。ADIPS フレームワークにおけるエージェントは以下の 3 つのモジュールから構成される。

(1) 通信モジュール: メッセージの送受信の制御とメッセージのパケット・デパケットを行う。

(2) Java プログラムモジュール: コンピューティングプラットフォームに対するエージェントの作用(アクション)を実現する手続きプログラムの集合であり、手続きを Java 言語でプログラムする場合と、コンピューティングプラットフォームの既存機能を Java でラッピング(インタフェース)する場合の 2 通りの方法が用意されている。本論文では、TIL 層への制御や監視機能は、上記の 2 つの方法を使って実装している。

(3) エージェントプログラムモジュール: エージェントプログラムとインタプリタから構成される。エー

エージェントプログラムは、ADIPS フレームワークのエージェントプログラミング言語 ADIPS/R を用いて記述される^{13),14)}。ADIPS/R はプロダクションモデルに基づく言語で、条件部にメッセージやイベントに関するパターンを記述し、アクション部でメッセージの送信や手続き呼び出しなどのアクションパターンが記述できる。また、ADIPS フレームワークのエージェント間通信言語は KQML¹⁵⁾ に基づいている。

5.2 エージェントの基本知識記述形式

4章で述べた各種の手続きは、プロダクションモデルのルールにより記述される。たとえば、手続きに含まれる「あるパターンのメッセージを受信したときの動作（メッセージ生成や送信など）」はプロダクションルールのアクション部に記述される。ADIPS/R では、通常のプロダクションシステムと同様に、条件部とファクト部は OAV(Object-Attribute-Value) の形式で記述される。さらに、4章で述べた手続きでは、ネットワーク管理者が持つ種々の経験的知識も利用する必要がある。そこで、これらの知識を効果的に記述するために、以下の記述形式を用意した。

(1) テーブル型データ

テーブル形式で定義されるデータの蓄積と検索を行うデータ構造と処理手続きは、エージェントの Java プログラムモジュール部に実装される。ルールのアクション部では、define-table, data-input, retrievable などのアクションが定義され、データの検索結果は、プロダクションモデルのファクトとしてワーキングメモリに格納される。

(2) リスト型データ

テーブル型データと同様に、データ構造と処理手続きが Java モジュール部に実装され、その操作は append, car, cdr などのアクションを用いて行われる。

(3) プラン型データ

エージェントは役割や目標を持つが、それを実現する手続きの系列はプランとして蓄積される。エージェントのプランモデルは DSPL¹⁶⁾ に基づいて設計され、プランを構成するために、次のようなルールが定義された。(a) 目標を副目標の系列に分割するルール、(b) 目標を行動の系列に分割するルール、(c) 例外処理を行うルールである。ルールが適用される前提条件は、ネットワークの状況とエージェントの状態により与えられる。そして、これらの条件を満足したとき、副目標や行動系列の展開が実行される。展開の結果は後処理手続きを起動して検査され、展開結果が所定の後処理条件を満たさない場合には、例外処理のルーチンが起動される。このようなプランを用いることにより、

同じ目標であっても、状況に応じて異なる行動の系列を生成することができる。以下、こうした行動の系列を、目標 g に対するプランと呼び、 $P(g)$ と表記する。そして、ある目標 g に対して生成されたプラン $P(g)$ に基づく行動は、 $P(g)$ を構成するルールを順次実行することにより実現される。

5.3 ゲートウェイマネージャの知識記述

(1) イン트라ネット i の運用知識 NMOR(i)

各イン트라ネットでは、ネットワークの規模、性能、業務が異なり、また、そのゲートウェイが他のイン트라ネットのゲートウェイに接続される際の回線の帯域幅も異なる。さらに、この状況が変更されることも予想される。したがって、エクストラネット全体の運用を柔軟に行うために、本手法では、イン트라ネットの情報は、各ゲートウェイマネージャが局所的に管理し、エクストラネット全体のアプリケーション間コネクションの制御は、エクストラネットマネージャ EM と、ゲートウェイマネージャとの協調により動的に実現する。

以下に、ゲートウェイマネージャの運用知識 NMOR(i) に含まれる主な情報を示す。

(a) TIL-table(i): 他のゲートウェイとの間の TIL における回線の帯域情報。bid 評価に用いられる(テーブル型)

TIL(i) の項目名 = (回線識別子, 接続先ゲートウェイ識別子, 帯域)

(b) sp-table(i): イン트라ネット i に含まれるコネクション要求を行うプロセスの情報(テーブル型)
sp-table(i) の項目名 = (プロセス識別子, 相手プロセス識別子, 優先順位, 標準帯域, 最低帯域, 希望帯域)
ここで、最低帯域とは、このプロセスが動作するために必要な最低の帯域であり、希望帯域とは、可能であれば確保したい帯域を表す。

(c) 要求コネクション提供手続き処理

要求コネクション提供手続き [step3] において、gwm(i) からの request メッセージに含まれる bw と、dp からの inform メッセージに含まれる bw' から、al-c の帯域幅を決定するルール集合が、gwm(j) に記述される。要求コネクション提供手続き [step4] における提案受諾の判定処理も、同様に gwm(i) のルール集合として記述される。

(2) gpath 探索手続きの知識

gwm(k) は、帯域を確保するためのネットワークパスに関する以下のテーブル型の情報 path-bw を持つ。path-bw の項目名 = (目標ゲートウェイ識別子, 帯域上限, 帯域下限, gwm-list)

ここで、gwm-list は、目標ゲートウェイに対して、帯域下限から帯域上限の範囲でアプリケーション間コネクションを提供する可能性のある隣接ゲートウェイの識別子のリストを表す。gwm(k) が gwm(m) からメッセージ search(m, j, bw, list, tminit) を受信すると、path-bw から、目標ゲートウェイ j, および、帯域 bw が帯域上限と帯域下限の間に入る gwm-list が選択され、SEND(k, j, bw) に代入される。

また、gwm(k) には、bid(m, j, bw', list) メッセージを受信したとき、NMOR(k) の TIL-table(k) を用いて bw' が提供可能かどうかを評価するルールの集合も格納される。

5.4 エクストラネットマネージャの知識記述

EM が持つ主な知識の内容を以下に示す。

(1) エクストラネットの運用知識

各イントラネット間で通常構成される gpath の情報、および、各イントラネット間の優先度に関する情報を持つ。

(2) TALC

gwm は、al-c の生成時と終了時に、EM に対してその事象を通知する。EM はこれをテーブル型データ TALC で管理する。

TALC の項目名 = (コネクション識別子, gpath, bw, req(sp), 優先順位)

(3) realloc-table

ある al-c が終了した時点で、空いている帯域の使用を希望するアプリケーションプロセス sp の再割当ての優先度を記録するテーブルである。

realloc-table の項目 = (プロセス識別子, 再割当て優先度, コネクション識別子, 割当て帯域, 再割当て希望帯域)

割当て優先度と再割当て希望帯域は、現在割り当てられている帯域、sp が存在するイントラネットの gwm が管理している sp-table の標準帯域、および sp の優先度から得られる。

5.5 ゲートウェイコネクタの知識記述

ゲートウェイコネクタ gc は、図 5 に示すように、gwm により生成される。gc は、gwm より与えられる制御パラメータに従って TCP/IP 層プロセス tp を制御して、相手 gwm との間で所定の帯域を持つ til-c を構成するための知識を持つ。これらの知識は、5.2 節で示したルール型知識により記述され、各ルールのアクション部から Java プログラムモジュールを起動して tp を制御する。また、gc は、gwm から与えられた待ち時間 γ が負になった場合、gwm へ消滅メッセージを送信した後、消滅する。

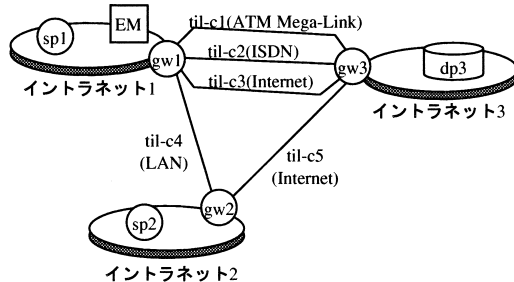


図 12 実験エクストラネット構成

Fig. 12 Composition of experiment extranetwork system.

6. 実験システム

6.1 実験システムの構成

本論文で提案したアプリケーション間コネクションの動的構成手法の実現可能性を検証し、その基本特性を明らかにすることを目的として、実験システムの試作と実験を行った。本システムの設計においては、エージェントに組み込まれる知識は、標準的なネットワーク管理・運用の専門家(1人)から抽出・獲得された経験的知識を用いている。また、各種パラメータの設定は、本方式に基づくシステムの基本的な振舞いを確認するために、最も基本的な設定となっている。

なお、エージェントに組み込まれる知識が、本試作における専門家とは異なる知識源から獲得される場合には、以下の各節で述べるルールやプランの構成や個数が異なったり、その結果として、システムの処理時間などが変化したりする可能性がある。これは知識型システムにおいては本質的な性質となっており、実用的なエキスパートシステムなどでは工夫のしどころといえる。この問題は、本手法の実用化を目指す際に別途検討すべき項目の1つとなるが、本論文において実験システムの試作や提案手法の機能的な検証を行う際には別問題としてとらえることができ、これが本論文での議論において、本質的な障害とはならないことに注意しておきたい。

本実験システムでは、ゲートウェイ層 GWL が実現するコネクションの動的構成サービスの機能を検証・評価するためにエクストラネットとして、図 12 に示すように、東北大学の研究室のイントラネット 3 と千葉工業大学の研究室のイントラネット 1, イントラネット 2 から構成される実験システムを構築した。このエクストラネットにおいて、イントラネット 3 に用意したマルチメディアデータベース dp3 をイントラネット 1, 2 上のアプリケーションプロセス sp1, sp2 がネットワークを介して利用する実験を行った。各イントラ

	C1	C2	C3	C4	C5	C6	C7	C8	C9
sp1	1K	64K	10M	1K	64K	10M	1K	64K	10M
sp2	1K	1K	1K	64K	64K	64K	10M	10M	10M

図 13 要求パターン

Fig. 13 Request pattern.

ネット間には、TCP/IP 層コネクションとして、til-c1 ~ til-c5 という 5 種類のコネクションが利用できるように設定した。各コネクションは、ATM Mega-Link, ISDN, Internet, および LAN の 4 種類であり、それぞれ、最大帯域は 10 Mbps (ATM), 64 Kbps (ISDN), 1 ~ 5 Kbps (Internet), および 100 Mbps (LAN) である。また、sp1, sp2 は、マルチメディアデータベースから検索されるデータのタイプに応じて、それぞれ所定のネットワーク帯域の確保を指定してコネクション要求を発行する。各 sp が要求するデータタイプと帯域要求は、動画 (10 Mbps), 静止画 (64 kbps), テキスト (1 kbps) であり、sp1, sp2 が発行する帯域要求のパターンは図 13 に示すものとした。

一方、各 gw 上には、ゲートウェイ層を構成/制御するために、前章までに設計したゲートウェイマネージャ gwm, ゲートウェイコネクタ gc の 2 種類のエージェントが配置される。また、gw1 上にはイントラネット 1, 2, 3 からなるエクストラネットの管理・運用規則知識を持つエクストラネットマネージャ EM が存在する。

6.2 ゲートウェイ層 GWL の実装

GWL を構成する各エージェントは、ADIPS フレームワークを用いて実現された。各エージェントには 5 章で述べた知識が格納されている。

各 gwm と EM が持つエクストラネットやイントラネットの管理運用知識や各手続き用の知識は、5 章で定義した記述形式に基づいて、テーブル型知識として記述される。たとえば、EM の持つネットワーク運用者の知識である TALC は図 14 のように記述される。ただし、前述したように、本実験システムではシステムパラメータの設定は簡素化されている。たとえば、図 14 の priority の項の値は、“low” と “high” が設定されているが、実用性を高めるためには、より木目の細かい設定が必要であろう。これらは今後の検討課題である。

さらに、EM が行うコネクション再割当て機能の際に用いるプラン P は、図 13 に示した要求パターンに従って sp1, sp2 の使用する帯域を設定することにより生成される。

本実験システムで実現した GWL は、エクストラネッ

```
(talc :alc-id al-c1 :gpath "1,3" :bw 10Mbps
: req(sp) "req(sp1)=<dp3, 10Mbps, susu>" :priority high)
(talc :alc-id al-c2 :gpath "2,1,3" :bw 64Kbps
: req(sp) "req(sp2)=<dp3, 10Mbps, konno>" :priority low)
```

図 14 TALC の記述例

Fig. 14 Example of TALC.

```
(rule rereceive-req
(Message :performative request
: dp dp3 :t-limit 100sec :bw 10Mbps :user konno)
-->
(SendMessage :performative ask-state :to dp3)
)
(rule create-propose
(dp-state :state no_problem :bw 10Mbps)
-->
(SendMessage :performative propose
: req req(sp1) :state dp-state :bw 10Mbps)
)
```

図 15 提案メッセージ作成ルール

Fig. 15 Creation rule of a proposing message.

トマネージャ、ゲートウェイマネージャ × 3, ゲートウェイコネクタ × 10 の計 14 個のエージェントによって構成されている。

6.3 要求コネクション提供機能の検証

要求コネクション提供機能を評価するために、sp2 から gwm2 に対して以下のような al-c 要求を与えて実験を行った。

```
req(sp2)=(dp3, 100 sec, 10 Mbps, konno)
```

これは、sp2 が dp3 に対して利用者 konno で 10 Mbps の帯域を持つコネクションを 100 秒以内に用意してほしいという内容を表現している。gwm2 は sp2 から要求を受け取ると、要求コネクション提供手続きに基づき、gwm3 との協調が開始される。gwm3 は、NMOR(3) により提供可能な帯域を計算し、その結論として 10 Mbps コネクションが必要であるとの提案メッセージを gwm2 に送信する (図 15)。gwm2 は、その提案を受諾し、知識として格納された gpath 探索手続きを実行し、経路探索要求 search(2, 3, 10 Mbps, “gwm2”) を gwm1 へ送信する。さらに、gwm1 は、gpath 探索手続きに従って、経路探索要求を gwm3 へ送信する。その結果、gwm3 から gwm1 へ bid(gwm1, gwm3, 10 Mbps, “gwm1, gwm3”) が送信され、gwm1 から gwm2 へも bid(gwm2, gwm3, 10 Mbps, “gwm2, gwm1, gwm3”) が送信される。その後、bid に対する受諾メッセージ notify が受信されると、各 gwm は各々の環境に gc を生成し、sp2 より要求された経路が設定される。要求された経路を提供するまでの処理

時間は平均して約 7 秒であった。

この実験により、要求コネクションを、エージェント型ミドルウェアであるゲートウェイ層 GWL が、ネットワーク運用者に代わって行えることが確認された。なお、より複雑な構造のネットワークでの運用や高速処理を目指す際には、知識の表現方式や知識処理系の高速化などが必要となるが、これらは今後の課題である。

6.4 コネクション再割当て機能の検証

コネクション再割当て機能を評価するために、sp2 が経路を使用中に、優先度の高い sp1 から、同じ帯域を持つコネクションに対して利用要求を発行し、要求コネクションの再割当てに関する実験を行った。まず、sp2 が dp3 に対して利用者 konno で 10 Mbps の帯域を持つコネクションを 100 秒以内に用意してほしいという要求、すなわち、

$$\text{req}(\text{sp2}) = \langle \text{dp3}, 100 \text{ sec}, 10 \text{ Mbps}, \text{konno} \rangle$$

という要求を満たしているコネクションを sp2-dp3 間で使用しているとする。このとき、sp1 より

$$\text{req}(\text{sp1}) = \langle \text{dp3}, 50 \text{ sec}, 10 \text{ Mbps}, \text{susu} \rangle$$

という要求が gwm1 へ通知される。これは、sp1 が dp3 に対して利用者 susu で 10 Mbps の帯域を持つコネクションを 50 秒以内に用意してほしいという要求を表現している。この新たな要求に対して、gwm1 は gpath 探索手続きを行うが、10 Mbps のコネクションが gwm2 らによって使用されているため、gpath 探索手続きは失敗する。そこで、gwm1 は EM に対してコネクション再割当て要求メッセージを送信し、EM は TALC や運用知識に基づいて、図 16 に示すルールからなるコネクション再割当てプランの行動系列を生成する。その後、EM は各 gwm に対して提案メッセージを送信し、受諾後にコネクション変更処理が実行されて新しいコネクションが構成される。

さらに、sp1 のコネクションが終了すると、gwm1 からレポートが送付され、EM は、再び sp2 の帯域を 10 Mbps の帯域幅を持つコネクションに変更する。

上述したコネクション再割当ての処理に要した時間は、コネクション変更処理前まで平均して約 5 秒で

あった。また、広帯域から狭帯域へのコネクション変更処理は約 2 秒で終了するが、逆に、狭帯域から広帯域への変更には数分を要した。これは、コネクション維持のために行われる Java プログラムのバッファリング処理によるものであることも確認された。

この実験により、従来困難であったエクストラネット運用中における要求コネクションの再構成を、エージェント型ミドルウェアであるゲートウェイ層 GWL が、ネットワーク運用者に代わって実行できることを確認した。なお、前述したように、これらの処理の高速化などは今後の課題である。

一方、本手法では、基盤となる TCP/IP 層の規模や構成が変化した場合には、新たに、変化に対応した管理者の経験的知識を、EM や gwm の知識として追加・更新しておくことが必要である。これは、TCP/IP 層の構成上の変化はエクストラネットそのものの変化としてとらえようとしていることによる。こうした TCP/IP 層の構成の変化や知識量の増加については、今後、エージェントの処理系の改良を含めて、より柔軟に対処できる手法に関する検討を重ねてゆく予定である。

7. おわりに

エクストラネット運用者は、遂行される業務に依存したネットワークの設定、あるいは業務や利用者要求の変更に基づく設定の変更などを行う。しかし、これらの作業は、ネットワーク運用者に対する大きな負担となっている。本論文では、こうしたネットワーク運用者の作業の一部を代行することにより、その負担の軽減を図る手段として、アプリケーション間コネクションの動的構成を支援する手法、および、これに基づくエージェント指向ミドルウェアを提案した。そして、インターネットとテレコムネットワークを含むエクストラネットを用いた実験システムを構築することにより、アプリケーションの要求に応じて、論理コネクションが自動的に提供できること、また、複数の利用者要求間で競合が発生した場合、新たなコネクションを確立するコネクション再割当てが動的に行えることを実験システムにより検証した。

本手法の導入により、ネットワーク運用者の持つ種類の知識を効果的に活用した作業代行が可能となり、その結果、運用者の負担軽減に寄与することが確認された。今後、

- 利用者要求モデルの確立
- ネットワーク帯域の実時間変動に対するコネクション再割当て方法

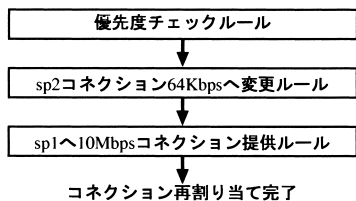


図 16 コネクション再割当てプラン

Fig. 16 Creation plan of reallocation.

などに関する検討を継続して進めていきたい。

謝辞 本研究は、日本学術振興会未来開拓学術研究推進事業・動的ネットワーキングプロジェクトの支援により行われた。

参 考 文 献

- 1) Barksdale, J.: The Next Step: Extranets (1996). Netscape Communications Corporation: Netscape Columns: The Main Thing.
- 2) Goldmann, N.: Extranet: The Third Wave' in the Internet Electronic Commerce, *Journal of Internet Banking and Commerce* (1997).
- 3) Hoffman, K., Pallansch, P. and Shafer, S.: Establishing a Medical Knowledge Extranet to Enhance MEditional Communications, *Drug Information Journal*, Vol.34, pp.1063-1068 (2000).
- 4) 加治屋安彦: インターネットの道路情報分野への活用とそのインパクト, *土木学会誌*, Vol.83, No.5, pp.18-21 (1998).
- 5) Shiratori, N., Sugawara, K., Kinoshita, T. and Chakraborty, G.: Flexible Networks: Basic Concepts and Architecture, *IEICE Trans. Commun.*, Vol.E77-B, No.11, pp.1287-1294 (1994).
- 6) Suganuma, T., Kinoshita, T. and Shiratori, N.: Flexible Network Layer in Dynamic Networking Architecture, *International Workshop on Flexible Network and Cooperative Distributed Agents (FNCA2000)* (2000).
- 7) 中沢 実, 須田飛志, 阿部倫之, 服部進実: ユーザアダプティブエージェントによる自律的ネットワーク QoS 制御とその評価, *電子情報通信学会論文誌*, Vol.J81-B-I, No.2, pp.58-69 (1998).
- 8) 明石 修, 菅原俊治, 村上健一郎, 丸山 充, 高橋直久: マルチエージェントを用いた自律組織間診断システム: ENCORE, *情報処理学会論文誌*, Vol.40, No.6, pp.2659-2668 (1999).
- 9) 富川祐樹, 高井昌彰: 移動プロキシエージェントによるアクセス経路選択, *電子情報通信学会論文誌*, Vol.J84-D-I, No.2, pp.183-190 (2001).
- 10) 今野 将, 菅沼拓夫, 菅原研次, 木下哲男, 白鳥則郎: マルチエージェントシステムを用いた利用者指向コネクションの実現方式, *信学技報*, Vol.100, No.529, pp.7-12 (2001).
- 11) 今野 将, 菅沼拓夫, 菅原研次, 木下哲男, 白鳥則郎: 動的ネットワーキングのためのエージェントを用いた経路制御方式, *信学技報*, Vol.100, No.321, pp.1-8 (2000).
- 12) 藤田 茂, 菅原研次, 木下哲男, 白鳥則郎: 分散処理システムのエージェント指向アーキテクチャ, *情報処理学会論文誌*, Vol.37, No.5, pp.840-852 (1996).
- 13) 原 英樹, 今野 将, 菅原研次, 木下哲男: ソフ

トウェアエージェント開発教育用システム TAF の設計と実装, *電子情報通信学会論文誌*, Vol.J84-D-I, No.8, pp.1202-1210 (2001).

- 14) 木下哲男(編): エージェントシステムの作り方, *電子情報通信学会* (2001).
- 15) Finin, T., Waber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McGuire, J., Shapiro, S. and Beck, C.: *DRAFT Specification of the KQML Agent-Communication Language* (1993).
- 16) Brown, D.C. and Chandrasekaran, B.: *Design Problem Solving: Knowledge Structures and Control Strategies*, Morgan Kaufmann Publishers (1989).

(平成 13 年 5 月 28 日受付)

(平成 13 年 12 月 18 日採録)



今野 将 (正会員)

1973 年生。2001 年千葉工業大学大学院博士後期課程情報工学専攻期間満了退学。現在、東北大学電気通信研究所助手。エージェント指向コンピューティングに興味を持つ。IEEE、電子情報通信学会各会員。



北形 元

1972 年生。1999 年東北大学大学院情報科学研究科博士前期課程修了。現在、同大学院同研究科博士後期課程在学中。エージェント指向コンピューティングに興味を持つ。電子情報

通信学会会員。



原 英樹 (正会員)

1970 年生。1998 年千葉工業大学大学院博士後期課程情報工学専攻満期退学。同年千葉工業大学助手。現在、千葉工業大学情報科学部情報ネットワーク学科講師。博士(工学)。ソフトウェア再利用, エージェント指向コンピューティングに興味を持つ。電子情報通信学会会員。



菅沼 拓夫 (正会員)

1966年生。1997年千葉工業大学大学院博士後期課程情報工学専攻修了。現在、東北大学電気通信研究所助手。博士(工学)。やわらかいネットワーク, エージェント指向コンピュー

ティングに興味を持つ。IEEE, 電子情報通信学会各会員。



菅原 研次 (正会員)

1950年生。1980年東北大学大学院博士課程中退。同年千葉工業大学助手。現在、千葉工業大学情報科学部情報ネットワーク学科教授。工学博士。1992年日本工業教育協会功

績賞。1994年本会山下記念賞受賞。1996年情報処理学会論文賞 2000年度電子情報通信学会業績賞各受賞。電子情報通信学会, 人工知能学会, IEEE 各会員。



木下 哲男 (正会員)

1953年生。1979年東北大学大学院修士課程修了。同年、沖電気工業(株)入社。1996年東北大学電気通信研究所助教授。2001年同大学情報シナジーセンター教授。知識工学,

エージェント技術, 知識情報システム等の研究開発に従事。工学博士。1989年度情報処理学会研究賞, 1996年度同論文賞, 2000年度電子情報通信学会業績賞各受賞。電子情報通信学会, 人工知能学会, 日本認知科学会, AAAI, IEEE, ACM 各会員。