システムソフトウェアと
オペレーティング・システム 64−23
マルチメディア通信と分散処理 65−23
（1994. 5. 20）

Service Switching 技術の研究

多田 信彦[1] 山口 英 山本 平一

奈良先端科学技術大学院大学

一般に行なわれているアクセスコントロールの方法は、サービス提供の可否の面で、0または1のディジタルな制御である。よって、さまざまな環境に応じて変化するケーパビリティに対応したサービスのレベルの設定ということはできなかった。許された人だけが、許されただけのサービスを享受できる機構が必要である。本論文では、柔軟なアクセスコントロールを実現するサービススイッチングのモデルを提案する。そして、現状のシステムに適用することを試みる。これは、フィルタリングとディスパッチとの機構をもったスイッチングを導入することにより実現されている。

The Research on the Service Switching Technique

Nobuhiko Tada[2] Suguru Yamaguchi Heiichi Yamamoto

Nara Institute of Science and Technology, Japan

It is popular to install firewall system at the gateway to the Internet. Currently available firewall systems like WRAPPER or XINETD provide access control functions which are "all or nothing" basis. Ideally, firewall system can accept or deny requests based on who requests the service or where the request was generated. Furthermore, it is preferable that the system can put some kinds of restrictions (e.g., bandwidth, service available period) on access requests. We propose a new firewall system which is called "Service Switching." Service Switching can control accesses more flexibly for the Internet service providing than other firewall systems. This paper describes the model of the Service Switching and its design and implementation.

[1]松下電器産業(株)より留学中
[2]He also works for Matsushita Electric Industrial Co., Ltd.

1 はじめに

現在、インターネットには、世界中の大学や企業などの研究機関が接続され、その上でさまざまなサービスが提供されている。そこで提供されるサービスは何らかのアクセスコントロールが行なわれている。

一般に、アクセスコントロールは、そのサービスを行なう範囲内 (サービス提供空間) で、サービスの要求者を見て権限を決定するというものである。これまでの方式では、利用可、あるいは利用不可の決定を行なうだけでありサービスの一部制限といった段階的な制御を実現しているものはない。既存の方式を現在のサービスに無理に適用した場合、本来アクセスさせるべきユーザを排除したり、あるいは必要以上の権限を与えるという問題が生じる。

また、インターネットでは、学術ネットワークだけでなく、商用ネットワークも数多く現れており、そのサービスの内容もさらに多様化し変化している。

たとえば、会員制のネットワークサービスを考えた場合、会員の種別によって要求するサービス、あるいはサーバを詳細に指定できるというようなサービス形態も考えることができる。このようなサービスでは、利用者の持つケーパビリティ(capability)によって与えるサービスを適切なものに変化させる機構が必要となる。このような機構では、サービス提供者のポリシがアクセスコントロールに反映されなければならない。

さらに、モーバイル環境が進む中、さまざまな環境から、ローカルな環境へとアクセスする場面が増えるだろう。環境が変化するとそれに応じて、ケーパビリティも変化すると考えられる。このように、環境に応じて変化するケーパビリティに対して、管理者には、もっと適切なサービスの割り当てを行なうことが求められる。許された人だけが、許されただけのサービスを享受する機構が必要である。

本論文では、多種多様化していくサービスに対して、求められる柔軟なアクセスコントロール機構であるサービススイッチングを提案する。サービススイッチングに対して、2つのモデルを示し、その統合された実装を提案する。これは、1台以上のホストで、あるポリシをもってサービスを提供する際に、ユーザやホストからの要求によるサービスの切替えを効果的に管理するための技術である。

本稿では、2節、3節で、サービススイッチングの考え方、提案するモデルを説明し、4節で、その具体的な例を記述する。そして、5節で、3節のモデルを実装する例を示している。さらに、6節で、モデルにおける議論を行なっている。

2 サービススイッチング

この節では、要求者に応じて、割り当てるサービスを変更すること — サービススイッチング — について述べる。

サービス提供者側の基本的ポリシを反映する制御を行なうことは重要である。これは、

> サービスのリクエストに対して、要求者のもつケーパビリティに応じたサービスの割り当てを行なうこと

ということを意味する。

2.1 構成要素

サービススイッチを構成する要素として以下を定義する。

サービス要求者
サービスの要求を出すユーザやホスト

サービスエンティティ
実際にサービスを提供するホストやポート

スイッチングマップ
各サービスにおいて、サービスを許すサービス要求者とサービスエンティティとの対応づけが記述されたテーブル

スイッチャ
スイッチングマップにしたがって、サービス要求者とサービスエンティティとを繋ぐために機能するもの

2.2 サービススイッチングの形態

サービススイッチングの形態は、“ポリシの公開性 (公開か非公開か)” と “スイッチングマップを共

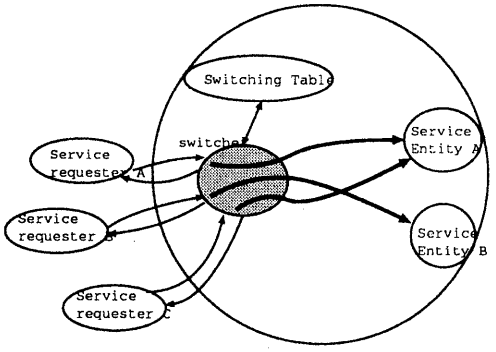

図 1: local model

有するスイッチャの数 (単数か複数か)" とのマトリックスによって 4 種類に分類される。

図 1 は、スイッチングマップを単一のスイッチャで用いて、サービススイッチングを行なうものである。この場合の利点は、内部のホスト以外にスイッチングマップの情報を持たない。したがって、サービスの提供者側のポリシを外部のホストに知られずにすむことである。また、すべてのサービスの要求はスイッチャが管理するので、サービスの利用状況などを把握しやすい。

3 サービススイッチングモデル

この節では、サービススイッチングを実現するモデルについて考える。

まず、コミュニケーションのエンドポイントを意識した場合、図 2 に示すように 2 つのタイプに分類される。

Type1 サービス要求者は、サービスエンティティを意識。サービスエンティティは、サービス要求者を意識。

Type2 サービス要求者は、スイッチャを意識する。サービスエンティティは、サービス要求者を意識する。

次にこれら Type1, Type2 に対応したスイッチングモデル (1),(2) を提案する。

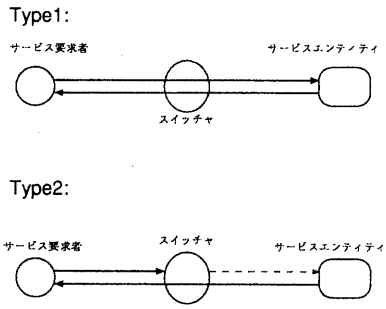

図 2: コミュニケーションエンドポイントによる分類

3.1 スイッチングモデル (1)

このモデルでは、要求者は、スイッチャの向う側 (スイッチャを介さなければ、アクセスできないところ) にいるサービスエンティティに対して直接アクセスしようとするが、その間でその接続をスイッチャが管理する。

このモデルでは、サービス要求者と接続先ホストと接続先ポートの組でそのリクエストを識別する。次に、スイッチングマップの情報を参照することによって、その接続先ホストに対してそのサービスを提供するか否かが決まる。これは、スイッチングマップから、サービス要求者, 接続先ホスト, 接続先ポートに対応づけられる権限 (通過 OK や、通過 NG や、通過しても良いが何らかのスイッチャからの制限がつけられる) を得ることによって、通過の可否が決まる。

このモデルは、電話回線に例えると、直通回線が通じているのに、スイッチャはその通過に関して、ある制限を加えることができるというものである。

3.2 スイッチングモデル (2)

このモデルでは、サービス要求者は常にスイッチャとなるホストに対してサービスを要求する。スイッチャは、サービス要求者と接続先ポートとの組でそのリクエストを識別する。次に、スイッチングマップの情報を参照することによって、そのリクエストをどのサービスエンティティに割り当てるかが決められる。その場合は、転送先ホストと転送先ポートとの組が得られる。何にも対応

づけられていない場合は、そのサービスは拒否されたことになる。

このモデルでは、サービス要求者は、そのサービスを要求するために、サービスを管理している“受け付け”に対してリクエストを投げる。リクエストを受けとったスイッチャは、サービスを提供する側のポリシにしたがって処理を行ない、サービスエンティティにフォワードする。フォワードされたサービスエンティティは、サービス要求者に対してサービスを返す。

これは、電話網でいうところの“代表者電話番号”に似ている。サービス要求者はこの論理アドレスを指定すれば良い。

また、このモデルによって、サーバプールを実現することもでき、応用範囲が広がる。

4 各モデルの具体例

この節においては、先の節で述べたそれぞれのモデルについて、その具体例を述べる。

4.1 スイッチングモデル (1) について

このモデルは、サービス要求者が、スイッチャを越えて直接にサービスエンティティのアドレスを指定して、そのサービスを得ようとするものである。この処理は次のように行なうことができる。

1. スイッチャは、サービス要求者からリクエストを受けとると、ユーザ認証が必要なものかどうかをチェックする。
2. スイッチングマップに対して、サービス要求者、接続先ホスト、接続先ポートを与えることで、通過させてよいかどうかが得られる
3. 通過させていい場合はそのまま通過させる。通過させてはいけない場合はパケットを捨てる。

この場合のスイッチャの操作は、パケットをフォワードするということに他ならないが、その他にスイッチャは、その接続に対して仮想的にバンド幅を制限するといったような制限を加えることができる。その実際の制限の加え方などは今後の研究課題である。

4.2 スイッチングモデル (2) について

このモデルでは、ネットワーク層におけるサービススイッチングについて考え、新たなアドレスを提案し、このモデルのデザインを再び考える。

要求者に、サービスを提供してくれる具体的なホストに対する知識を持たなくても、ある論理アドレスにパケットを送出すると、サービスを提供するホストに対してパケットが届くメカニズムを servicecast と呼ぶことにする (これ以降、SC と呼ぶことにする)。

SC を認識し、SC を受けとり、要求者に対して、サービス提供者が考えるサービスを返すことのできるホストを SC ホストと呼ぶ。SC ホストには、スイッチャとサービスエンティティとがなることができる。とこの SC ホストのグループを SC グループと呼ぶ。また、上で述べたアドレスを SC アドレスと呼ぶことにする。

4.2.1 SC アドレス

SC アドレスは、あるグループで受けとるができるアドレスである。このアドレスの割り当てに関しては、次の 2 つのことを考える。

- **特別なクラスのアドレスを割り当てる場合**
  その送信アドレスから SC アドレスであることを判別できる。しかし、この場合はルーティング情報をやりとりするメカニズムを、新たに考える必要がある。
- **内部のネットワークのアドレス空間の一つを割り当てる場合**
  従来の技術を利用して、ルーティングすることができる。たとえば、host route などで SC の経路情報をアナウンスすることができる。

SC は各サービスの提供者側のポリシに依存するものであり、SC を知らないホストから見れば普通の unicast と同じように見える。

4.2.2 SC のルーティング

SC アドレスを認識しないホストは、SC アドレスのついたパケットを受けとった場合、通常の unicast のルーティングを行なう。SC アドレスを認識するホストは、ルーティングテーブルを参照

することにより、サービス提供者側のポリシに対応した宛先にSCアドレスのパケットを次のように処理を行なう。

1. ルーティングテーブルを参照することにより、サービス提供者側のポリシに対応した宛先を見つけてフォワードする。この際に、送信先のアドレスの変更を行なわない。すなわち、SCアドレスのままでフォワードする。フォワードする方法としては、次にあげるようなことで実現できる。
   - トンネリングによる配送
   - 下の層での工夫(ARPテーブルの細工)
2. ルーティングテーブルから判断して、自分自身で処理する場合には、サービス要求者に対してサービスを返す。

ルーティングテーブル

SCに対して、サービス提供者側のポリシを反映させた条件が記述され、処理すべきパケットか否か判断するのに使われるルーティングテーブルをSCテーブルと呼ぶことにする。SCテーブルはスイッチングマップから作成される。

これには以下の情報が埋め込まれている。

- 要求を切り分けるためのエントリ
  要求元アドレス, 送信先アドレス, 送信先ポート, ユーザ
- フォワードするために必要なエントリ
  転送先ホスト, 転送先ポートなど
- 自システムで処理するために必要なエントリ

ルーティングテーブルの管理方法

SCグループ間ではSCテーブルの情報を共有する。この共有の仕方には、SCグループの構成によって、次の3つに分けられる。

- SCホスト - Non-SCホスト (SH-NS)
  SCホストとNon-SCホストとの結合。この場合には、Non-SCホストは、どのSCが流れてきても自分自身がサービスを行なわなければならない。

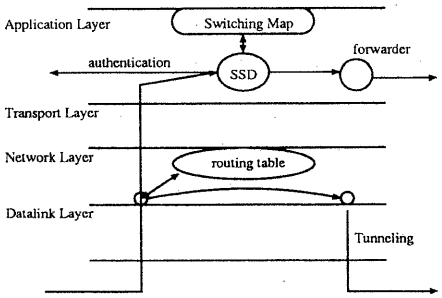

図 3: Service Switching Model

- SCホスト - SCホスト (SH-SH)
  各SCホストが、SCテーブルを保持し、必要である場合にそれらをマージする方法。この場合には、データの完全性を保つことが困難である。
- SC-Master-SC-Slave (SM-SS)
  SCグループのポリシを反映して、SCテーブルを管理するSC-Masterと、その情報を受けるSC-Slaveとで管理する。

5 統合した実装

スイッチングモデル(1)とスイッチングモデル(2)とにおいては、その性格が異なるモデルである。スイッチングモデル(1)は、直接的なアクセスをどのように許すかを条件とともに決めているものであり、スイッチングモデル(2)では、グループに対してリクエストを割り振る(ディスパッチする)ということを行なっている。

この(1),(2)は、モデル的には異なっているが、実際のケーパビリティと権限の対応を得るところで、リクエストに用いられているアドレスによって変化させることによって、スイッチングモデル(1)と(2)とを一緒に実装することができる。

図3はそのモデルを示す。

ここでは、SCをサービスエンティティへ運ぶために、encapsulationしてトンネリング技術を用いている。

次にその処理の流れを見る。

1. スイッチャのインタフェースより、上がってきたパケットがencapsulationされたものか

否かを見る。

2. encapsulation されたものの場合

   - SC のための encapsulation である場合には decapsulation して、再度そのパケットの中身を見て処理を行なう。

3. encapsulation されていないものの場合

   - SC である場合
     SC テーブルを見て処理を行なう。
   - unicast である場合
     ユーザ認証が必要でないサービスの場合、通過させて良いかどうかを判断して処理を行なう。
     ユーザ認証が必要な場合は、上位の層にあげて、SSD(スイッチャの役割を果たす) が認証して処理を行なう。

6 議論

6.1 Host Anycast Service との関係

Host Anycast Service[1] は、最適なサーバを探したいというユーザ側からのビューであり、SC を Host Anycast Service に対応させることは可能である。つまり、Host Anycast Service においては、最適な SC アドレスを返すことで、その要求に対応することができる。

6.2 モデル (2) の別の実装について

4.2では SC によってネットワーク層における、サービススイッチングを述べたが、これは、アプリケーション層においても実現することもできる。スイッチャが、最初にリクエスト要求を受けとったときに、スイッチングマップによって得られる情報から配送先ホストへの接続を行ない、要求者との間をとりもってやらなければならない。このスイッチャは2つのコネクションを管理することになる。

7 おわりに

本論文では、サービススイッチング技術における2つのモデルを示し、そのメカニズムを述べ、それらを統合したサービススイッチングを提案した。このサービススイッチング技術によって、より柔軟なアクセスコントロールが可能になる。

また、スイッチングモデル (1) とスイッチングモデル (2) を一緒に実装することを提案したが、これらは、別々に取り扱われることも可能である。多段に構成されたモデル (1) とモデル (2) ではどのような制御が必要なのか。SC を用いたスイッチングの場合にどのように SC テーブルを管理すべきか、モデル (1) における接続の管理はどのように行なうべきか、どのようにしてスイッチャとサービスエンティティとの間の通信の整合性をとるか、など解決しなければならない問題が残っている。

今後、UNIX マシンに提案したモデルのサービススイッチングを実装し、実用に耐えることを実験していく。また、応用として、firewall をネットワークとの接続点として捉えた上で、このサービススイッチングモデルを適用することにより manageability の向上を図りたい。

謝辞

最後に本研究にあたって、貴重な御意見を与えてくれた WIDE Project の皆様に感謝いたします。

参考文献

[1] C. Partridge, T. Mendez, and W. Milliken. Host Anycasting Service, RFC1546. November 1993.