A Method of Software-Hardware Integration for QoS Policy Combination in Gigabit Routers

Yasusi Kanada
Hitachi Ltd., Systems Development Laboratory
Takeki Yazaki
Hitachi Ltd., Central Research Laboratory

Introduction to Policy-based Networking

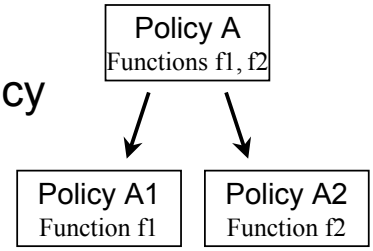
- **What is policy-based Networking?**
  - Network node configurations are vendor- and/or device-specific.
  - Policy-based networking replaces such configuration methods by a unified (and standard-based) method.
- **What is a policy?**
  - Policy rule: a condition-action rule
    - if *condition* then *action*
  - Policy: a list of policy rules.
    - { $rule_1$, $rule_2$, …, $rule_n$ }

Problem: Complexity of Policy Transformation

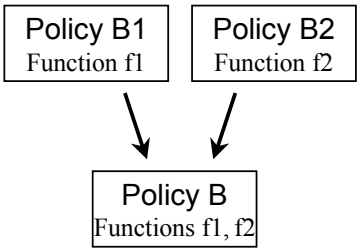
- **Human operators handle *high-level* policies.**
- **"*Low-level* policies" must be deployed to network nodes.**
- **Transformation from high- to low-level policies may be complex; i.e., it is not necessarily one-to-one.**
  - Routers, especially high-performance routers, require specific forms of policies (commands).

Transformation Types: Policy Division and Fusion

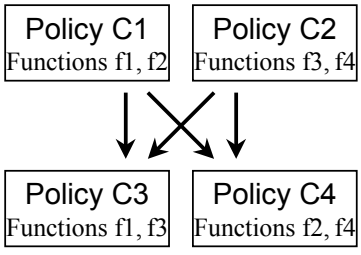
- **Policy division**
  - A transformation of a high-level policy into two or more low-level policies.



- **Policy fusion**
  - A transformation of two or more high-level policies into one low-level policy.



- **A combination of policy division and fusion**



Policy Division: Example

- **Input: Marking and priority queuing policy for Diffserv**
  - EC2 = { if (Source_IP is 192.168.1.1) {
    DSCP = "EF"; Priority = "High"; },
    if (true) {
    DSCP = "BE"; Priority = "Low"; }
    }.
- **Output: Marking policy and queuing policy**
  - E2 = { if (Source_IP is 192.168.1.1) { DSCP = "EF"; },
    if (true) { DSCP = "BE"; }
    }.
    C2 = { if (Source_IP is 192.168.1.1) { Priority = "High"; },
    if (true) { Priority = "Low"; }
    }.

Each rule is divided into two rules.

Policy Division: Example (cont'd)

- **Input: Marking and priority queuing policy for Diffserv**
  - EC2 = { if (Source_IP is 192.168.1.1) {
    DSCP = "EF"; Priority = "High"; },
    if (true) {
    DSCP = "BE"; Priority = "Low"; }
    }.
- **Output: Marking policy and queuing policy**
  - E2 = { if (Source_IP is 192.168.1.1) { DSCP = "EF"; },
    if (true) { DSCP = "BE"; }
    }.
    C2 = { if (Source_IP is 192.168.1.1) { Priority = "High"; },
    if (true) { Priority = "Low"; }
    }.

Conditions are copied.

Policy Division: Example (cont'd)

- **Input: Marking and priority queuing policy for Diffserv**
  - EC2 = { if (Source_IP is 192.168.1.1) {
    DSCP = "EF"; Priority = "High"; },
    if (true) {
    DSCP = "BE"; Priority = "Low"; }
    }.

Marking action

Queuing action

- **Output: Marking policy and queuing policy**
  - E2 = { if (Source_IP is 192.168.1.1) { DSCP = "EF"; },
    if (true) { DSCP = "BE"; }
    }.
    C2 = { if (Source_IP is 192.168.1.1) { Priority = "High"; },
    if (true) { Priority = "Low"; }
    }.

Marking actions

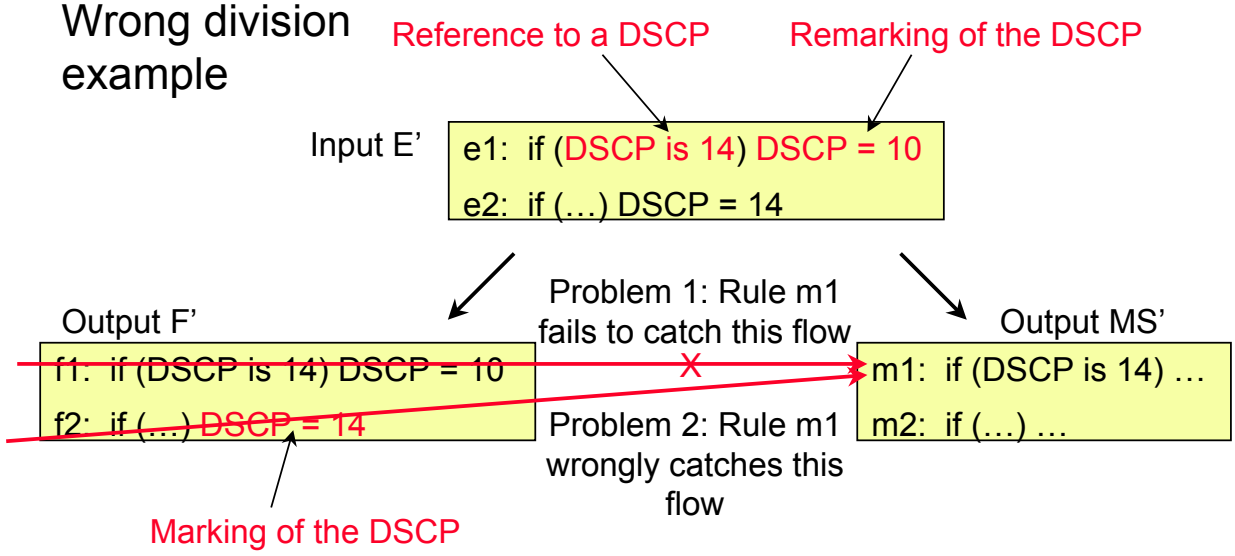
Queuing actions

Actions are divided.

How complex?: Restrictions on Policy Division

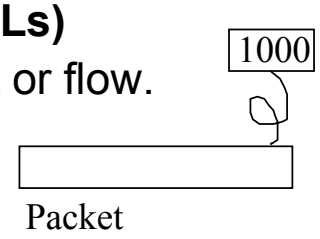
- **Restrictions on data reference and marking**
  - The naive transformation must be inhibited
    - if rules in the high-level policy refer to field in the packet, and
    - if this rule or another rule writes the same field

Wrong division example

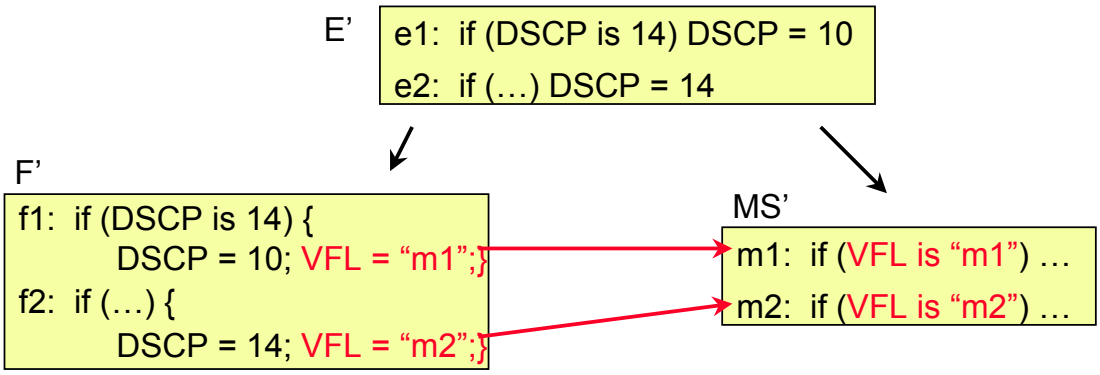


Elimination of the restrictions by using VFLs

- **Introduction of virtual flow labels (VFLs)**
  - A VFL is a label attached to a packet or flow.
  - A VFL is similar to a DSCP but it exists outside the packet.

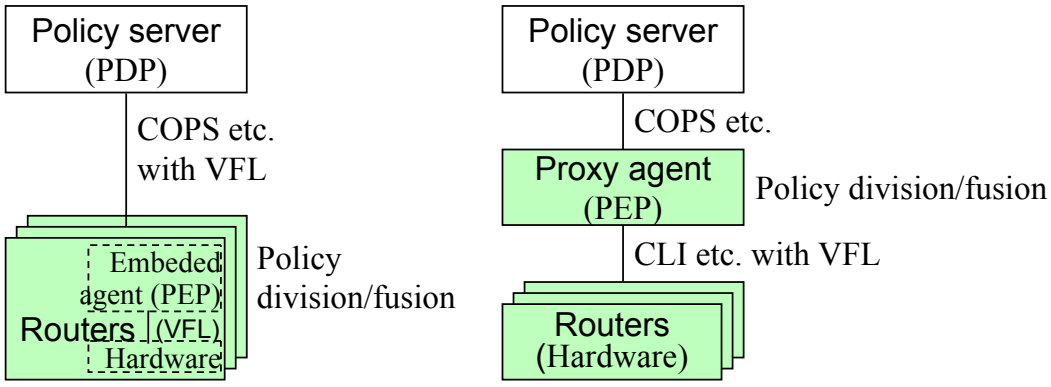


- **Policy division using VFLs**
  - The restrictions can be eliminated by introducing VFLs in a policy division. (See [Kan 01b] for detail.)



Method of Software-Hardware Integration for Policy-based QoS

- **Restrictions of policy division can be resolved by a software-hardware integration.**
  - Hardware-based VFLs (called flow IDs) are introduced into routers.
  - Policy division with VFLs are implemented in policy agents.

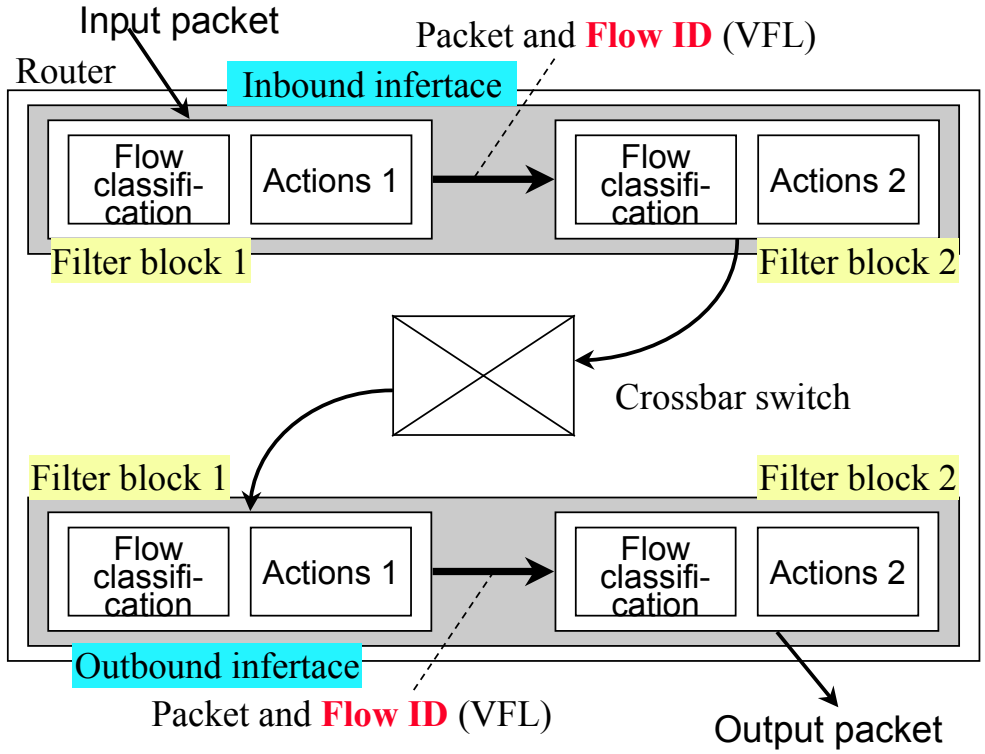


Prototype Development for Diffserv Policies

- **Diffserv policies in PolicyXpert™ were implemented for a gigabit router.**
  - PolicyXpert™ is a QoS policy server developed by Hewlett Packard and Hitachi.
- **Diffserv policies in PolicyXpert™ sometimes require policy division and/or fusion.**
  - These transformations enables flexible use of Diffserv policies. (not strictly necessary)
- **The restrictions are going to be eliminated by a software-hardware integration.**
  - VFLs (called flow IDs) were implemented by hardware.
  - A policy agent that use flow IDs is going to be developed.

A VFL Function Implemented in Hardware

- **Two filter blocks and flow IDs (VFLs)**



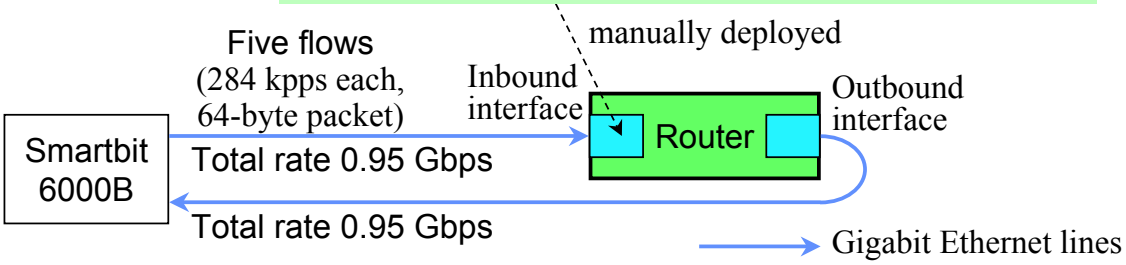
Policy Transformation for the Elimination

- **Instead of copying conditions, flow IDs are used.**
- **Example**
  - Input
    - EC2 = { if (Source_IP is 192.168.1.1) {
      DSCP = "EF"; Priority = "High"; },
      if (true) {
      DSCP = "BE"; Priority = "Low"; } }.
  - Output
    - E2' = { if (Source_IP is 192.168.1.1) {
      Flow_ID = "EF_FID"; DSCP = "EF"; },
      if (true) {
      Flow_ID = "BE_FID"; DSCP = "BE"; } }.
      - In addition to DSCP, flow IDs are set.
    - C2' = { if (Flow_ID is "EF_FID") { Priority = "High"; },
      if (Flow_ID is "BE_FID") { Priority = "Low"; } }.
      - Instead of copying the conditions, flow-ID conditions are introduced.

Evaluation

- **Performance of marking rules was measured.**
- **Method**

Policy F: 100 aggregation rules, the flows hit the 10th, 30th, 50th, 70th and 90th rules.
Policy S: 5 marking rules.
Five flow IDs connect rules in F and S.



- **Result**
  - The total input and output rates were both measured to be 1.42 Mpps, i.e., no performance degradation occurred.
  - This means introduction of VFLs does not degrade the performance.

Conclusion

- We have developed a method of software-hardware integration for resolving the restrictions of policy division.
- We are developing a policy agent and a gigabit router integrated by using this method to support the Diffserv policies of PolicyXpert.
- A preliminary evaluation result shows that both high-performance and flexibility are achieved by this integration.

Policy Division: Example 2

- Input: a rule with flow aggregation
  - E3 = { if (Source_IP is 192.168.1.1 || Source_IP is 192.168.1.3) {
    if (Information_Rate <= 1 Mbps) {
    DSCP = "EF";
    } else { absolute_drop; }; }
    }.

This rule aggregates two flows

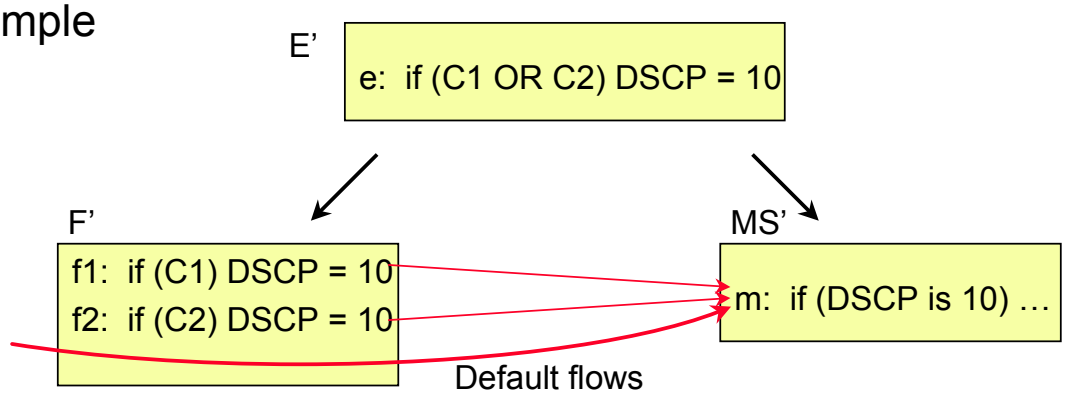
- Output
  - E31 = {if (Source_IP is 192.168.1.1) { DSCP = "EF"; },
    if (Source_IP is 192.168.1.3) { DSCP = "EF"; } },
    E32 = {if (DSCP is "EF") {
    if (Information_Rate > 1 Mbps) {
    absolute_drop; }; }
    }.

Restrictions on Policy Division (cont'd)

Restrictions on flow aggregation

- If specific data is used for identifying an aggregated flow, flows that are not caught by any rule in F' (called default flows) must be inhibited.

Example



Policy Transformation for Resolution (cont'd)

Example 2

- Input
  - E3 = { if (Source_IP is 192.168.1.1 ||
    Source_IP is 192.168.1.3) {
    if (Information_Rate <= 1 Mbps) {
    DSCP = "EF";
    } else { absolute_drop; }; }
    }.
- Output
  - E31 = { if (Source_IP is 192.168.1.1) { Flow_ID = "EF_FID"; },
    if (Source_IP is 192.168.1.3) { Flow_ID = "EF_FID"; }
    }.
    - A flow ID is used twice.
    - No need to set DSCP here. (This transformation is simpler.)
  - E32 = { if (Flow_ID = "EF_FID") {
    if (Information_Rate <= 1 Mbps) {
    DSCP = "EF";
    } else { absolute_drop; }; }