高機能転送層を持つ広域ネットワークの包括的制御システムの開発

Development of the Comprehensive Control System of a Wide Area Network with High Efficiency Forwarding Layer

森島 直人[1)]　小川 晃通[2)]　垣内 正年[3)]
Naoto MORISHIMA　Akimichi OGAWA　Masatoshi KAKIUCHI

1) 奈良先端科学技術大学院大学 情報科学研究科 (〒630-0101 奈良県生駒市高山町 8916-5 E-mail: naoto@kuma-project.net)

2) 慶應義塾大学大学院 政策・メディア研究科 (〒252-8520 神奈川県藤沢市遠藤 5322 E-mail: akimichi@kuma-project.net)

3) 奈良先端科学技術大学院大学 情報科学研究科 (〒630-0101 奈良県生駒市高山町 8916-5 E-mail: kakiuchi@kuma-project.net)

ABSTRACT. The service which a wide area network offers came to be asked for various added value with the stability and expansion of the Internet. However, with the present structure maintained which a packet forwarding mechanism and its control mechanism have closed within each node, it is difficult to extend only a control unit and many problems occur in extendibility. As a new concept which solves this problem, the separation structure of a packet forwarding mechanism and its control mechanism is considered. Standardization and deployment of a forwarding mechanism are advanced, but the comprehensive control mechanism of a forwarding mechanism is still absent. Therefor, we proposed the comprehensive control system of a wide area network. In this paper, we discuss the design and implementation of the KUMA's Unified Management Architecture (KUMA).

1 背景

インターネットの安定と拡大にともない、広域ネットワークが提供するサービスにさまざまな付加価値が求められるようになった。このような次世代ネットワークが提供すべきサービスは個々のネットワークノードによる独立分散処理では実現することができず、制御単位をグループ化された複数のノードへと拡張する必要がある。しかし、パケット転送機構とその制御機構が各ノード内で閉じている現在の構造を維持したまま、制御単位だけを拡張することは困難であり、拡張性に多くの問題が発生することは容易に予想できる。

この問題を解決する新しい概念として、パケット転送機構とその制御機構の分離構造が考えられている。この構造は、パケット転送機構を実現するノードの集合としてのグループと、その制御空間を包括的に扱う制御システムから構成される。この構造のもと、広域ネットワークにおける協調分散処理を実現することで、多様化するネットワークへの要求を実現することができると考えられる。現在、次世代ネットワークを担う、さまざまなパケット転送機構の標準化、および、設計開発と実ネットワークへの展開が進められている。しかしその一方、転送機構の包括的な制御機構は未だ不在のままである。そのため、転送機構の持つ柔軟性に富んだ機能を活用し、複雑化する要求に応えるサービスを構成することができない。

以上のような背景から、我々は広域ネットワークの包括的制御システム KUMA を開発した。

(1) これまでの研究開発

我々は、制御パラメータトランスポートに関する研究開発を 1999 年秋から行なってきた。

a) 2000 年春の実験

2000 年春には、Diffserv[1] EF (Expedited Forwarding)[2] を利用した動的資源予約実験を行なった [3, 4, 5]。2000 年春の実験は、WIDE Project[6] の合宿において行なわれた。合宿には 236 人が参加し、対外線には ATM over T1 が用いられた。この実験により、動的な品質保証サービスが提供された環境下におけるユーザのとり得る挙動を理解することができた。実験の結果、動的な SLA (Service Level Agreement) が提供される Diffserv ネットワークでは、ユーザは品質保証サービスを頻繁に利用することがわかった。また、ユーザは輻輳に対して敏感であり、輻輳時の通信品質を高く評価していることがわかった。

b) 2000 年秋の実験

2000 年秋には、衛星回線と T1 の対外線選択および優先制御を Diffserv AF (Assured Forwarding)[7] を利用してできるネットワークを構築し、動的資源予約実験を行なった [8, 9]。2000 年秋の実験は、WIDE Project の合宿において行なわれた。合宿には 252 人が参加し、対外線には ATM over T1 および衛星回線が利用された。実験では、特に、品質の異なるデータリンクを合宿参加者が直接選択できるシステムを提供し、参加者の挙動を観察した。これにより、動的な品質保証サービスが提供された環境下におけるユーザのとり得る挙動を理解することができた。実験の結果、回線自体の品質が異なる複数のリンクや経路を顧客に選択させてサービスを行う場合、選択できる属性や課

金の対象の設定を慎重に行わなければならないことが明らかになった。2000 年秋の実験では RTT が著しく異なるふたつの回線を選択の対象としたにも関わらず、予約および課金の対象を帯域としたため、多くの参加者が狭帯域を長時間にわたって予約するという現象が見られた。

c) 2001 年春の実験

2001 年春には、衛星回線と 128kbps 回線を 5 本トランキングした地上線の選択および優先制御を Diffserv EF を利用してできるネットワークを構築し、動的資源予約実験を行なった。また、2001 年春の実験では、単一の制御機構を利用して複数の転送機構を制御できた。2001 年春の実験は、WIDE Project の合宿において行なわれた。2001 年春の対外線の選択には、MPLS (MultiProtocol Label Switching)[10] および L3TE (Layer 3 Traffic Engineering) が利用された。合宿参加者は、対外線選択のための制御機構を選ぶ事ができ、動的に資源の予約を行なう事ができた。

2 目的

本実装である、KUMA は、1999 年秋から研究開発が行われてきた同名の制御パラメータトランスポートに対し、複雑なポリシや要求に対応できるパラメータ決定機構と、多様化する転送機構への対応モジュールを追加するものである。また、個々の機構の抽象化および細分化により、相互協調による柔軟性を追求する。本開発の目的は、このシステムの実現により、パケット転送機構が提供する多様な機能の利用を容易にし、次世代ネットワークサービスの提供を大きく前進させることである。

3 開発ソフトウェア

本章では、開発を行なったソフトウェアについて述べる。

(1) 開発ソフトウェアの概要

我々は、高機能化が進んでいる数多くの転送機構に対応し、ネットワークの機能としてさまざまなサービスを提供するために、汎用的・統合的な制御機構ソフトウェア KUMA の開発を行なった。

次世代ネットワーク制御機構では、多様化する要求やポリシを、多様化する転送機構に反映するという機能が必要である。また、これらの要求やポリシ、転送機構は時間の経過と共に変化していく。このため、制御機構を必要に応じて再構成し、それらの間隙をスムーズに結び付けることができなければならない。

提案システム KUMA の概略図 1 に示す。

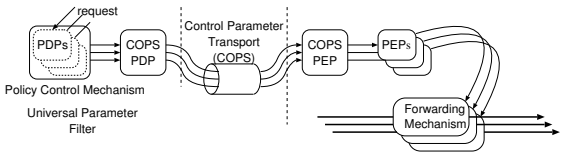

図 1: 提案システム (KUMA:次世代広域ネットワーク制御システム)

システムには以下の概念が含まれる。

1) 汎用ライブラリ
2) ポリシ制御機構
3) 一般化パラメータフィルタ

これにより、ポリシおよび要求と転送機構の柔軟な対応づけを行い、次世代ネットワークサービスの提供を可能にした。また、SDK (Software Development Kit) の提供を行い、コンポーネントの組合わせで対応しきれない変化にも、最小のコストで追従できる環境を実現した。

(2) 汎用ライブラリ KA

共有ライブラリ KA は、KUMA を構成するデーモン群開発にもちいたツールキットである。本ツールキットは、あやめプロジェクト [11] の協力を得て開発しており、同プロジェクトでも利用されている。

本ライブラリにより、以下の効果が得られた。

- 開発コストの削減
- デバッグコストの削減
- ユーザへの共通インターフェースの提供

共有ライブラリ KA は、基本機能を提供する「きのこ」と、ユーザインターフェースを提供する「たけのこ」により構成される。

a) きのこ

「きのこ」は、デーモンが必要とする基本機能を提供する。デーモンを開発する際にしばしば問題になる、メモリの割り当て状態の管理やバッファ管理、ロギングとシグナルの設定、ネットワーク入出力や定期的なイベント管理などは、この機能を利用する事により、簡単に扱える。

「きのこ」の主な機能を以下に示す。

- メモリ管理 / リサイクルメモリ管理
- リスト
- バッファ
- I/O ストリーム
- 疑似スレッド
- ロギング
- シグナル
- アクセス制御

b) たけのこ

「たけのこ」は、デーモンの挙動を動的に制御するためのユーザインターフェースを提供する。この機能を利用する事で、動作中のデーモンに接続し、動作状態や設定変更を行なうためのインターフェースを容易に開発できた。

「たけのこ」の主な機能を以下に示す。

- メディア (telnet / ssh / コンソール)
- PTY
- シェル
- コマンド管理/検索
- パスワード・ユーティリティ

メディア/PTY/シェル

メディア/PTY/シェルは、図 2 のような 3 層構造になっている。

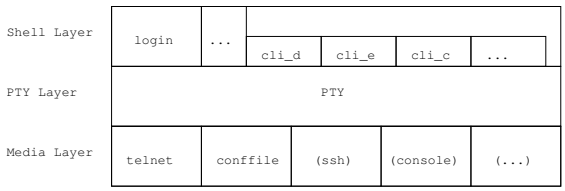

図 2: メディア/PTY/シェル

メディア層では、telnet や ssh など、ターミナルサービスへの窓口を提供している。また、設定ファイルの読み込みや書き込みも、メディアのひとつとして実現されている。現在は、telnet および設定ファイルのみが実装されているが、将来的には ssh やコンソールへの対応も検討している。

PTY 層は、様々なメディアからのアクセスを統合管理している。また、各アクセスに対して起動されるシェル群を管理し、メディアと対応するカレントシェルとの間のデータ交換を制御している。

シェル層は、PTY 層から渡されたデータに対する処理を行なう。シェル起動方法には以下の二つがある。

exec
: 新しく起動されたシェルがカレントシェルと入れ替わる。(カレントシェルは終了する)

invoke
: カレントシェルを維持したまま、新しく起動されたシェルがカレントシェルになる。

exec では、新しく起動したシェルが終了しても、直前のシェルに復帰しない。一方、invoke の場合は元のシェルが新しいシェルの終了を待ち、シェルが終了すると元のシェルに復帰する。

例として、telnet で接続した場合の一連の動作を以下に示す。また、そのときの挙動を図 3 に示す。

1. メディア層で telnet を受け付け、インスタンスを作成
2. PTY を新規に割り当て
3. シェル「login」を起動
4. 認証に成功すると、シェル「cli_disable」を exec
5. enable コマンドでシェル「enable」を invoke
6. 認証に成功すると、シェル「cli_enable」を exec
7. シェル「cli_enable」が終了すると、「cli_disable」が復帰
8. シェル「cli_disable」が終了すると、接続を切断

```
 $ telnet localhost 4000
Trying ::1...
Connected to xxx.xxx.wide.ad.jp
Escape character is '^]'.

  KA Library Test Server

Login: naoto
Password: *********
libka> enable
Password: *********
libka# quit
libka> quit
Connection closed by foreign host.
```

図 3: メディア/PTY/シェルの例

コマンドライン・インターフェース

「たけのこ」は、標準的なシェルのひとつとして CLI (Command Line Interface) を提供している。このシェルは、コマンドの補完や候補の出力、コマンドライン・エディタ、ヒストリなど、利用するユーザの利便性を高める多くの機能を持っている。

CLI は、コマンド・ルートとバックエンドから構成されている。コマンド・ルートとは、コマンドの集合を表すもので、その CLI 上で利用できるコマンドを規定している。一方、バックエンドは、コマンドの補完などのコマンド・ルートに依存しない機能を提供している。

現在、CLI バックエンドを持つシェルは以下の 3 種類が用意されている。

cli_disable
: 非特権モードの CLI シェル。状態の確認などを行なうことができる。

cli_enable
: 特権モードの CLI シェル。各種設定や設定ファイルの閲覧、保存ができる。

cli_conf
: 設定モードの CLI シェル。各種の設定が動的に変更できる。

これらは、前述したように、図 4 のようになっている。

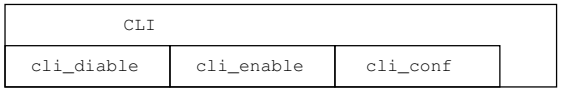

図 4: CLI の構造

コマンド管理/検索

「たけのこ」の大きな特徴に、コマンド管理/検索とシェルが完全に分離していることが挙げられる。そのため、コマンド管理/検索機構を利用し、メニュー・インターフェースを持つような新しいシェルを作成することができる。

図 5 に、コマンド管理/検索機構を示す。

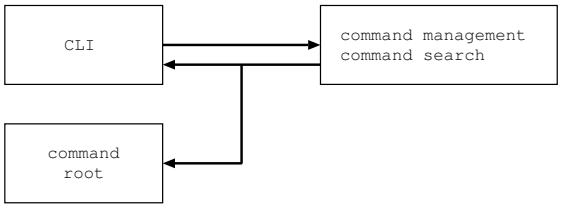

図 5: コマンド管理/検索機構の構造

(3) ポリシ制御機構

a) ポリシ制御系に対する要求事項

通信品質保証サービスを提供する場合には、さまざまな制御や決定を行う必要がある。なんらかのエンティティが通信品質に対する要求を発行した場合を考えた場合、制御系は大きく分けて以下のふたつの処理を行わなければならない。

- 要求に対する意思決定
- 転送層への決定した意思の反映

要求に対する意思決定

受け取った要求は、そのままネットワークに反映できるわけではない。確保できる資源は有限であり、確保できたとしても、どの機器を設定すれば良いかを判断する必要がある。また、その他の条件で、要求が拒否される場合もある。これらの条件には、以下のようなものがある。

認証
: 要求を発行したユーザが有効かどうかを判断する。また、ここで認証されたユーザのユーザ名は、後の判断で利用されることもある。

ポリシ
: ネットワークのポリシとして、発行された要求が受理可能かどうかを判断する。例えば、企業において特定のサービスを重役以上にだけ許可するような場合、ユーザ名と要求、および、ネットワークのポリシデータを比較してその要求を受理するかどうかを判断しなければならない。

資源確保
: 要求を実現するために、ネットワーク上のリンク帯域や経路、機器のバス帯域などを場合に応じて確保する必要がある。また、どの機器にどのような設定をするか、ということも決定しなければならない。

中間エンティティへの対応

通信品質を制御するとき、対象となる通信を識別する必要がある。しかし、現在のインターネット上には通信を識別するための要素を、途中で変換するようなエンティティが多く存在している。アドレスを変換する NAT が代表的な例である。通信に対して e2e で一貫した制御を適用するためには、このようなエンティティから変換情報を取得し、通信の識別を矛盾なく行えるようにする必要がある。

ここにあげた条件は、すべて必要なわけではない。組織の方針によって、多くの組み合わせ方が考えらる。また、ここにあげたもの以外にも判断基準は考えられる。

多くの判断基準をどのように組み合わせるか、あるいは判断基準自体の内容をどのようなものにするか、ということは、サービス提供者によって大きく異なる。したがって、これらの判断基準を必要に応じて再構成し、判断基準間の間隙をスムーズに結び付けるようなフレームワークが必要になる。

転送層への決定した意思の反映

要求に対する意思を決定したあと、その決定を転送層へ反映しなければならない。従来のインターネットアーキテクチャでは、制御層と転送層が同一機器内に閉じて存在していたため、特にこの処理が注目されることはなかった。

一方、『転送層と制御層の分離モデル』では、転送層は従来どおり広範囲に分散しているのに対し、制御層は単一になる。そのため、転送層への意思反映には、制御層から転送層への意思伝達が必須となる。

われわれは、細分化された意思決定機構、および、それらの意思統合機構、さらに転送層への意思反映までを含めた通信品質要求に対する管理フレームワーク、KUMA (KUMA's Unified Management Architecture) の設計と開発を行なった。このフレームワークの実現で、従来は一般化が困難であったポリシを、細分化ポリシの組み合わせとして表現することが可能になる。このフレームワークを広域に展開すれば、そのドメイン内での e2e 通信品質保証が実現できる。

b) 制御パラメータトランスポート

KUMA のポリシ制御機構では、設定が必要になるネットワークエンティティの自動的な検出を行なう。ポリシ制御機構によって制御パラメータが決定されたのち、対象となる個々のエンティティのみにポリシを送信する。

c) COPS (Common Open Policy Service)

本機構では、制御パラメータトランスポートプロトコルとして、COPS (Common Open Policy Service)[12] を採用した。COPS は、ポリシ転送機構として標準化されており、設定が必要になるネットワークエンティティの状態を管理する機構を有しているため、採用した。

COPS では、PDP (Policy Decision Point) と PEP (Policy Enforcement Point) 間でポリシの交換が行なわれる。PDP は、ポリシ判断を行なう機構であり、PEP は PDP により判断された情報が反映される機構である。PDP は、ネットワークへの要求やポリシの集合を入力とし、転送機構へ通知する制御パラメータを出力とする制御機構である。さまざまな抽象度や粒度の要求を入力とするため、多段階での要求の具体化を行った。そのため、具体化する対象によって異なるコンポーネントを用意し、全体の協調によって意思決定を行った。

PEP は、PDP に対して TCP (Transmission Control Protocol)[13] によるコネクションを確立する。PEP と PDP 間のコネクションは維持され、PDP は、PEP での状態を把握する。PDP は、複数の PEP からのコネクションを維持できる。通知されたパラメータを転送層に依存した形式に変更し、実際の機器に反映させるサーバ群の設計と実装および評価を行った。COPS PEP は、ポリシ制御情報を転送するのみであるため、機器の設定をする PEP 群が必要である。本機構では、抽象的な記述を個々の機器に適した具体的な設定に変換し、機器を設定する PEP を実現した。

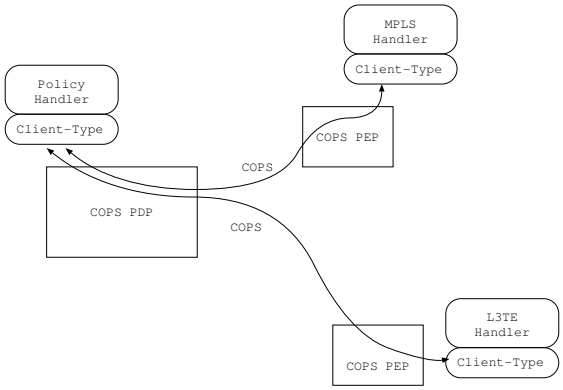

図 6: COPS による制御パラメータトランスポート

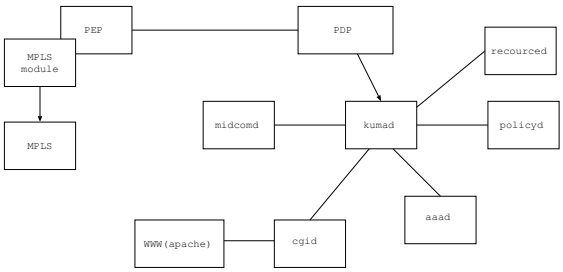

図 7: 制御パラメータトランスポート概要

d) COPS 実装

本機構では、制御パラメータトランスポートとしての COPS の設計と実装を行った。図 6 に本機構における COPS 設計を示す。

本機構では、COPS PDP と COPS PEP は Client-Type により識別される PDP (Policy Decision Point) 群と PEP (Policy Enforcement Point) 群のためのトランスポートである。意思決定や機器に固有のパラメータに関しては COPS は関知しない。そのため、転送機構に依存しない汎用制御パラメータトランスポートとして利用することができる。

本機構では、PDP は複数の PEP とのコネクションを維持し、必要に応じてポリシ制御情報を PDP から各 PEP に送信する。PDP は、ポリシ制御情報を最適な PEP にのみ送信する。

本実装における制御パラメータトランスポート概要を図 7 に示す。

図 7 では、PEP は MPLS 制御機構と接続している。MPLS 制御機構との接続は PEP 内の MPLS モジュールにより実現される。本システムでは、PEP に組み込みモジュールを追加することができる。組み込みモジュールは、PEP の load コマンドによりロードされる。MPLS 制御機構は、PEP から転送されるポリシ情報に基づいて設定を行なう。

図 7 では、PDP は TCP を利用して、kumad と接続している。本システムでは、kumad が包括的にポリシ情報を扱い、PDP はポリシ情報トランスポートとしてのみ動作する。ポリシ制御機構である kumad に関しては、e) で述べる。

e) kumad

制御機構に対する外部からの入力には、ユーザからの高度に抽象化された要求や、管理者からの具体化された制御パラメータなど、さまざまな抽象度および粒度のものが考えられる。

そのため、数多くの条件を柔軟に組み合わせ、思い通りのサービスを提供するためのフレームワークが必要になる。また、サービス提供者独自の条件を追加する場合にも、出来るだけ少ないコストで実現できるような仕組みが必要である。

これを実現するため、KUMA では条件を細分化し、それぞれの条件に基づく意思決定のためのモジュールを、細分化条件決定機構として分離した。さらに、細分化条件統合機構によって、細分化された意思を統合し、系全体の意思決定を行う。

細分化条件決定機構は、各々が独立したデーモンになる。また、課金管理システムのように、ほかのサービスとの共用も考慮されいる。これらのデーモンが生成する意思は、次に述べる細分化条件統合機構によって、通信品質要求に対する系全体の意思に統合される。

本設計では、細分化されたそれぞれの要素は、kumad により包括的に制御される。kumad は、様々なポリシ判断デーモンに接続し、必要に応じてそれらに判断を仰ぐ。細分化された条件として、現在の KUMA では以下のものを実装した。

- CGId
- AAAd
- RESOURCEd
- POLICYd
- MIDCOMd

各デーモンに関しては次節にて述べる。

f) CGId (**要求受付**)

通信品質要求はさまざまな方法で発行される。ここでは、そのひとつとして WWW のページ上から CGI で発行する手段を提供する。CGId はは CGI からのデータを受取り、細分化条件統合機構へ転送する。また、細分化条件統合機構から受け取った結果を、該当する CGI に返送し、ユーザに対する出力を促す。

g) AAAd (**認証/権限委譲/課金**)

認証/権限委譲/課金の処理を行う。特に、現在は課金と認証を中心に実装している。

認証は、品質保証要求を発行したユーザが正規のユーザであるかを判定する。また、課金を行ったりポリシを適用する際、ユーザを見分けるために利用する。

課金は、ある特定のユーザが資源を長期にわたって占有することを防止できる。これにより、資源配分の平等性を実現することになる。

外部からの情報操作には、独自のプロトコルを利用している。しかし、将来的には RADIUS[14] や DIAMITER 等、標準的なものを採用し、さらなる相互接続性および既存資源の有効利用を図る予定である。

h) RESOURCEd (**資源管理**)

通信品質保証を実現するためには、ネットワーク上の経路やリンク帯域を確保しなければならない。ここでは、要求にしたがってこれらの資源を計算する。

要求を満たせるだけの資源を確保できなかった場合は、その旨を細分化条件統合機構に通知し、要求を拒否する。

一方、資源が確保できた場合はその資源を借り押さえる。また、設定を行う必要のある機器をリストアップし、それらの機器に対する設定を生成したのち、細分化条件統合機構に通知する。最終的に要求が許可された場合、借り押さえした資源は使用中として認識される。一方、ほかの細分化条件決定機構に拒否されたり、設定に失敗した等で要求が満たされなかった場合には、借り押さえした資源を解放する。

i) POLICYd (**ポリシ**)

ユーザごとに許可するサービスを変えるなど、あらかじめ決められたポリシにしたがって意思決定をする部分である。細分化条件統合機構から受け取ったユーザ情報や要求の内容などを検査し、ポリシ情報を記録したデータベースと照合を行ったのち、その結果を返す。

j) MIDCOMd

現在のインターネット上には、識別のための要素を変更するエンティティが数多く存在している。たとえば、NAT (Network Address Translation)[15] や NAPT (Network Address and Port Translation) が代表的である。

通信品質保証を実現するためには、対象となる通信を識別する必要があるが、上記のようなエンティティが介在した状態では、そのエンティティの前後で通信を一貫して認識することができない。

MIDCOMd はそれらのエンティティから識別用その変更情報を取得し、機器に設定するためのフィルタを変更する。これによって、e2e における一貫した通信の識別が可能となり、上記のようなエンティティが介在した状態でも、通信品質保証サービスを提供することができるようになる。

k) **細分化条件統合機構**

細分化条件統合機構は KUMAd として実装されている。KUMAd 自体は、通信品質要求に対して独自の判断を加えることはない。すべて、細分化条件決定機構 (群) に委譲し、各細分化条件決定機構の意思を統合する。

通信品質要求を許可した場合には、制御パラメータトランスポートに対して機器設定情報の配布を依頼する。制御パラメータトランスポートから設定の完了通知が来ると、各細分化情報決定機構にその旨を通知し、契約を完了する。契約は KUMAd で管理する。

各細分化条件決定機構との間は、それぞれが独自のプロトコルを利用している。そのため、細分化条件決定機構ごとにモジュールを作成し、動的にロードする形式をとる。

(4) **一般化パラメータフィルタ**

a) **パラメータフィルタに対する要求事項**

通信品質を制御する時、対象となる通信を識別し、その通信に対して適用する処理を決定しなければならない。通信の実体は、当事者間である意味を持った一連のデータの流れである。通信を制御するには、この一連のデータの流れ、フローを識別し、それに対応する操作の集合を決定することが必要となる。

多くの場合、フローに対応する操作を決定するために、インターネット層の始点アドレスと終点アドレス、および、トランスポート層の始点ポートと終点ポート等を利用する。しかし、これらを利用する順序や優先順位によって、結果が変わったり、結果が競合することもある。

例えば、透過型プロキシと特定ユーザの優先制御を、並行して利用する場合を考える。この場合、次のふたつのルールが必要である。

- 終点ポートが 80 のフローはプロキシへ
- 始点アドレスが a.b.c.d のフローは専用線へ

図 8 は、このルールが適用されるフローを表したもので、縦軸が終点ポートを、横軸が始点アドレスを示している。また、青色の線は 1 のルールが適用されるフローの集合を、赤色の線は 2 のルールが適用されるフローの集合を表している。

ここで、始点アドレスが a.b.c.d、終点ポートが 80 であるようなフローに着目する。このようなフローは図中の真中、ふたつの線が交差する部分に該当する。この場合、どちらのルールを適用すれば良いか明確に決定する事ができ

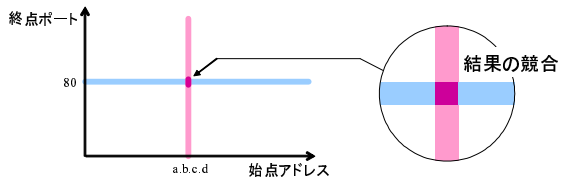

図 8: 結果の競合

ない。

このような問題は、ほかにも考えることができる。たとえば、始点アドレスとそのプレフィックス長、および、終点アドレスとそのプレフィックス長で表されるルールを考える。このルールでは、入力となる IP データグラムに対して、始点アドレスと終点アドレスの最長一致を考えるものとする。このとき、始点アドレスが最長一致するルールと、終点アドレスが最長一致するルールが同じでない場合、どちらを適用すれば良いか明確に決定する事ができない。

上記のふたつの例では、複数のルールが競合するため、ここにあげたそれぞれの条件だけでは一意に定めることができない。

このような問題は、以下のモデルに一般化することができる。

- ある対象をキーとして
- ある分類スキーマに基づく分類レコード群から
- 適合レコード (群) を抽出する

ここで、対象、分類スキーマ、分類レコード、適合レコードとは、それぞれ次のものを示す。なお、括弧の中は、上記のふたつの例に対応する。

対象　入力となるオブジェクト (IP データグラム)

分類スキーマ　対象が持つ属性のうち、分類レコード群の抽出に利用するもの (始点・終点アドレスおよび始点・終点ポート等)。

分類レコード　分類スキーマに基づく、実際のパラメータを定めたインスタンスと、それに対応する適用操作集合の組 (個々のルール)。通常は複数のルールが存在し、分類レコード群を形成する。

適合レコード　分類スキーマに基づいて抽出された分類レコード (結果となるルール)。抽出結果は、一般に複数の分類レコードから構成される適合レコード群となる。

上記の例では、入力となる IP データグラムを元に適合レコード群が決定され、それに基づく操作が適用されることになる。

KUMA では、この一般化したモデルを実現するための柔軟で直感的なフレームワーク、KUPF (KUMA's Universal Parameter Filter) を設計し、この問題に対して解を与えた。KUPF フレームワークを実現することにより、より複雑なポリシを構成するフィルタを記述できる。さらに、このフレームワークを広域に展開することにより、e2e でフローに対する一貫した操作の適用が可能となる。KUMA では、このパラメータフィルタをポリシ制御機構等に応用した。

b) KUPF

適合レコード群中の適合レコードは、明らかに分類スキーマに基づいて対象に一致する。これに着目すると、分類レコード群から適合レコード (群) を抽出する手順は、大きく次の二段階に分類することができる。

第一段階
　対象に一致する分類レコードをすべて選出
第二段階
　第一段階の結果から、抽出ポリシにしたがって適合レコード (群) を抽出

第一段階

第一段階では、対象に一致する分類レコード全てを選出する。そのため、単純な抽出処理のみで実現できる。

第二段階

第二段階では、抽出処理にポリシが反映されなくてはならない。抽出処理時に影響を与えるポリシは、最終的には、適合レコードの検索方法や検索順序などにより反映される。

第一段階において、複数のレコードが適合した場合、第二段階では第一段階の結果から最適なものを選択しなくてはならない。第二段階における選択は、各レコードの適合度合を数値化して比較することにより行なわれる。この数値化の方法や基準を決定するのがポリシである。

しかし、各レコードを数値化した適合度合が同一であるレコードが複数検出されてしまう場合もある。本機構では、そのような問題を解決するため、以下の方法を利用した。

1) 分類レコードの各フィールドの検索順序の設定
2) 適合レコードの優先順序の設定

1) は、適合度合を数値化するために検索を行なう各フィールドの検索順序を決定する。例えば、「IP アドレスの送信元を宛先より先に検索する」などのポリシを決定する事により、数値化された適合度合は変化する。

2) は、検索されるレコード全てに順位づけを行なう方法である。これにより、同一な適合度合が発生しても結果は一意に定まる。

4　実験

本章では、開発を行なった KUMA システムを利用して行なう予定である実証実験について述べる。

(1) **実験概要**

WIDE Project はインターネットとその関連技術に関する研究と開発を行なう非営利団体である。WIDE Project で は春と夏の毎年 2 回、メンバによる 4 日間の合宿を行なっている。WIDE 合宿では「WIDE Camp-net」と呼ばれる一時的な運用ネットワークを構築し、インターネットとの接続性を確保するとともにさまざまな実験を行なう。例えば、IPv6 や MPLS のような最新のネットワーク技術を合宿参加者に提供している。

我々は、2002 年 3 月に行なわれる WIDE Project の 2002 年春の合宿においてユーザから動的に帯域要求を行なえるネットワークを運用・実験を行なう。

実験項目として、以下のものが挙げられる。

- 細分化条件統合機構
- 細分化条件決定機構
- 制御パラメータトランスポート
- 一般化パラメータフィルタ
- あやめプロジェクトとの協調による、MPLS/CR-LDP の制御
- LIN6[16, 17, 18] との協調による、品質契約を維持した透過移動

この項目からわかるように、今回の実証実験では、他のふたつのプロジェクトと協調して実験を行う。

あやめプロジェクトは、次世代インターネットの基盤構造に対する考察をもとに、新しい MPLS[10] の実装と研究を行っている。今回の実験では、KUMA に対する転送層としてあやめプロジェクトの MPLS 実装を利用する。また、MPLS のシグナリングプロトコルとしては、トラヒックエンジニアリングに適した CR-LDP[19] を利用する。

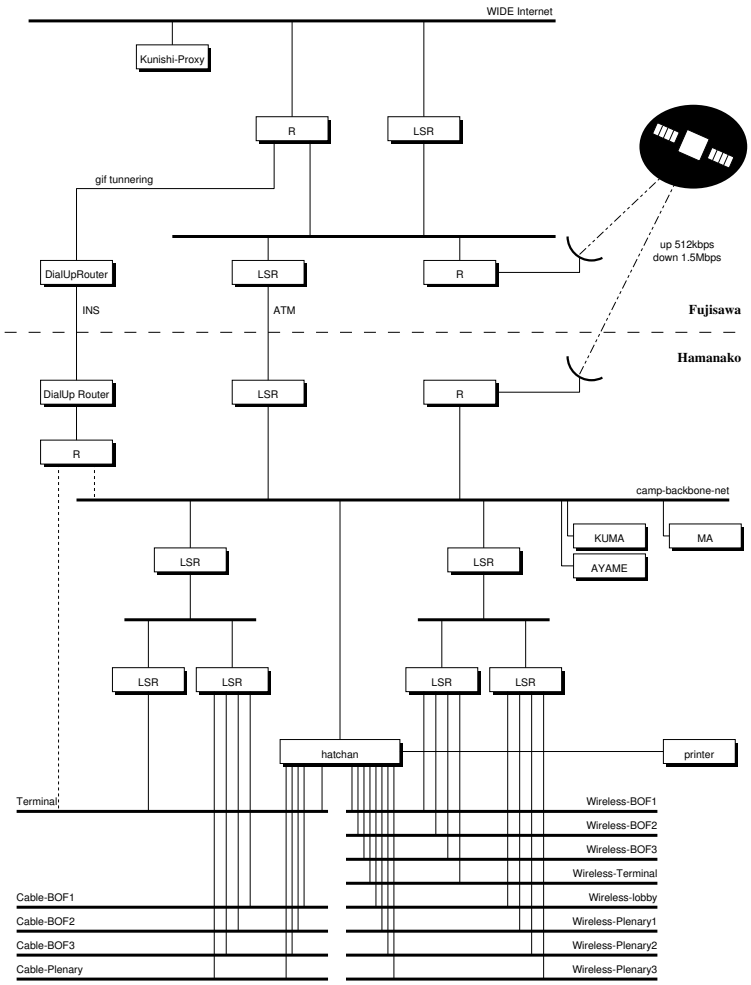

図 9: WIDE 合宿実験トポロジ

LIN6 は、IPv6 を利用した携帯端末端末に対し、透過的なネットワーク間移動を実現するための仕組みである。LIN6 では、通信に利用するアドレスが、ユーザランドから見た場合とネットワーク上で観測した場合で異なる。さらに、ネットワーク間を移動した場合にも変更される。今回の実験では、ユーザランドから見た単一のアドレスでの通信品質要求に対し、実際に通信に使われるアドレスと動的な対応付けを行うことによって、LIN6 ユーザの利便性を高める。

図 9 に、WIDE 合宿における実験ネットワークのトポロジを示す。WIDE 合宿ネットワークの対外線として、ATM による地上線と衛星回線が用意される。衛星回線は、遅延が大きいといった特徴を持った回線である。一方、地上回線は衛星回線に比べ著しく遅延が小さい回線である。

本実験では、WIDE 合宿参加者をユーザとして動的 SLA 環境でのユーザの挙動を調べる。ユーザに対してのインターネットコネクティビティは、衛星回線を利用して提供される。

衛星回線は大きな遅延があるため、telnet や ssh などの即時性が要求されるアプリケーションを利用するには不適切である。ユーザは本システムを利用する事により地上回線を利用できる。地上線を利用することにより、衛星回線と比較して小さな遅延による通信ができる。

本実験により、本システムの有用性を示す事ができる。

5 結び

我々は、高機能化が進んでいる数多くの転送機構に対応し、ネットワークの機能としてさまざまなサービスを提供するために、汎用的・統合的な制御機構ソフトウェア KUMA の開発を行なった。

次世代ネットワーク制御機構では、多様化する要求やポリシを、多様化する転送機構に反映するという機能が必要である。また、これらの要求やポリシ、転送機構は時間の経過と共に変化していく。このため、制御機構を必要に応じて再構成し、それらの間隙をスムーズに結び付けることができなければならない。

実装を行なった KUMA は、これらの要件を満たすソフトウェアである。また、実装を行なったソフトウェアは、フリーソフトウェアとして、BSD タイプのライセンスによる配布を行なっている。フリーソフトとしての配布を行なう事により、同様の研究を行なっている他の団体も本実装を利用する事により、容易に研究が進められる。

6 参加企業及び機関

本開発は、情報処理振興事業協会の未踏ソフトウェア育成事業プロジェクトおよび有限会社ぬいの協力を得て開発

を行った。

参考文献

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service, December 1998. RFC 2475.

[2] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB, June 1999. RFC 2598.

[3] Naoto Morishima, Akimichi Ogawa, Keijiro Ehara, and Youki Kadobayashi. Field-trial of dynamic sla in diffserv-capable network. In J.Crowcroft, J.Roberts, and M.I.Smirnov, editors, Quality of Future Internet Services, number 1922 in Lecture Notes in Computer Science, pages 117–128, Berlin, Germary, September 25-26 2000.

[4] 森島 直人, 小川 晃通, 顕原 桂二郎, 染川 隆司, and 山口 英. 動的資源予約システムの運用実験. インターネットコンファレンス 2000 論文集, November 2000.

[5] 森島 直人, 小川 晃通, 顕原 桂二郎, and 染川 隆司. Bandwidth broker system の開発と運用実験. 情報処理, 41(10), October 2000.

[6] WIDE Project. http://www.wide.ad.jp/.

[7] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group, June 1999. RFC 2597.

[8] Naoto Morishima, Akimichi Ogawa, Hiroshi Esaki, Osamu Nakamura, Suguru Yamaguchi, and Jun Murai. Field-trial of dynamic sla in diffserv-capable networks. International Workshop on Next Generation Internet and its Applications 2001, February 2001.

[9] Naoto Morishima, Akimichi Ogawa, Hiroshi Esaki, Osamu Nakamura, Suguru Yamaguchi, and Jun Murai. Preliminary field-trial for qos routing and dynamic sla. IEICE: Special Issue on Internet Technology, January 2001.

[10] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture, January 2001. RFC 3031.

[11] Ayame project. http://www.ayame.org/.

[12] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) Protocol, January 2000. RFC 2748.

[13] J. Postel. Transmission Control Protocol, September 1981. RFC 793.

[14] C. Rigney, A. Rubens, W. Simpson, and S. Willens. Remote Authentication Dial In User Service (RADIUS), April 1997. RFC 2138.

[15] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT), January 2001. RFC 3022.

[16] Masahiro Ishiyama, Mitunobu Kunishi, Keisuke Uehara, Hiroshi Esaki, and Fumio Teraoka. Lina: A new approach to mobility support in wide area networks. IEICE Transactions on Communications, 2001.

[17] Mitsunobu Kunishi, Masahiro Ishiyama, Keisuke Uehara, Hiroshi Esaki, and Fumio Teraoka. An analysis of mobility handling in lin6. International Symposium on Wireless Personal Multimedia Communication, 2001.

[18] Masahiro Ishiyama, Mitsunobu Kunishi, and Fumio Teraoka. Lin6: A new approach to mobility support in ipv6. International Symposium on Wireless Personal Multimedia Communication, 2000.

[19] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas. LDP Specification, January 2001. RFC 3036.