

ポリシーサーバ PolicyXpert における Diffserv ポリシーとそのくみあわせ

金田 泰 (日立製作所)

Brian O'Keefe (Hewlett Packard Company)

ポリシーとは？

- ポリシーとは、ネットワーク (機器) を制御する規則をあつめたもの。
 - ◆ この発表では機器制御のための低水準ポリシーをあつかう。
- 各規則は 条件-動作 型
 - ◆ if 条件 then 動作
- ポリシーが適用される時、通常は 1 個の規則だけが実行される。

ポリシーのくみあわせとは？

■ 複数のポリシーの協調が必要なときがある。

◆ 例: Diffserv におけるエッジとコアのポリシー

- Diffserv (Differentiated Services): ネットワークのエッジでフローをクラス分けし、クラスごとにサービスに差をつけるサービス。
- フローをクラス分けするポリシーをエッジルータに適用し、クラスごとにキュー / スケジュールするポリシーをコアルータに適用する。
- これらのポリシーがうまく協調しないと Diffserv が動作しない。

■ ポリシーの協調のため複数のポリシー規則を明示的にくみあわせるべきときがある。

◆ 例: Diffserv ポリシーは DSCP によってくみあわされる。

- エッジのポリシー規則で特定の DSCP をマークし、コアのポリシー規則でその DSCP をテストする。

■ 「ポリシーのくみあわせ」 (Policy combination) とは

- ◆ ポリシー間の積極的な関係を明示的に指定すること。
- ◆ 「ポリシーの衝突」 (消極的な関係) と対立する概念。

PolicyXpert とは？

■ PolicyXpert は HP / 日立 のポリシーサーバ製品

◆ ネットワーク QoS ポリシーの管理

- Diffserv, RSVP による優先度・帯域制御等

◆ PolicyXpert 1.0 は HP 社単独開発, 2.0 以降は日立と共同開発

- HP 社製品名: OpenView PolicyXpert
- 日立製品名: JP1/PolicyXpert

PolicyXpert における基本的なポリシー構造

■ ポリシー規則の例: DSCP をマークする規則

```
if (Source_IP_address is 192.168.1.1) {  
    DSCP = 10;  -- マーキング  
}
```

- ◆ ただし, この構文は PolicyXpert のものではない
 - PolicyXpert では GUI または XML で入力.

■ ポリシーの形式: P = {規則1, 規則2, ... }.

- ◆ ポリシー中の規則には順序がある.
 - 条件は上から下へ評価される.
 - 最初に条件がヒットした規則だけが実行される.
- ◆ PolicyXpert のポリシーはほぼ IETF の情報モデルに準拠している (PCIM, QPIM のサブセットである).
 - PCIM = Policy Core Information Model
 - QPIM = QoS Policy Information Model

Diffserv のための 3 種類のポリシー

■ Traffic Classifier (CL) ポリシー

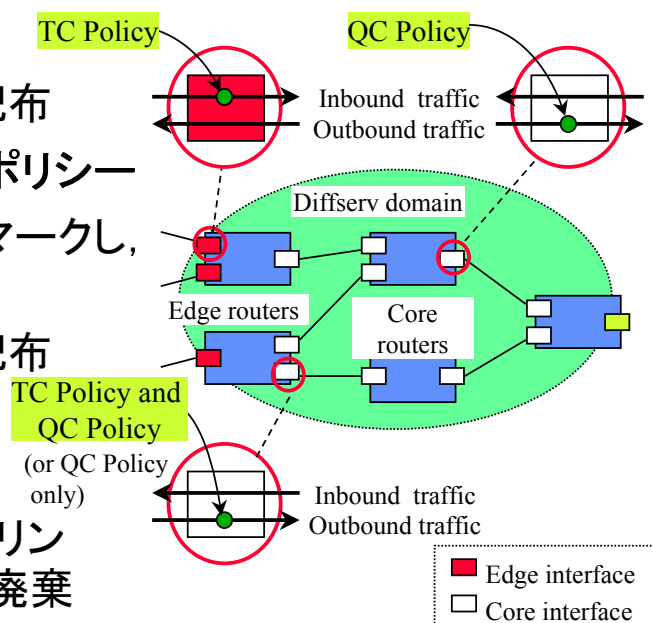
- ◆ フローをクラス分け
- ◆ エッジ・インタフェースに配布

■ Traffic Conditioner (TC) ポリシー

- ◆ パケットを監視・計測し, マークし, 全廃棄する.
- ◆ エッジ・インタフェースに配布

■ Queue Control (QC) ポリシー

- ◆ キューイング, スケジューリング, (ランダムな) パケット廃棄
- ◆ コア・インタフェースに配布



Diffserv のための 3 種類のポリシー (つづき)

■ CL ポリシー規則の例

◆ if (Source_IP_address is 192.168.1.1) {
 CID = "EF_CID"; -- CID は文字列
 }.
}

■ TC ポリシー規則の例

◆ 単純な (独立に機能する) 規則

```
if (Source_IP_address is 192.168.1.1) {  
    DSCP = 46;                -- マーキング  
}
```

◆ より複雑な規則

```
if (CID is "EF_CID") {  
    if (InformationRate <= 10 Mbps) {  
        DSCP = "EF"; -- marking  
    } else {  
        Discard;        -- 絶対廃棄  
    };  
};
```

Diffserv のための 3 種類のポリシー (つづき)

■ QC ポリシー規則の例

◆ 単純な規則

```
if (DSCP is "EF") {  
    SchedulingAlgorithm = "B-PQ"; -- "制限つき優先度キューイング"  
    Priority = 6;                -- 優先度がたかい  
    ShapingRate = 20 Mbps; }.
```

◆ より複雑な規則

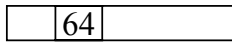
```
if (DSCP is ["AF11", "AF12", "AF13"]) {  
    SchedulingAlgorithm = "A-BW"; -- "Aggregated BandWidth"  
    Max_Queue_Size = 200 packets;  
    CommittedRate = 64 kbps;    -- 最低保証レート  
    DiscardAlgorithm = "Deterministic Discard";  
    if (DSCP is "AF11") {  
        DiscardLevel = 100%;    -- キュー全体が使用可  
    } elsif (DSCP is "AF12") {  
        DiscardLevel = 70%;    -- キューの 70% が使用可  
    } elsif (DSCP is "AF13") {  
        DiscardLevel = 50%; }; -- キューの 50% が使用可  
};
```

ポリシーくみあわせの方法

■ くみあわせるポリシー間のデータフローの指定

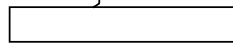
◆ データフロー指定のための 2 種のフローラベル

1) 実フローラベル



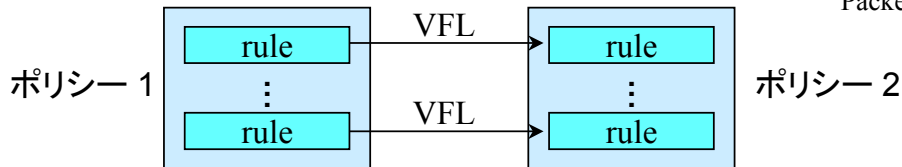
Packet

2) 仮想フローラベル (VFL)



Packet

◆ VFL によって結合されたポリシー規則:



◆ PolicyXPert における仮想フローラベルの 3 分類

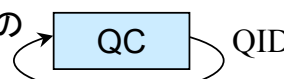
- Classifier ID は CL, TC ポリシーの規則を結合する.



- Traffic ID は TC ポリシー内の規則どうしを結合する.



- Queue Set ID は QC ポリシー内の規則どうしを結合する.



ポリシーくみあわせの方法 (つづき)

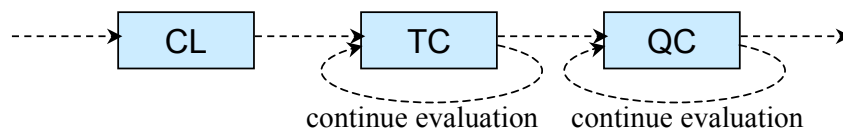
■ くみあわせられたポリシー間の制御フローの指定

- ◆ 制御フローは適切なポリシー記述言語を導入すれば明示的に記述できる.

- 例 order CL1 -> TC1 -> QC1. (CL1, TC1, QC1 はポリシー)

- ◆ ポリシー型の定義において、ポリシーの評価順序をあらかじめきめておくこともできる.

- PolicyXpert ではつぎのようにきめられている:



- TC または QC 規則の実行後には 4 つの可能性がある.

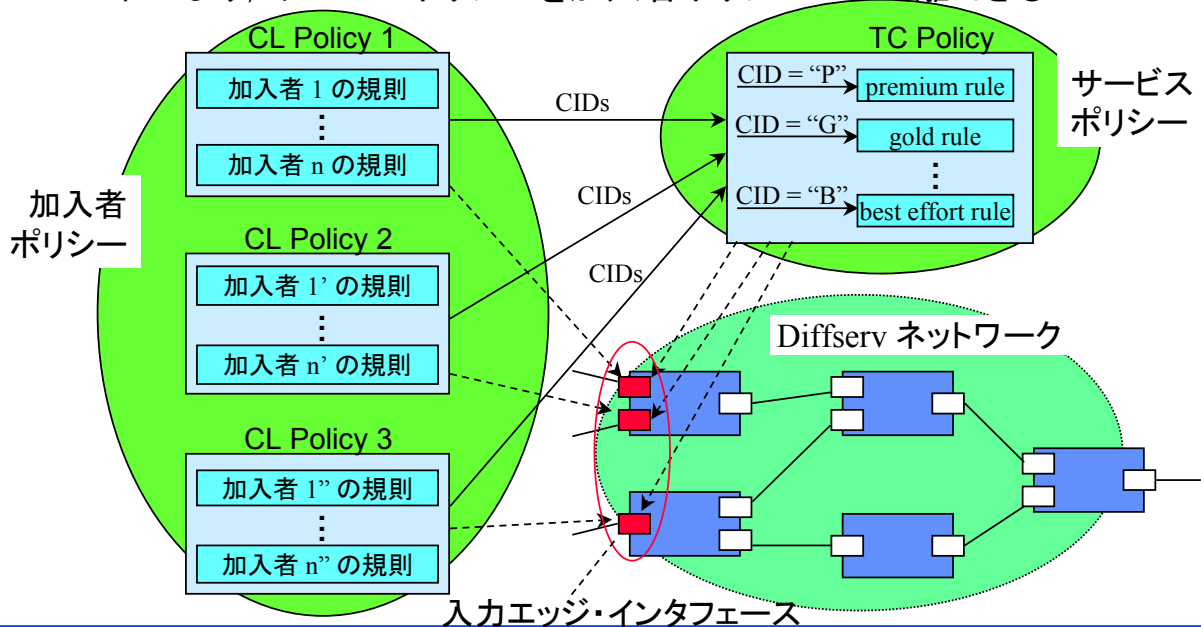
- TC → TC (反復), or TC → QC,
QC → QC (反復), or QC → end.

- この非決定性をなくすため、PolicyXpert においてポリシーの適用を反復するときは規則中に “continue evaluation” を指定する.

ポリシーくみあわせの応用

■ 加入者ポリシーとサービスポリシーとの分離

- ◆ CL Policy を加入者ポリシーとして 1 個のエッジ・インタフェースに配布.
- ◆ TC Policy をサービスポリシーとして複数のエッジ・インタフェースに配布.
- ◆ これにより, サービスポリシーを加入者ポリシーから分離できる.

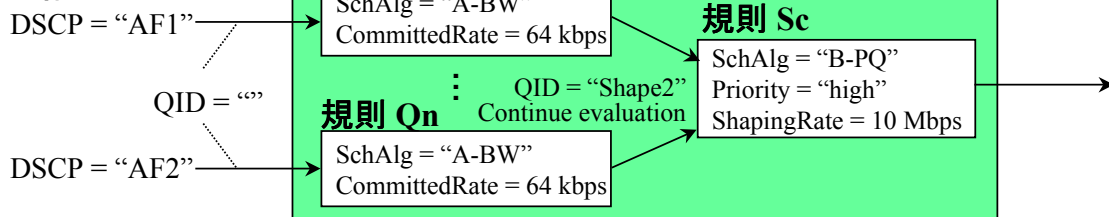


ポリシーくみあわせの応用 (つづき)

■ 階層的シェイパー

QC ポリシー

◆ 構造



◆ 機能

- 規則 Q1, ..., Qn は帯域フェアキューイング (A-BW) により最大 64 kbps にシェイプして出力する.
- 規則 Sc はシェイプされたトラフィックをうけとり, 制限つき優先度キューイング (B-PQ) により最大 10 Mbps のトラフィックを出力する.

■ 階層的ポリサー (帯域をこえるパケットをすてる)

- ◆ 上記のシェイパーと同様の方法によって表現できる.

■ 実装

- ◆ 階層的シェイパー / ポリサーが利用可能な機器はまだない.

PolicyXpert における柔軟さと単純さの両立

- ポリシーを柔軟にくみあわせて複雑なポリシーをつくることができる。
 - ◆ 複数の規則をつなぎあわせることができる。
- 単純なポリシーは単純に表現できる。
 - ◆ 1 個の規則だけで表現できる。
 - ◆ 論理的なうつくしさより単純に表現できることを重視した。

結論

- 複数のポリシーを協調させるために「ポリシーのくみあわせ」が必要である。
 - ◆ データフロー指定のためのフローラベル, とくに VFL
 - ◆ 制御フローの指定: 暗黙に, または明示的に
- PolicyXpert においては, ポリシーのくみあわせによってつぎのことが可能になっている。
 - ◆ 複雑な Diffserv ポリシーの表現
 - ◆ サービスポリシーと加入者ポリシーとの分離
- PolicyXpert においては, 複雑なポリシーが表現できる一方で, 単純な Diffserv ポリシーは単純な形式であらわせる。