

規則にもとづく QoS ポリシーのモジュラーな表現 (Rule-based Modular Representation of QoS Policies)

日立製作所 中央研究所
金田 泰

インターネットにおける QoS 保証とそのアプローチ

■ インターネットにおける QoS 保証の必要性

- ミッション・クリティカルな通信の増大
- マルチメディア・トラフィックの増大

■ IntServ と DiffServ (IETF)

- Integrated services (IntServ)
 - フローごとの QoS 制御アーキテクチャ
 - オーバヘッドがおおきく、かつスケーラブルでない
- Differentiated services (DiffServ)
 - クラスごとの QoS 制御アーキテクチャ
 - オーバヘッドがちいさく、かつスケーラブル — 大規模ネットにおいて実用的

DiffServ に準拠したネットワークのモデル

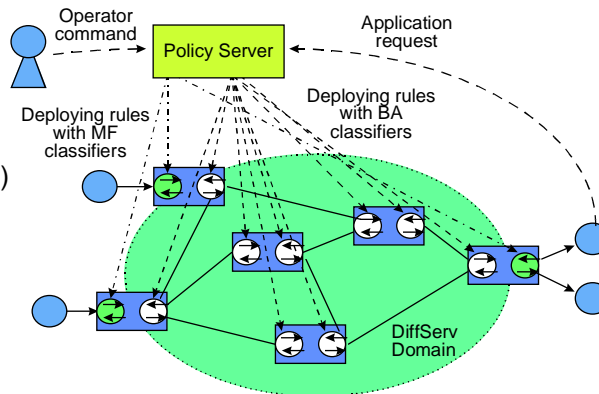
■ QoS ポリシーサーバ

- ルータの QoS 条件の制御のために必要.

■ QoS 機能があるルータ (とスイッチ)

■ ポリシーサーバ - ネットワーク・ノード間のインタフェース

- SNMP
- COPS
- API (CORBA IIOP)
- CLI (コマンド言語)



Networking Architecture Workshop 2000-2-3

3

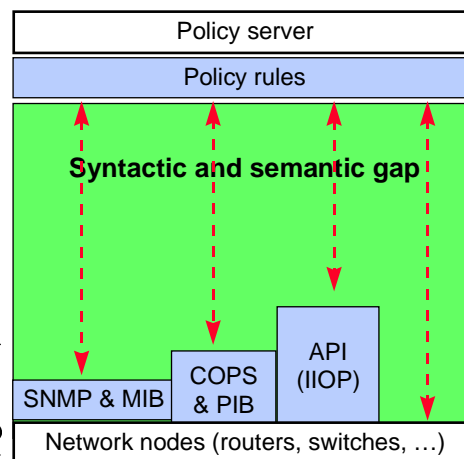
従来のポリシーサーバ-ルータ間インタフェースの問題点

■ 構文が貧困

- SNMP: 個別の値の get/set だけ.
- API: 関数よびだしだけ.
- 構造化の方法がない (制御構造がない).

■ セマンティクスが貧困

- 関係や制約が記述できない.
- プロトコルだけでは非常にかぎられた意味しか記述できない.
- ネットワークの相互運用のために意味の形式的な記述が必要
 - プロトコルの標準化だけでは相互運用性は確保されない. (COPS がいい例)



Networking Architecture Workshop 2000-2-3

4

インタフェースの代案: 規則ベースのプログラミング言語

■ なぜ言語か？

- 構文と意味のくみあわせが言語だから。

■ なぜプログラミングか？

- ポリシーベース制御はプログラミングである。
 - ネットワーク・ノードはパラメタ (データ) だけで設定できた。
 - ノードの機能が複雑化したため、設定用にプログラムが必要になった。

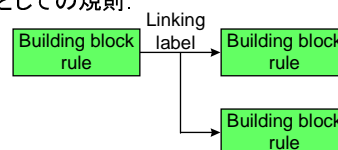
■ なぜ規則ベースか？

- ポリシーは規則ベースのプログラムだから。
- この言語はエキスパート・システム開発用の言語 (OPS5, Nexpert Object など) に類似している。

規則ベース言語の要素

■ 言語の構成要素

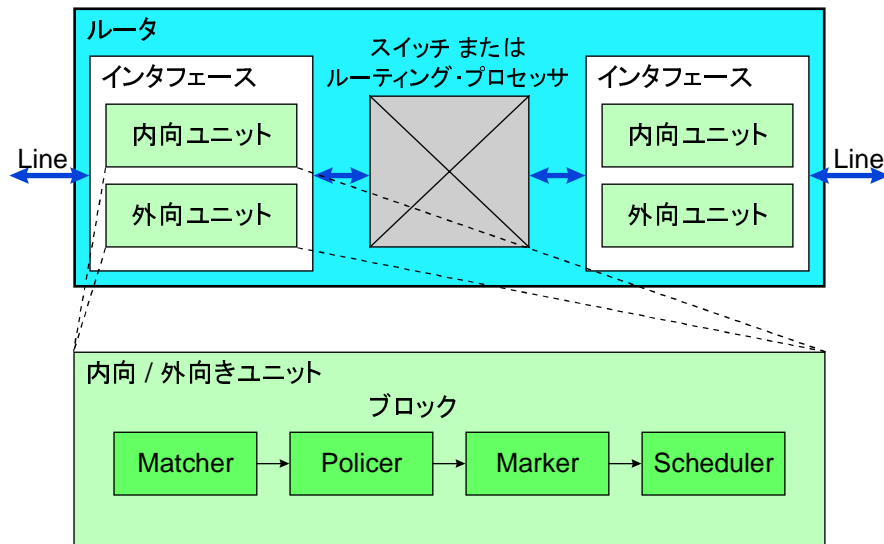
- ブロック規則 (building block rules)
 - ポリシー規則を構築するための素材としての規則。
- 結合ラベル
 - ブロックをつなぐもの。



■ 結合ラベルとはなにか？

- DSCP のようなもの (DSCP = DiffServ code point).
- 結合ラベルの数は無制限。
 - DSCP の数は 64 だけ — 不十分！
- 結合ラベルはパケット上にはない。
 - 結合ラベルはネットワーク・ノード内だけでつかわれる。
 - 結合ラベルはノード内ではパケット外に実際につけられるか、または仮想的につけられる。

DiffServ 機能をもつルータのモデル



Networking Architecture Workshop 2000-2-3

7

要求される素材: ブロック規則の型

■ Matching 規則

- フロー分類のための規則.
- 例: if (Source_ip == 192.168.1.*) ...;

■ Policing 規則 (Metering 規則)

- ポリシング (帯域制御等) のための規則.
- 例: if (Average_rate <= 1Mbps) ...;

■ Marking 規則

- DSCP かきこみのための規則.
- 例: if (...) DSCP = 46;

■ Discarding 規則

- パケット廃棄のための規則.
- 例: if (...) discard_all;

■ Scheduling 規則

- シェイピング / スケジューリングのための規則.
- 例: if (...) queue_priority = 6;

Networking Architecture Workshop 2000-2-3

8

規則ベース言語の概要

■ ポリシー規則の例 (手続き的な記述)

- ```
if (Source_ip == 192.168.1.*) {
 if (Average_rate <= 1Mbps) {
 DSCP = 46; // EF
 queue_priority = 6;
 } else {
 discard_all;
 }
};
```

## 規則ベース言語の概要 (つづき)

---

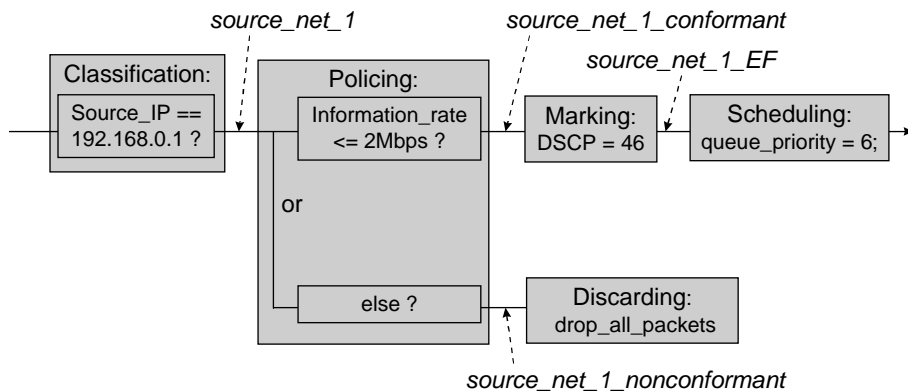
### ■ この言語によるポリシーの記述例

- Matching 規則
  - ```
if (Source_ip == 192.168.1.*) Label = s1;
```
- Policing 規則
 - ```
if (Label == s1 && Average_rate <= 1Mbps)
 Label = s1_conformant;
if (Label == s1 && Average_rate > 1Mbps)
 Label = s1_non_conformant;
```
- Marking 規則
  - ```
if (Label == s1_conformant) {  
    DSCP = 46; Label = s1_EF; };
```
- Discarding 規則
 - ```
if (Label == s1_non_conformant) discard_all;
```
- Scheduling 規則
  - ```
if (Label == s1_EF) queue_priority = 6;
```

規則ベース言語の概要 (つづき)

■ 分解した規則はあまりに複雑?

- 図示してみると, それほど複雑でないことがわかる.



従来のプロトコル / API 上でのブロック規則

■ この規則ベース言語はつぎのインタフェース上に実装可能

- SNMP (MIB 使用)
- PIB (PIB 使用)
- API (関数よびだし使用)

■ ブロック・アプローチのための MIB/PIB

- 旧版を 46th IETF に提案 (1999 年 11 月)
 - ドラフト名: draft-kanada-diffserv-qospifmib-00.txt
 - 以下の WG/BOF で発表:
 - RAP WG (Resource Allocation Protocol WG)
 - CFGMGMT BOF (Configuration Management BOF)
 - Diffserv WG — 質疑だけ

結言

■ DiffServ で必要な大半のポリシー規則は規則ベース言語によって記述可能.

- 5 種類のブロック規則:
matching, policing, marking, discarding, scheduling.
- 結合ラベル.

■ ブロックにもとづくアーキテクチャは DiffServ に限定されない.

- 他の QoS 制御にも適用可能.
- アクティブ・ネットワークにも適用可能 (プログラミング・ネットワーク).

■ 今後の課題

- 規則ベース言語の定義と実装 (COPS 上)
 - 日立 GR2000 への実装もふくむ.