

経路障害診断システム用エージェント管理システムの提案

寺内敦[†] 明石修[†] 丸山充[†] 福田健介[†] 栗原聡[†] 菅原俊治^{††}
[†]NTT 未来ねっと研究所 ^{††}NTT コミュニケーション科学基礎研究所
{terauchi, akashi, mitsuru, fukuda, kurihara, sugawara}@core.ntt.co.jp

概要

マルチエージェントによる AS 間経路障害診断システム ENCORE の scalability の向上および柔軟な診断を実現するためのエージェント管理システム ARTISTE を提案する。ARTISTE は複数のエージェントにより ENCORE エージェントや BGP トポロジなどのネットワークに関する情報を自律的に収集・解析し、その情報を利用して ENCORE エージェントに適切な役割を環境の変化に応じて割り当てることで ENCORE におけるエージェント協調を支援し、scalability の向上および柔軟な診断を実現する。

ARTISTE: an Agent Role Management System for inter-AS Routing Diagnosis

Atsushi Terauchi[†] Osamu Akashi[†] Mitsuru Maruyama[†] Kensuke Fukuda[†]
Toshiharu Sugawara^{††} Satoshi Kurihara[§]
[†]NTT Network Innovation Laboratories. ^{††}NTT Communication Science Laboratories
[§]Osaka University
{terauchi, akashi, mitsuru, fukuda, kurihara, sugawara}@core.ntt.co.jp

Abstract

This paper proposes an agent role management system for the inter-AS routing diagnosis system called ENCORE. The proposed system called ARTISTE analyzes information about network and ENCORE agents' status and assign roles to ENCORE agents. This dynamic role allocation effectively supports cooperative action among ENCORE agents and improve the scalability of ENCORE.

1. はじめに

近年のインターネットの急速な拡大は、その運用管理を困難かつコストの掛かるものとし、さまざまな問題が発生している。そのような問題の中でも、BGP(Border Gateway Protocol)[1]を用いた Autonomous System(以下、AS)間での経路制御における障害は、その発見や調査において AS 間での相互作業が必要であるため対策が難しく、また障害の影響が広範囲に渡ることもあるため、特に有効な対策が求められている。

この経路情報障害に対処するために、広域 IP 網経路障害診断システム ENCORE (Inter-AS Diagnostic Ensemble System using Cooperative Reflector Agents) が提案されている[2]。ENCORE は複数の AS に配置されたエージェント (以下、ENCORE エージェント) が協調することにより、AS 間の経路障害の検出、診断を行うシステムである。ENCORE が診断を効率的に柔軟に実行するためには、診断の内容や環境 (ENCORE エージェントやネットワーク) の状態に応じて適切なエージェント群を選んで、そのエージェント群が協調作業を行う必要がある。

しかし、現在の ENCORE では診断毎に協調すべきエージェント群の選択を手動で行っているため、以下のような場合にエージェント群を適切に選択することが難しい

という問題があった。

- ENCORE エージェントの数が多の場合
- ENCORE エージェントの状態やネットワークの状態が頻繁に変動する場合

この問題を解決するために、エージェント管理システム ARTISTE (Agents' Roles and Topological Information management System for agent TEamwork) を提案する。ARTISTE は、ENCORE エージェントや AS 内での BGP スピーカからエージェントの状態や BGP トポロジなどのネットワーク情報を自律的に収集・解析して、ENCORE エージェントに対する適切な役割(role)を割り当てる機能を持つ。ARTISTE を用いると、ENCORE は診断を行う際に協調すべきエージェント群をほぼ自動的に取得できるようになり、scalability の向上や環境に応じた柔軟な診断が可能になる。

本稿では、ARTISTE のシステムアーキテクチャや実システム化について説明した後、プロトタイプを作成して実ネット上に適用した検証結果について述べる。

2. 経路障害診断システム ENCORE

本節では、経路障害診断システム ENCORE の概略について説明する。

2.1. ENCORE

広域 IP 網経路障害診断システム ENCORE は、AS 間において発生した経路障害を迅速に検知、診断を行うためのシステムである。経路障害の特徴として、ある AS によって発行された経路情報はネットワークを伝搬する過程において各 AS のポリシーに基づいて随時変更される可能性があり、そのため、経路の発行元の AS ですら自分の経路が一旦ネットワーク上に流れてしまった後の変化を把握することができず障害の内容や原因を単独で推測することが困難である、という性質が挙げられる。このような性質を持つ経路障害に対応するために、ENCORE ではネットワーク上の複数の AS にエージェントを配置して、それらのエージェントが協調して経路情報を相互に観測したり情報交換することによって経路障害の検知や診断を行っている。ENCORE が診断の対象とする経路情報の伝搬障害とその診断手順の簡単な事例[5]を図 2-1 で説明する。

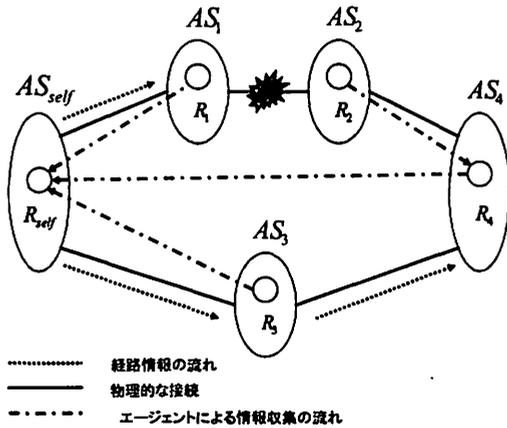


図 2-1 ENCORE による経路診断の例

図中において、 AS_{self} から advertise された経路情報は AS_1 , AS_2 あるいは AS_3 を経由して AS_4 に伝搬されるものとする。ここで、 AS_1 と AS_2 との間で AS_{self} に関するフィルタの設定に誤りが発生して、 AS_{self} に関する経路情報が AS_2 に流れなくなった状況を想定する。この状況を従来の方法で、 AS_{self} から診断ツールなどを用いて調査を行っても、 AS_1 から先へとパケットが到達しないという以上の解析は不可能である。つまり、 AS_1 , AS_2 間の接続の障害なのか AS_2 自身のフィルタの設定誤りなのか判断できない。

これに対して、ENCORE を用いた診断では、 AS_{self} 中のエージェント R_{self} が AS_1 , AS_2 のエージェント R_1 , R_2 と通信し、それぞれの AS において観測された経路情報を送り返してもらう。その結果、 AS_1 では自分の advertise した経路情報が存在するのに対し、 AS_2 では AS_1 の経路情報はあっても自分の経路情報が含まれていないことが分かる。以上より、 AS_{self} では AS_1 と AS_2 間で AS_{self} の経路情報に関するフィルタの設定に誤りがあるということが推論できる。

2.2. エージェントの role

ENCORE では、診断における基本機能を 3 つ定義しており、診断に関わるエージェントはこれらの機能のいずれかを実行する。このエージェントと実行する機能との関係を表す概念がエージェントの役割(role)である。すなわち、エージェントがある role を持つということは、そのエージェントが、その role に求められているタスクを実行可能であることを意味する。現在の ENCORE においては、以下の 3 つの role が定義されている。

- investigation・・・診断の主体となる ENCORE エージェントからの問い合わせに対して、指定された経路情報を取得して、問い合わせ元のエージェントに返答する。
- neighbor・・・早期の障害発見のために診断の主体となる ENCORE エージェントと相互監視、通知および診断を行うために設定するエージェント群である。このエージェント群は、例えば、契約を結んで自分とトラフィックを互いに交換しているピアの AS や指定されたネットワーク上の地点の「近傍」に存在するエージェント群に設定する。
- relay・・・ENCORE エージェント間のメッセージ通信を中継する。ARTISTE の動作する状況では、IP レベルでの経路が正常であるという前提はおけないため、本 role を持つエージェントによりアプリケーションレベルでの経路を確保する。

3. エージェントの role の管理

3.1. ENCORE のエージェント協調における課題

前述の通り、ENCORE が診断を効率的かつ柔軟に実行するためには、診断の状況に応じて適切なエージェント群を選び、その群内のエージェントに対して 2.2 で述べた role を適切に割り当てる必要がある。また、エージェント群の選択や role の割り当てはエージェントの状態やネットワークの状態を十分考慮して行う必要がある。例えば、図 2-1 で示した状況においては図中にある AS 同士が協調することが問題発見のために最も効率的であり、それ以外のエージェントと協調したとしても有益な情報が得られる可能性は少ない。現在の ENCORE では協調すべきエージェント群の選択および各エージェントへの role の割り当ては、ネットワークの運用に関する過去の知識や経験に基づいてオペレータが手動で行っている。そのため、ENCORE エージェントの数が大きい場合や ENCORE エージェントの状態やネットワークの状態が頻繁に変動する場合にはエージェント群の選択と role の割り当てを適切に行うことが困難であり、ENCORE の scalability や柔軟性が阻害されていた。この問題を解決するためには、ENCORE エージェントやネットワークの状態を自律的に収集して、それに基づいて各々のエージェントに role を自動的に割り当てる仕組みが有効である。

3.2. role 決定のための要求条件

本節では、各 role の内容を分析して、エージェントにそれぞれの role を割り当てるための要求条件を整理する。

3.2.1. investigation

investigation エージェントは、いろいろなエージェントから経路情報の問い合わせに答えるために、多くの経路情報を観察できることが求められる。多くの経路情報が交換されているネットワーク上の場所の例として、いわゆるトランジット AS が挙げられ、そのため、トランジット AS に属する ENCORE エージェントは有効な investigation エージェントとして機能すると考えられる。そこで筆者らは、ENCORE エージェントの所属 AS がいくつのお他 AS と BGP ピア(自分と BGP を用いて経路情報を交換している他 AS)の関係を結んでいるかに注目した。具体的には、ENCORE エージェントの所属 AS の BGP ピアの数に一定数以上である場合に、そのエージェントは investigation エージェントに適すると判断することとした。

3.2.2. neighbor

ENCORE における neighbor は、契約などで相互に支援する関係にある「仲間」の意味とある場所からのネットワーク的な「近傍」の 2 つの意味で使われている。前者は相互監視および通知のために使われ、後者は診断の後期段階において障害の原因の絞り込みを行う際に有効である。ここで「仲間」の関係は環境依存であり、割り当てを自動化することは現時点では難しい。そのため、本稿の範囲では neighbor の role は後者の「近傍」の意味で用いるものとする。「近傍」を具体的に決定するために、AS の接続状況を表すネットワークトポロジを利用する。具体的には、自分から見て指定ホップ数内に存在する AS の集合を「近傍」と定義することとした。

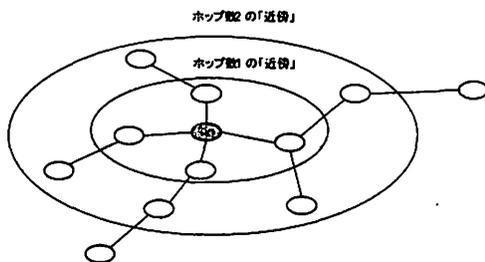


図 3-1 「近傍」の概念図

BGP でやりとりされる経路情報には、さまざまな情報が path 属性として付加されており、その中の一つに AS-path 情報がある。AS-path 情報とは、経路情報が自 AS に伝搬されてくるまでに経由した AS のリストである。AS-path 情報は伝搬された経路毎に存在するので、それらのすべてを取得して合成することで AS の接続状況を作成することができる。

3.2.3. relay

relay に適したエージェントの選び方はネットワークの状況やタスクの内容によっていくつかの方法が考えられる。例えば、図 2-1 で示した状況においては AS_{self} から経路情報が AS_2 に到達していないので、そのままでは R_{self} と R_2 は情報交換できない。しかし、 R_{self} と R_1 、 R_1

と R_2 は通信が可能であるので R_1 を relay エージェントとすれば R_{self} と R_2 の間の通信が可能になる。また、図を見てわかるとおり、この場合は R_4 を relay にしてもよい。この他にも、通常の経路として選択されている best path 以外の AS-path 上に存在する AS に配置する方法が考えられる。再度図 2-1 で説明すると、 AS_{self} から AS_4 へのトラフィックにおいて AS_1, AS_2 の route が best path であるとするならば A_3 が relay エージェントの候補になる。これらの例を見ても分かるとおり、relay エージェントの選択にも neighbor の選択と同様にネットワークトポロジの利用が有効である。ただし、複数の strategy のいずれを選択するかは診断の内容に応じて変化するので決定においては ENCORE との密な連携が必要である。

3.3. 情報の収集のための要求条件

次に、role を決めるための情報を効率的に収集、管理するためのアーキテクチャについて考察する。role の分析の結果から、アーキテクチャが満たすべき要件は次の 2 つであると考えた。

1. ENCORE エージェントと同じ視点でネットワーク情報を収集できること

investigation の場合を例に取ると、ある ENCORE エージェントを investigation に割り当てるかどうかの判断基準は ENCORE エージェントが所属する AS がいくつのお AS と BGP ピアの関係にあるかであった。これは当該 AS において最も正確に分かる情報である。伝搬される BGP 情報などを用いることで他の AS から当該 AS の接続状況を推測することは不可能ではないが、best path だけが伝搬される BGP の仕様などから他から観測できるのは当該 AS において観測できる情報の一部にとどまり、十分とはいえない。このことから、role の決定のためには、各 ENCORE エージェントが動作している環境から観測できる局所的な情報が重要であることが分かる。それらを効率的に収集するためにはネットワーク上で分散して情報収集する仕組みが有効である。

2. それぞれの観測地点で収集した情報を必要に応じて交換できること

診断の内容によっては、neighbor として自 AS の「近傍」に加え、他 AS の「近傍」の情報も必要になることがある。しかし、インターネットのような全体のトポロジが常に変化している動的な系においてはある AS が他 AS の「近傍」を知ることは容易ではない。インターネット全体のトポロジを管理している集中型のセンタのようなものを仮定すれば可能であるが、現実的ではない。これを解決するためには、1 で説明したようにネットワーク上で分散して情報収集することに加え、それらの観測地点間で観測した情報を必要に応じて交換できることが必要である。これにより、neighbor の例で言うと、自分以外の AS の「近傍」の情報が知りたい場合には、その AS に対して「近傍」情報を問い合わせるだけでよくなる。

これら 2 つの要求条件を満たすために、ARTISTE では、

ネットワークの各地点にエージェントを配置してそれらのエージェントが局所的な情報収集を行いエージェント間で適宜情報交換を行う ENCORE と同様の分散アーキテクチャを採用する。

4. エージェント管理システム ARTISTE

4.1. 概要

ARTISTE システムは以下の流れで ENCORE と連携して動作する。

1. ENCORE エージェントから協調するエージェント(群)を検索するための条件を ARTISTE に送信する。
2. ARTISTE は送信された条件に合致する ENCORE エージェント(群)を探査して、該エージェント(群)の情報、具体的には IP アドレスとポート番号を回答する。

ARTISTE のシステム構成を図 4-1 に示す。ARTISTE システムは以下の機能要素を持つ。

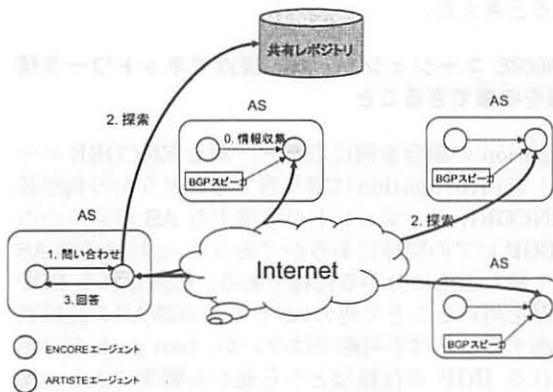


図 4-1 システム構成図

- ARTISTE エージェント
 - ENCORE エージェントやネットワークの状態を収集解析して role の割り当てを行う。ENCORE エージェントと 1 対 1 に対応して AS 毎に配置される。
- 共有レポジトリ
 - エージェント情報やネットワーク情報を集中管理する。集中型を採用した理由は、情報の取得には 3.3 で述べた分散アーキテクチャが有効であるが、情報の利用を考えると集中型の方がシステム構築の容易性や情報の検索性に優れると判断したためである。
- BGP スピーカ
 - BGP により他の AS と経路情報を交換するルータであり、ARTISTE エージェントがネットワーク状態を収集するために利用する。
- ENCORE エージェント
 - 複数の ENCORE エージェントと協調し、経路情報の監視診断を行なう。診断は ENCORE エージェントのみによって行われ、ARTISTE は実際の診断には関知しないことに注意する。

上記の ARTISTE エージェントは Java で記述し、共有レ

ポジトリは openLDAP を使って作成した。ARTISTE エージェント同士の通信には JXTA[6]を利用する。JXTA の discovery 機能を利用することにより、ARTISTE エージェント同士が role や所属 AS を指定するのみで他のエージェントと通信できる。

4.2. 情報の収集と管理

ARTISTE エージェントは、ENCORE エージェント、BGP スピーカなどから情報を収集し、かつ、それらが最新であるように維持管理する必要がある。

4.2.1. ネットワーク情報

ネットワーク情報の収集のために ARTISTE エージェントは自 AS 内の BGP スピーカと通信を行い、ネットワーク情報を収集する。

● AS-path 情報

BGP スピーカにログインして "show bgp paths" コマンドを実行すると図 4-2 に示すような実行結果が得られる。各行が個々の経路情報を示しており、Path カラムの値が、各経路が経由した AS のリスト、つまり AS-path を示している。例えば、1 行目の経路情報における AS-path は "x y z a b" である。リスト中の一番右の AS 番号が経路の発生元(origin)であり、一番左の AS が最終的な受信先、すなわち、自 AS を示している。各行の結果をすべて合成することにより自 AS(この例では AS "x")を中心としたネットワークトポロジを取得することが可能である。

Address	Hash	RefCount	Metric	Path
0x63209C80	0	8	0	x y z a b
0x621301E0	0	3	0	x y z c d e f
0x63F3C7B0	0	1	0	x y z g h

図 4-2 AS-path 情報

● BGP ピア情報

"show bgp peers" コマンドを実行した結果の一部を図 4-3 に示す。

```
BGP neighbor is a.b.c.d, remote AS x, internal link
Description: - hoge
Index 0, Offset 0, Mask 0x0
BGP version 4, remote router ID a.b.c.d
BGP state = Established, table version = 470808, up for 4d02h
```

図 4-3 BGP ピア情報

これは一つの BGP ピアについて出力される情報の一部であり、BGP ピアが複数ある場合はこの出力がピアの数だけ繰り返される。この出力を解析することで BGP ピアの数およびそれぞれの BGP ピアのアドレスや AS などが取得できる。

4.2.2. エージェント情報

個々の ENCORE エージェントに関する情報も定期的に収集管理する。先に述べた方法で所属 AS の BGP ピアの数取得できるので、その数によって investigation になれるかどうかを決定して状態を保持する。また、relay の role の決定に関しては診断の内容に依存する部分が多いため、今回の実装では、問い合わせ元の AS に直接接続されている AS の集合を relay の候補として返すことに留めた。より高度な relay エージェントの選択は今後の課題とした。また、管理対象となる ENCORE エージェン

トの IP アドレスやポート番号は、今回の実装では ARTISTE エージェント毎に用意する初期設定ファイル中に static に記述することとした。その他、ENCORE エージェントに定期的に polling を行ってその生存状況も管理した。

4.3. エージェントの系への加入と脱退

ENCORE や ARTISTE が動作するのはインターネットというオープンな系であり、ENCORE エージェントが自由に系に加入、脱退できることが望ましい。しかし、系の安全性を確保するために一定の認証手段は必要であり、ARTISTE では ENCORE エージェントの認証機能も提供している。ARTISTE では認証手段として ENCORE エージェント自身の IP アドレスによるアクセス認証を行っている。ENCORE が問い合わせを行ってきた際に通信相手の IP アドレスが取得できるので、各 ARTISTE エージェントはあらかじめ知っている同一 AS 内の ENCORE エージェントの IP アドレスと比較して、一致しているときのみ加入を許可して通信を行うこととした。アクセス認証を通過した ENCORE エージェントは系の加入（ログイン）、一時脱退（ログアウト）を行うことができる。これに加え、ARTISTE エージェントによる ENCORE エージェントの生存状況の監視結果を加えて、以下の3つの状態を系への加入状態として管理している。

in	ENCORE エージェントが系にログイン中であることを示す。
temp out	ENCORE エージェントが系からログアウトしていることを示す。
out	ENCORE エージェントが停止した、などの理由で通信できない状態にあることを示す。

「系」への加入管理は ARTISTE エージェント自身についても必要である。ARTISTE への加入を管理するために今回は JXTA の peer group の機能を利用した。具体的には、ARTISTE エージェントは動作を開始する際に「系」全体を表す JXTA の peer group への加入を行うこととし、その際にあらかじめ設定された ID、パスワードを用いた認証を行うように実装した。

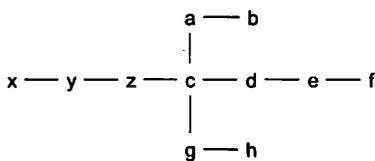


図 4-4 ネットワークトポロジの例

4.4. 「近傍」の作成

4.4.1. 自 AS の「近傍」

自 AS の「近傍」は先の AS-path 情報から計算する。図 4-2 の AS-path 情報から作成されるネットワークトポロジを図 4-4 に示す。これにより AS "x" のホップ数 1 の「近傍」は AS "y" でホップ数 2 の「近傍」は "z" であることが分かる。

4.4.2. 他 AS の「近傍」

ENCORE からの問い合わせ中に他 AS の「近傍」が含ま

れていた場合は次の手順で問い合わせを行う。「近傍」の問い合わせに含まれるパラメータは①近傍を求める対象の AS(任意のものを指定可能)、②近傍の範囲となるホップ数、の 2 つであるので、まず、JXTA を用いて、近傍を知りたい AS に属する ARTISTE エージェントを discover する。具体的には、各 ARTISTE エージェントは立ち上げ時に自分に対する通信 pipe を "pipe: "+所属 AS 番号という名前でネットワーク上に発行しているため、ある AS "X" に属する ARTISTE エージェントと通信したいエージェントは JXTA の discovery 機能を用いて "pipe: X" という名前の pipe を discover すればよい。ARTISTE エージェントが discover できた場合は、該エージェントに問い合わせメッセージを送る。個々のエージェントでは 4.4.1 に示した方法により、自己の「近傍」の情報を取得しているので、問い合わせ元にその情報を返すだけでよい。

4.4.3. 他 AS の「近傍」の推測

他 AS の「近傍」を知りたいときに当該 AS のエージェントと通信できる場合は問題ないが、エージェントと通信ができない、ネットワーク障害のために該 AS との通信が不能、などの理由から返答が得られない場合もある。そのような場合は、指定された AS の近傍候補を自分自身が管理する AS-path 情報から推測する。エージェントとの通信が不調である場合は、近辺でネットワーク障害が起こっている可能性もあり、本機能により、できる限り情報収集を継続することには意義がある。例えば、図 4-4 で示すトポロジからは AS "c" のホップ数 1 の「近傍」として "a", "d", "g", "z" の 4 つが候補として推測できる。

4.5. ENCORE との通信

ENCORE と ARTISTE との間で問い合わせを行うためのプロトコルについて説明する。プロトコル自身は simple な query-response 形式のプロトコルであり TCP 上に実装した。ENCORE は Lisp で実装されているので、ENCORE との親和性を考慮して、通信されるメッセージの形式は S 式風の text 形式とした。

4.5.1. メッセージフォーマット

● ENCORE→ARTISTE

ENCORE から ARTISTE に送られる問い合わせの一般形は次の形式で表す。

```

(query
  (and|or
    (investigation|relay|tier1 t|nil)*
    (neighbor #1 #2)*
    (asnumber #1)))
  
```

このフォーマットは対象となる ENCORE エージェントの持つ属性に関する条件節を任意の数だけ and/or により論理結合したものとなっている。各条件節の内容は次の通りである。

investigation	investigation 属性が指定した値(t nil)を持つエージェントを検索する。
relay	relay 属性が指定した値(t nil)を持つエージェントを検索する。

asnumber	指定された AS に属するエージェントを検索する。
neighbor	最初のパラメータは AS 番号、2 つ目はホップ数を指定する。指定した AS のホップ数内の近傍にいる ENCORE エージェントを検索する。

● ARTISTE→ENCORE

条件に合致する ENCORE エージェントの情報が得られたときには (agents "address1:port1" "address2:port2" ...) というフォーマットで返答を返す。条件に合致するエージェントが見つからなかった場合は (agents nil) というメッセージを返す。

4.5.2. 通信プロトコル

ENCORE と ARTISTE 間の通信プロトコルの概略を以下に示す。ENCORE エージェントを "E", ARTISTE エージェントを "A" と表記する。

(1) (E→A) 指定されたアドレスとポートに対して socket 通信を用いて接続を開始する。接続が完了したら (attach) メッセージを送信して系へのログインを申請する。

(2) (A→E) socket から送信元(ここでは E)の IP アドレスを取得して、自分の内部情報にある接続可能 IP アドレスの一覧に含まれているかどうかチェックする。含まれていれば認証済み ENCORE エージェントとして当該アドレスを登録する。認証が成功したときは (attach succeeded) というメッセージを返信する。

(3) (E→A) 認証が成功した場合、ENCORE エージェントは自分の知りたい ENCORE エージェントの検索条件を記述した問い合わせメッセージを送信する。

(4) (A→E) ARTISTE エージェントはメッセージの内容を解析した後、共有レポジトリや他の ARTISTE エージェントと通信を行いながら、条件に合致する ENCORE エージェントを探索して、リストを作成し、問い合わせ元の ENCORE エージェントに返答を行う。系に加入している間はこの問い合わせと回答は何度でも繰り返すことができる。

(5) (E→A) 問い合わせが完了して通信を終了するときは、(detach)メッセージを送って系からログアウトする。

(6) (A→E) (detach)メッセージを受信したら、内部の Hashtable の当該エントリの値を "in" から "temp out" に更新する。更新が成功したら (detach succeeded) というメッセージを返す。

(7) (E→A) ログアウトが成功したら、(quit)メッセージを送って通信を終了する。(quit)メッセージを受信したら ARTISTE エージェントは socket を close する。

4.6. 基本機能の評価

作成した ARTISTE のプロトタイプを利用して基本機能の評価を行った。実験用の環境は、筆者らが運用する実験用ネットワーク中に実験用のサブネットを構築し、サブネット中に BGP スピーカを用意して実験ネット中のルータから BGP により経路情報 (およそ 120,000 経路) を feed することで構築した。共有レポジトリにはこの経路情報を元に事前に作成した約 100 の ENCORE エー

ジェントの情報を保存した。この環境上で FreeBSD 4.10R 上で動作させた ARTISTE (エージェント数 4) に対し ENCORE から問い合わせを送信し、エージェントが協調して情報の収集・解析を行えることと回答された結果の正当性を検証した。その結果、回答されたエージェント群は人間がネットワークの状態から判断した場合と一致しており、ARTISTE の有効性が確認された。

5. まとめ

本稿では、AS 間経路障害診断システム ENCORE の scalability の向上および柔軟な診断を実現するために、ENCORE エージェントやネットワークの状態に関する情報を自律的に収集してエージェントに適切な役割を自動的に割り当てるエージェント管理システム ARTISTE について述べた。今後は、ARTISTE の scalability に対する性能評価やアーキテクチャを一般化して適用領域の拡大を図っていく予定である。

参考文献

- [1] Y. Rekhter and T.Li " A Border Gateway Protocol 4 (BGP-4)", RFC1771, 1995.
- [2] O. Akashi, T. Sugawara, K. Murakami, M. Maruyama and K. Koyanagi, "Agent System for Inter-AS Routing Error Diagnosis", IEEE Internet Computing, Vol.6, No.3 pp.78-82, 2002.
- [3] O. Akashi, T. Sugawara, K. Murakami, M. Maruyama, N. Takahashi, "Multiagent-based Cooperative Inter-AS Diagnosis in ENCORE", In Proceeding of NOMS2000, pp.521-534
- [4] <http://www.jxta.org/>