

## 解説

# アクティブネットワーク ：ネットワークを オーダメードに使う

小花 貞夫・杉山 敬三

(株) KDD 研究所

## アクティブネットワークとは何？

### ●直感的な理解

「ネットで株取引 中断相次ぐ！」

インターネットを使ったオンライン取引が普及するにつれ、こんな記事が新聞をにぎわすようになってきました。たとえば、買い注文を出した後に株価が下がり出したためキャンセルしようとしても、売り買い注文が殺到したためにコンピュータにつながらず損をしたという実例があります。このような事態を避けるためにコンピュータの処理能力やネットワークを大容量にするだけでは、ユーザ数の増加とのいたちごっこになりかねません。もっと根本的な対策が必要となります。

WWW (World Wide Web) を用いて株価情報を提供している場合、株価の値動きが激しい状態ではサーバの負荷も高まりますが、そのようなときこそユーザは最新の株価情報を得ようとしてサーバへのアクセスが集中します。WWWでは、ネットワーク上を転送されるデータ量を削減したり応答時間を改善するために、ユーザがアクセスしたページをキャッシュするプロキシが一般に用いられます。しかしながら、株価は動的に変動し、個々のユーザは特定の銘柄にしか注目していないため、キャッシュにヒットする確率は小さくなります。そこで、プロキシではページ単位でなく銘柄単位でキャッシュするとともに、ユーザの希望する頻度でキャッシュが更新されるようサーバからプロキシに情報が配信されれば、キャッシュのヒット率が高まり全体のやりとりを削減できます。

また、オークションのように現在の売値を見たユーザ

が買い注文を出す場合、多くの買い注文がきても最高値だけに意味があり、それより低い値段の買い注文はサーバに届いても無駄になってしまいます。そこで、途中のノードに最新の最高値を通知し、その配下のユーザから最高値より低い値の買い注文を受信した場合には、途中のノードはサーバに転送せずに否定応答を返すことで、サーバへのアクセスを大幅に削減できます。

これらを実現するためには、ネットワーク内のノードが株取引やオークションなどに特化したプロトコル処理を行う必要があります。従来のネットワークでは、ルータやスイッチ等のネットワーク内のノードは、基本的にパケットを宛先アドレスのホストに向けて転送するだけでした。しかしながら、通過するノードにプログラムを投入してパケットごとに異なる処理を行わせることができれば、個々のアプリケーションやユーザに特別あつらえしたネットワークサービスを提供できます。このようにネットワークをプログラムする技術は、ネットワークのユーザが能動的にネットワークを制御できるという意味で「アクティブネットワーク」と呼ばれ、最近注目を集めています。これに対し、従来のネットワークをここでは「パッシブネットワーク」と呼ぶことにします。

### ●生まれた背景

パソコンに代表される最近の端末の性能向上には、目覚ましいものがあります。また、WWW等のネットワークアプリケーションの普及により、ネットワーク・コンピューティングも日常的に行われるようになってきました。しかしながら、そのインフラストラクチャとなるべきネットワークの技術革新は、端末等の進歩に比べ非常に緩やかです。たとえば、インターネットで用いられているTCP (Transmission Control Protocol) /IP (Internet Proto-

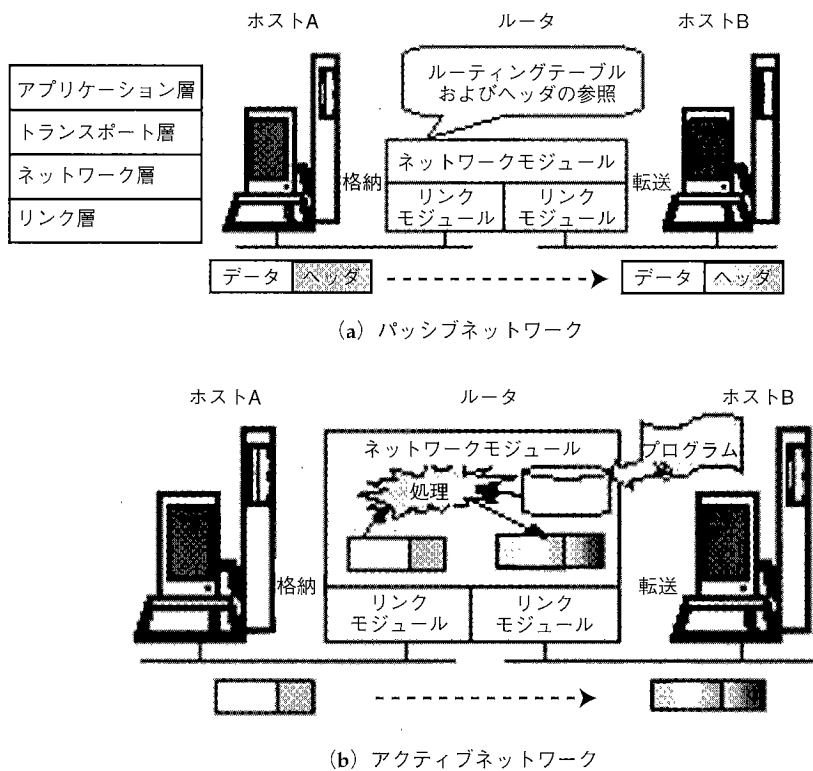


図-1 パッシブネットワークとアクティブネットワーク

col) プロトコルは、米国防総省の研究・開発部門である DARPA ( Defense Advanced Research Projects Agency) によって今から約30年前に開発されたものですが、1990年代に入ってようやく広く使われるようになりました。

このようにネットワークの技術革新が遅れる一因として、相互接続性の確保が挙げられます。ネットワークが正しく機能するには、ルータ等のネットワーク内のノード同士が相互接続できなければなりません。これまで、IETF ( Internet Engineering Task Force) や ISO (International Organization for Standardization) 等の標準化機関がプロトコルを規定し、各ベンダがこのプロトコルに従った実装を行うことで相互接続性を確保してきました。しかしながら、標準化作業にはベンダの利害が絡むなどのため多大な労力や時間を要し、さらにベンダによる標準の実装や顧客による要求を待ってから設備を導入することになるため、新しい標準が実際に使用されるまでにはかなり時間がかかります。

また、既存の標準の大幅な変更や従来のモデルと異なる新しい標準などの場合には、相互接続性の検証のために、インフラストラクチャである既存のネットワークを使用することができません。たとえば次世代のIPプロトコルのバージョンであるIPv6では、アドレスフィールドを現行のバージョンであるIPv4の32ビットから128ビット

に拡張し、実時間性やアドレス自動設定のサポート、セキュリティ機能の強化といった機能拡張がなされ大幅な変更が行われています。このため、既存のIPv4ネットワークとは別に6boneと呼ばれる実験的な世界規模のIPv6ネットワークを構築し、相互接続性を検証するといった努力がなされています。

そこでDARPAでは、これらの問題点を解決するために、1994年頃アクティブネットワークの概念を提案しました。現在DARPAがスポンサとなり、大学等の研究機関を中心に約40のプロジェクトが進行しています<sup>1)</sup>。また、Java等のプログラミング言語や移動エージェント技術の出現、コンピュータの高速化により、プログラムをネットワーク上で安全に移動させ、効率的に実行できるようになったのも、アクティブネットワークの研究を後押ししています。

### ●従来のネットワークとの違い

先ほど直感的にパッシブネットワークとアクティブネットワークを述べましたが、その差異を明確化するために各々の定義を行うことにします<sup>2)</sup>。ここで、ネットワーク層の処理とはパケットを転送する処理であり、またアプリケーション層の処理とは特定のアプリケーションに固有の処理であるとしします。

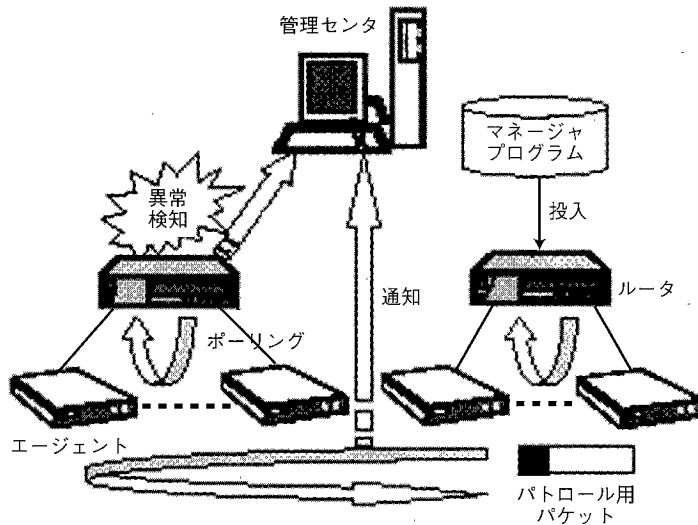


図-2 ネットワーク管理へのアクティブネットワーク技術の適用例

#### • パッシブネットワーク

パッシブネットワークは、アプリケーション層まで処理能力を持つネットワークの端に位置するホストと、そのホストを相互接続しネットワーク層までの処理しか行わないルータから構成されるネットワークです。

#### • アクティブネットワーク

アクティブネットワークは、中間のルータがアプリケーション層までの処理を行うネットワークであり、ユーザやネットワークの運用者がその処理のプログラムをルータに動的に投入することで、ルータの振る舞いや状態を変更できます。

パッシブネットワークの場合、図-1 (a) に示すように中間のルータはパケットをいったん格納して宛先アドレス等パケットのヘッダを解析し、経路情報を示すルーティングテーブルに従って宛先ホストに向けてパケットを転送します。ルータは、転送されるデータのフィールドであるパケットのユーザデータに対し処理を行いません。

一方アクティブネットワークの場合、中間のルータはパッシブネットワークのようにパケットを転送するだけではなく、図-1 (b) のようにパケットのユーザデータに対しユーザやアプリケーションの要求にカスタマイズした処理を行うことができます。ユーザやネットワークの運用者は、ルータに対しそれらの処理を行うプログラムを投入することで、まさに『ネットワークをプログラムする』ことができます。場合によっては、ネットワーク内のルータとユーザの端末に同じ処理を行わせることができるため、ホストとルータは機能的に差異がないともいえます。

## アクティブネットワークになると何がうれしいの？

#### ● メリット

これまでにアクティブネットワークの概念や生まれてきた背景などについて述べましたが、これが普及するとどのようなメリットがあるか、異なる立場から考えてみます。

ネットワークを利用するエンドユーザは、通信事業者が提供する標準化されたサービスを、自社の要求等に合わせてカスタマイズして利用できます。つまり、自社特有のサービスなど多様なサービスを、効率的かつ迅速に利用できるようになります。ネットワークにプログラムを投入する場合、エンドユーザがプログラムすることもできますし、サードベンダから提供される種々のアクティブネットワーク用のプログラムパッケージをユーザが選択し、自分の目的に応じてネットワークをカスタマイズすることもできます。

一方ネットワークを保有する通信事業者は、上記のような付加価値サービスに対して課金できます。また、これまでは新しいネットワーク技術やサービスが出現すると、常に標準化動向や製品における標準のサポート状況を把握する必要がありました。アクティブネットワークであれば、標準化動向に左右されずにルータやスイッチ等の導入を行うことができます。つまり、新しいサービスやアプリケーションが出現しても、それに対応するプログラムを動的に投入すればよいのです。ただし、投入し

たプログラムが各ノードで正しく動作するためには、アクティブネットワークの技術自身は標準化される必要があるでしょう。

### ●アプリケーションの例

アクティブネットワークは具体的にどのような場面で有効なのか、冒頭の株取引やオークションの他にアプリケーションをいくつか取り上げることにします<sup>2), 3)</sup>。

#### • ネットワーク管理

ネットワーク管理では、管理するホストであるマネージャが、管理される対象であるエージェントを定期的にポーリングし、異常かどうかを調べることがよく行われます。しかし、エージェントの数が増えたりエージェントの保持する管理情報が複雑化すれば、ポーリングにより授受されるデータ量も多くなり、ネットワーク管理のためにネットワークが輻輳することになりかねません。そこで、図-2に示すように中間のルータにマネージャのプログラムを投入し、重要な変化があるときのみセンタ側に通知させることで、データ量を削減できます。あるいは、監視機能を持たせたパケットにネットワーク内をパトロールさせ、随時障害を探索させることもできます。

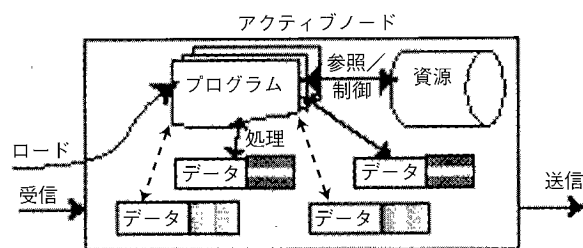
#### • QoS 制御

最近IETFにおいて、diffserv (Differentiated Services) と呼ばれる標準化が行われています。これは、さまざまなアプリケーションの要求するQoS (サービス品質) を満足するために、ネットワークが複数のQoSを柔軟に提供するメカニズムです。ここでは、IPv4やIPv6パケットのヘッダにおける特定のフィールドをdiffservのためのフィールド (DSフィールド) とみなし、DSフィールドのビット列に対しどのようにパケットを扱うかを各ルータで取り決めることで、パケットごとにQoSを扱います。しかしながら、DSフィールドの解釈およびそれに対するルータの振る舞いは、変更や追加が予想されます。もしアクティブネットワーク技術が利用できるなら、このような変更や追加に対し、ヘッダの解釈変更やビット列に対応した振る舞いを記述したプログラムを各ルータに投入することで、容易にサポート可能となります。

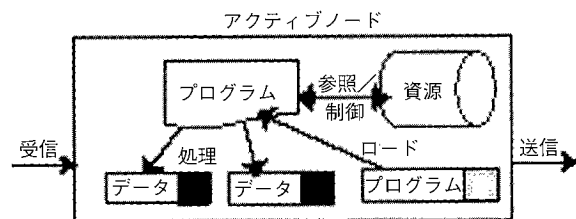
## 実現のためには何が技術課題なの？

### ●アクティブネットワーク実現の2つのアプローチ

アクティブネットワークを実現するためのアプローチには、大別して①プログラマブル・スイッチ方式、②カプセル方式、があります (図-3)。ただし、これらは独立したものではなく、必要に応じて両者を組み合わせたアプローチも存在します。



(a) プログラマブル・スイッチ方式



(b) カプセル方式

図-3 アクティブネットワーク実現のアプローチ

#### ①プログラマブル・スイッチ方式

プログラマブル・スイッチ方式は、パケットに対しルータが行う処理と、その行うべき処理のプログラムをルータへ投入するダウンロード処理を分離して扱います。アクティブノード方式とも呼ばれます。

#### ②カプセル方式

カプセル方式は、転送されるパケット内にユーザデータとともにカプセル化されたミニプログラムを挿入し、それがルータで実行されます。つまり、ルータがカプセルの実行エンジンになります。アクティブパケット方式とも呼ばれます。

### ●技術課題

アクティブネットワークの実現には、一般に以下のような技術課題が存在しています。これらの中には、一方が向上すれば他方が悪化するような相反する条件があります。たとえば、セキュリティレベルを上げると処理性能の悪化につながったり、資源のモデル化では柔軟性を重視すると複雑になるなどのトレードオフが存在します。

#### (1) セキュリティと安全性

悪意を持ったプログラムやパケットによる攻撃から保護するセキュリティや、プログラムのバグ等により誤った処理を行わないようにする安全性に加え、特定の処理が資源を独占しないようにする公平性に対処する必要があります。

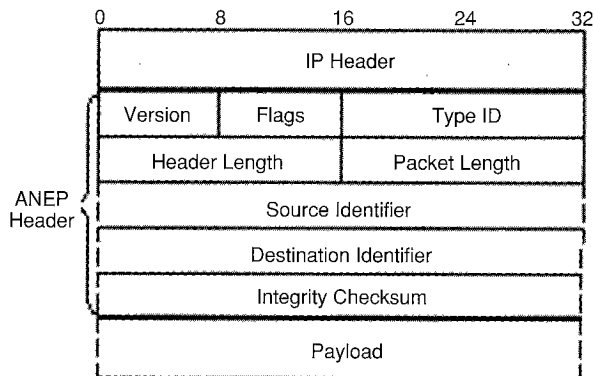


図-4 ANEPパケットフォーマット

```

fun ping (src:host, dst:host) : unit =
  if (not (thisHostIs (dst))) then
    onRemote (ping)(src, dst), dst, getRB (), defaultRoute)
  else
    onRemote (|ack|(src, dst), dst, getRB (), defaultRoute)
fun ack () : unit = print("Success")

```

図-5 PLANによるpingの記述例

ります。

(2) 処理性能

ルータとして動作するからには、当然パケットを高速に中継できなければなりません。従来のルータはパケットを転送することに焦点が当てられましたが、さらに投入されたプログラムの効率的な符号化方法やそれを扱うノードの処理性能が課題となります。

(3) 資源のモデル化

ストレージやメモリのような物理的な資源や、ルーティングテーブルやノードの各種管理情報のような論理資源を種々のプログラムから扱えるようにするため、これら資源の汎用的なモデル化を行い、API (Application Program Interface) として提供する必要があります。

## これまでどんな研究がされているの？

以下に、アクティブネットワークに関してこれまで行われている代表的な研究を紹介します。

### ● ANTS-プログラマブル・スイッチ方式

ANTS (Active Node Transport System)<sup>4)</sup> はMIT (マサチューセッツ工科大) のプロジェクトであり、中間ノードに任意の新しいプロトコルを実装するためのツールキットを開発しています。ANTSでは、そのためにカプセル、アクティブノード、コード配送機能を提供します。

カプセルはアクティブノード内の特定のプログラムを呼び出すための識別子を運び、カプセルが到着したアクティブノードでは識別子で指定されたプログラムが実行されることで、プログラマブル・スイッチ方式を実現します。アクティブノードに指定されたプログラムが存在しない場合、コード配送メカニズムによりプログラムは自動的に配送されます。アクティブノードはプログラムに

対し、ノードの状態やルーティングテーブルのような環境へのアクセス、カプセルのヘッダおよびユーザデータの操作、カプセルのコピーや転送およびノードへの格納といったカプセル処理ルーチンのためのAPIを提供します。

ANTSではプログラムの開発を容易にするためJavaをプログラミング言語として使用しており、セキュリティや安全性はある程度Javaに依存することができます。また、カプセルに含まれる識別子は、MD5 (Message Digest 5) により改ざんの危険を防いでいます。さらに、資源の利用はIPパケットのTTL (Time To Live) と同様の方法で制限されます。つまり、資源が消費されるたびにTTLの値を減らし、0になったらカプセルは廃棄されます。

### ● SmartPackets -カプセル方式

SmartPackets<sup>5)</sup> はBBN Technologiesのプロジェクトであり、スマートパケットというプログラムを含んだカプセルを転送することでカプセル方式を実現します。

スマートパケットに含まれるプログラムは、それ単独で処理が完了しなければならず、かつ1つのスマートパケット内 (1Kバイト以内) に収まらなければならないという制約があります。そのため、ネットワーク管理のアプリケーションのために特化することで符号化されるプログラムサイズを小さくしたSprocketというC言語に似た言語とSpannerというアセンブリ言語を用意しています。

スマートパケットを実行するルータでは、認証されたユーザからのプログラムであることをチェックしたり資源の使用に制約を設けたり、エラーが生じた場合には実行を終了して送信元にエラーパケットを返送することで、セキュリティや安全性を確保しています。スマートパケットは、図-4に示すANEP (Active Network Encapsulation Protocol) パケットのPayloadフィールドに含まれます。スマートパケットには、プログラム・パケット、データ・パケット、エラー・パケット、メッセージ・パケットの4種類があります。

## ● SwitchWare-プログラマブル・スイッチ方式とカプセル方式の組合せ

SwichWare<sup>6)</sup> はペンシルバニア大とベルコアのプロジェクトであり、セキュリティや性能、柔軟性等のトレードオフを解決するために、アクティブパケット・スイッチレット・アクティブルータからなる階層的なアーキテクチャを提供しています。

SwichWareでは、PLAN (Packet Language for Active Network) という言語を用いたプログラムをアクティブパケットにより運ぶことで、カプセル方式を実現します。PLANは機能的に制限された簡易な関数型の言語であり、セキュリティや安全性のため資源の使用に制約を設けたり実行の終了を保証できるようになっています。図-5に、インターネットにおけるping機能をPLANにより記述した例を示します。このプログラムを含んだパケットが到着したノードは、自身が宛先ホストでなければonRemoteプリミティブにより宛先ホストに向けてこのパケットを転送し、宛先ホストであれば送信元ホストにACKパケットを送ります。ここで、getRBは残りのホップ数を取得するためのAPIであり、TTLと同様の役割を果たします。

PLANの機能制限を補うため、必要に応じてスイッチレットという任意のプロトコルや機能が実装されたルーチンと呼び出すことで、プログラマブル・スイッチ方式を実現します。スイッチレットはネットワーク上で動的にロードでき、Camlという型チェックの厳しい関数型言語により記述されることで安全に実行されます。また、アクティブルータはアクティブパケットやスイッチレットが構築されるベースであり、セキュリティチェックのためにSANE (Secure Active Network Environment) という環境を利用します。

## ●その他

アクティブネットワークの実行環境として、プログラマブル・スイッチ方式ではワシントン大がDAN (Distributed code caching for Active Networks)<sup>7)</sup> を、カプセル方式ではMITが既存のIPヘッダのオプションフィールドでTCL言語を運ぶ方法を、また両方式の組合せとしてコロンビア大がNetScript<sup>8)</sup> という通信のためのデータフロー型言語を用いた環境を開発しています。また、アプリケーションとして、ネットワーク管理分野ではCMU (カーネギーメロン大) のDarwinプロジェクト<sup>9)</sup> や上記のNetScriptにより、ネットワーク管理プログラムをエージェントに送り込むDelegationが行われています。その他に、ジョージア工科大による輻輳制御やキャッシュノードの最適配置<sup>10)</sup> などの研究が行われています。さらにアクティブノードの処理を高速化するために、ハードウェア化も検討されています<sup>11)</sup>。これら多くの研究プロジ

ェクトはAbone (Active Network Backbone)<sup>12)</sup> という実験用ネットワークで相互接続されており、ANEPを共通プロトコルとして使用しています。

また、国内でもストリームコードというカプセル用プログラミング言語<sup>13)</sup>、エニーキャストのためのサーバ負荷分散方式<sup>14)</sup>、ネットワーク管理への適用方法<sup>15)</sup> などの研究が行われています。

## 今後どうなるの？

インターネットの今日の発展は、ネットワークの機能はできるだけ単純化 (best-effort) して高速・大容量化を図り、エンドノードに各種の機能を持たせることで実現されたともいえるでしょう。したがって、ネットワークにアクティブな機能を持たせることには、当然反対意見があります。このような議論は、エンド・ツー・エンド論争 (End-to-end Argument) と呼ばれています。すべてのネットワークがアクティブネットワークに置き換わる必要はありませんが、ここで挙げたいいくつかのアプリケーションを効率的にサポートするアプローチとして、アクティブネットワークは有力なものであることは確かです。ただし、今のところすべて実験段階の技術であり、インターネットのように世界的に広がるネットワークに対し、実用的な性能でサービスを提供できるかどうかは疑問などもあります。現在Abone上で各種の実験が行われており、今後世界的な規模でその実用性や適用領域を探っていく必要があるでしょう。

### 参考文献

- 1) <http://www.darpa.mil/ito/research/anets/projects.html>
- 2) Psounis, K.: Active Networks: Applications, Security, Safety and Architectures, IEEE COMSOC SURVEYS, 1stQuarter (1999).
- 3) Tennenhouse, D.L. et al.: A Survey of Active Network Research, IEEE Communications Magazine (Jan. 1997).
- 4) Wetherall, D.J. et al.: ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols, Proc.IEEE OPENARCH (Apr. 1998).
- 5) <http://www.net-tech.bbn.com/smtpkts/smtpkts-index.html>
- 6) Gunter, C.A. et al.: The SwitchWare Active Network Architecture, IEEE Network, Vol.12, No.3 (1998).
- 7) Decasper, D. et al.: DAN: Distributed Code Caching for Active Networks, Proc.IEEE INFOCOM (1998).
- 8) Yemini, Y. et al.: Towards Programmable Networks, Proc.IFIP/IEEE DSOM (1996).
- 9) Chandra, P. et al.: Darwin: Customizable Resource Management for Value-Added Network Services, Proc.IEEE ICNP (1998).
- 10) Bhattacharjee, S. et al.: Self-Organizing Wide-Area Network Caches, Proc.IEEE INFOCOM (1998).
- 11) Decasper, D.S. et al.: A Scalable High-Performance Active Network Node, IEEE Network (Jan/Feb 1999).
- 12) <http://www.csl.sri.com/ancors/abone/>
- 13) 江川他: ストリームコードを用いたアクティブネットワーク, 信学技報, IN98-66 (1998).
- 14) 三浦他: Active Network技術を用いたサーバ負荷分散方式, 信学TMワークショップ99-9 (1999).
- 15) 吉原他: アクティブネットワーク技術の網管理への適用効果に関する検討, 第58回情報処全大 (1999).

(平成11年5月10日受付)