

動的ネットワークにおける FN ゲートウェイエージェントの実装

6L-05

渡邊将一郎[†], 北形元^{††}, 菅沼拓夫^{††}, 木下哲男^{†††}, 白鳥則郎^{††}[†] 東北大学工学部情報学科/電気通信研究所^{††} 東北大学電気通信研究所/情報科学研究科^{†††} 東北大学情報シナジーセンター/情報科学研究科

1. はじめに

近年, ネットワーク技術・接続形態の多様化 (ISDN, ADSL, 無線 LAN, ATM 等) やアプリケーションの多様化に伴い, その組み合わせは複雑化しており, その結果利用者からの様々な要求を満足するために, ネットワークとアプリケーションを効果的に運用する枠組みが求められている. そこで我々は, 利用者要求を最大限満足するサービスを提供することを目標に, 次世代広域ネットワークアーキテクチャとして動的ネットワークを提案している. 本稿では動的ネットワークアーキテクチャにおける流量制御モデルに焦点を当て, アプリケーションフローを調整する FN ゲートウェイエージェントの実装について議論する.

2. 動的ネットワークの概要

従来のアプリケーションの多くは TCP/UDP などが提供する通信サービスを静的/固定的に用いて構成されてきた. しかしながら, 時々刻々と変化する利用者要求やネットワーク環境の中で, 従来の固定的なアーキテクチャでは様々な弊害が生じているのが現状である. そこで我々はソフトウェア工学的, あるいは利用者指向的な観点を考慮した新たなネットワークアーキテクチャとして動的ネットワークアーキテクチャを提案している.

動的ネットワークアーキテクチャでは, 現行の IP ネットワークのサービス機能層 (論理ネットワーク層: LN 層) とアプリケーション層 (AP 層) の中間に位置する機能層として, 新たに「やわらかいネットワーク層 (FN 層)」を導入している. FN 層は, AP 層からの要求と LN 層のサービスを効果的に連結し, AP 層と LN 層の双方における大・小/一時的・長期的変動を自律的に調整する機能を持つ. また FN 層は, 下位層のサービスを高度化して上位層に提供するなどの一般的な層機能のほかに, 上/下位層の状況を監視しつつ, それらに対し積極的に働きかけを行う, アクティブなミドルウェアとしての性質を有する.

3. 流量制御システム

3.1 現状の流量制御システム

ネットワークサービス品質 (QoS) 保証方式として IntServ や DiffServ が挙げられるが, 経路上の IP ルータへのシグナリングプロトコルの実装や, アプリケーションへの DS フィールドを扱うための実装などの, 特別な実装が必要となる. その結果, 利用できるネットワークやアプリケーションが制限され

てしまうという欠点があった. そこで我々は流量制御機構を LN 層レベルで導入するのではなく, 従来の AP 層にあたる FN 層に導入することで, 従来のネットワークを変更なしに QoS 制御を可能にした. 具体的には図 1 に示すように, AP 層と LN 層間に流れるトラフィックを, 流量調整機能がアプリケーション別にキューイングし, それぞれのキューからデータを取り出す頻度を優先度に応じ調整する.

また優先度つまり QoS パラメータを, ネットワークや使用しているアプリケーションの状況に応じて適切に設定するためには, 利用者に QoS に関する知識が必要となる. そこで我々はアプリケーション監視機能と QoS 解決機能を用いてアプリケーション毎の QoS パラメータを適応的に生成し, 利用者に特に QoS に関する知識がなくとも, QoS 要求をシステムに伝達できるよう設計を行った.

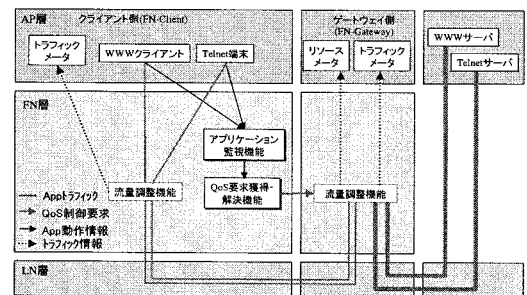


図1. 動的ネットワークにおける流量調整システムの機能構成

3.2 現状のシステムの問題点

3.1 で述べた流量制御システムのプロトタイプによる実験を通じ, 以下の問題点が明らかとなった.

- (P1) スループットを上げることが困難
- (P2) 新規アプリケーション対応時のコスト (手間) が大きい
- (P3) プロキシに非対応なアプリケーションへの適用が困難

上述の問題のうち, (P1), (P2) は性能の問題であるが, (P3) は流量制御に限らず AP 層と FN 層の接続に関して解決しなければならない重要な問題である. よって本稿では (P3) の問題点に着目し, これを解決する手法を提案する.

3.3 問題分析

本稿の着目する問題点 (P3) について詳細を述べる. 現状の FN 層流量調整システムの実装では, 各々のアプリケーションフローをプロキシに入力し, プロキシからの出力を制御することで流量を制御している.

この手法は、Web ブラウザ等をはじめからプロキシを用いることを前提としているアプリケーションの場合には問題ないが、プロキシを通すことを前提としていないアプリケーションの場合、そのアプリケーションの接続先を FN 層内のプロキシに向けた場合、本来の送信先アドレスを指定できないという問題がある。これは図2に示すように、proxy に接続してしまうと、本来の接続先アドレスをシステム側、すなわち図2中の proxy へ伝達する方法が無くなってしまう事に起因している。

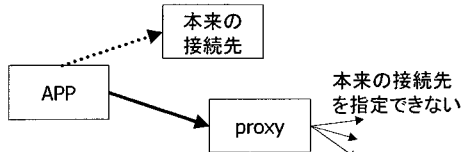


図2. Proxy非対応アプリケーションに関する問題

4. FN ゲートウェイエージェントの実装

上述の問題点(P3)を解決するために、transparent proxy 技術を活用した FN ゲートウェイエージェントの実装を行う。transparent proxy 技術とは Linux の ipchains コマンドや FreeBSD の ipfw コマンドを用い、ゲートウェイを通過する IP パケットを、上位層で稼動する proxy プロセスのソケットへリダイレクトする技術である。この際、proxy プロセスは、リダイレクトを受け付けたソケットから、IP パケットの送信元アドレスと送信先アドレスを取得することが出来る。Proxy プロセスは、取得した送信先アドレスを元に本来の接続先にアクセスし、受信結果を送信元ホストへ返送する。すなわちクライアントアプリケーションから見た場合、プロキシを意識することなくサーバアプリケーションへ接続でき、このときのアプリケーションフローは、透過的に proxy プロセスを経由することになる。

具体例として WWW を例に説明を行う。ここで、web ブラウザは、プライベートアドレスを持つマシン上で稼動していると仮定する。通常、プロキシ経由で web サーバーにアクセスする場合、利用者は web ブラウザにプロキシのアドレスを指定した後に web サーバーにアクセスする。これに対し、transparent proxy を用いる場合、利用者は web ブラウザにプロキシの設定をせず、直接 web サーバーにアクセスを行う。ここで、伝送路途中に存在する transparent proxy が 80 番ポートへのフローを吸い上げ、web サーバーに対し代理アクセスを行う。この場合、利用者からは web サーバーへ直接接続しているように見える。FN ゲートウェイエージェントの実装に transparent proxy 技術を活用し、ゲートウェイにおいてアプリケーションフローを LN 層から FN 層に透過的に吸い上げ、プロキシに対応していないアプリケーションの流量調整を可能とする。

(Javaプログラム内)

```

//本来の接続先アドレスを取り出す
SERVER_ADDRESS = sock.getLocalAddress();
SERVER_PORT = 80;

//本来の接続先ホストに接続する
server_sock = new Socket(SERVER_ADDRESS, SERVER_PORT);

```

図3. transparent proxy技術の利用

図3に、FN ゲートウェイエージェントが OS からのリダイレクトを受け取り、本来の接続先のアドレスを取得する部分の、Java によるコード例を示す。図3の sock は、OS からのリダイレクトを accept したソケットであり、クライアントのプロセスと接続されている。従って sock に対して getLocalAddress() メソッドを呼び出すことで、クライアントの対向のアドレス、すなわち本来の接続先のアドレスを取得できる。

図4に、FN ゲートウェイエージェントを含む流量調整システムの構成を示す。アプリケーション App1~App3 からのフローを FN ゲートウェイエージェントが透過的に吸い上げ、流量調整 Ag へ経由させることで、プロキシに対応していないアプリケーションの流量を制御することが可能となる。

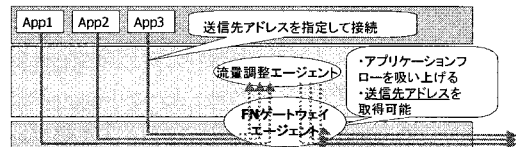


図4. FNゲートウェイエージェントを用いた流量調整システム

5. おわりに

本稿では動的ネットワーキングにおける現状の流量制御システムの問題点を示し、その中で、(P3)プロキシに非対応なアプリケーションに適用出来ないトイウ問題点を解決するため、transparent proxy 技術を用いた FN ゲートウェイの実装について述べた。

今後は、(P1),(P2)それぞれの改良案を具体的に設計・実装し、その有効性を実験により検証する。

【参考文献】

- [1] T.Kinoshita, S.Sugiura, T.Suganuma, K.Sugawara, and N.Shiratori: Evolutional Flexible Network, IPSJ Technical Report, 143-148, 1999.
- [2] T.Suganuma, T.Kinoshita, and N.Shiratori: Flexible network layer in dynamic networking architecture. Proc of the 1st International Workshop on Flexible Networking and Cooperative Distributed Agents (FNCDA2000)473-478, 2000.
- [3] 北形 元, 菅沼 拓夫, 木下哲男, 白鳥則郎: 動的ネットワーキングアーキテクチャにおける応用の一例: アプリケーション指向フロー制御システムの提案, 情報処理学会第9回 DPSワークショップ論文集, pp.103-108, 2001.