Policy Division and Fusion: Examples and A Method

Yasusi Kanada
Hitachi Ltd., IP Network Research Center
Japan

This presentation consists of:

1. Introduction to policy division and fusion
2. Examples and problems in Diffserv
3. A Method of policy division and fusion
4. Resolution of the problems

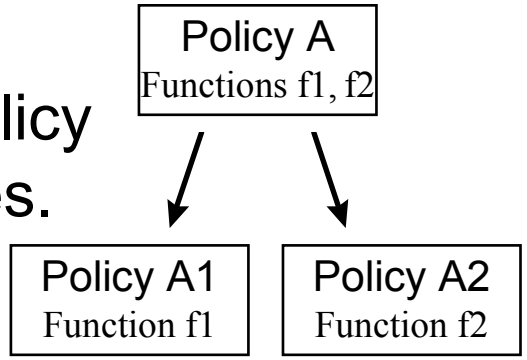
1. Introduction to policy division and fusion

Introduction

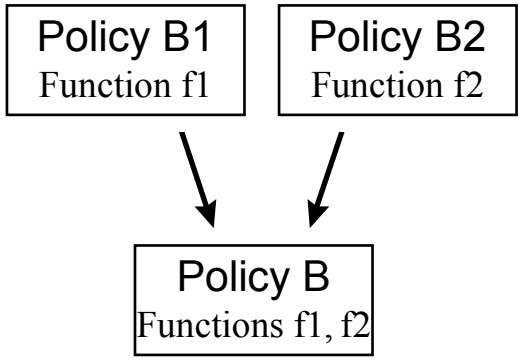
- **In a policy-based network**
  - A policy server deploys policies to network nodes.
  - Policies may work in cooperation.
    - E.g., in Diffserv, marking and queuing/scheduling policies — the latters depend on the formers.
- **Higher-level (HL) and lower-level (LL) policies**
  - A “Policy” means a list of condition-action rules.
  - Both a policy server and network nodes work with policies.
    - LL commands for a network interface (e.g., ACLs) form a *LL policy*.
    - A policy server has *HL policies*.
  - HL policies must be translated into LL polcies.
    - This translation is similar to compilation of programming languages such as C/C++.
    - This process is much more complex; the correspondence between HL and LL policies is not one-to-one.

What are policy division and fusion?

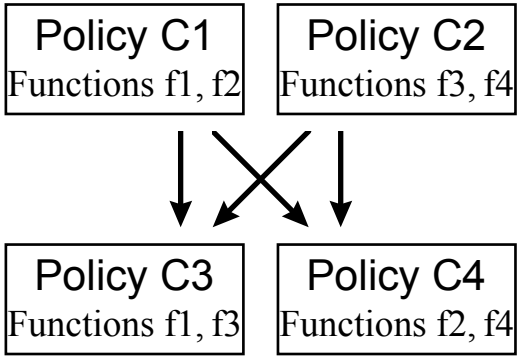
- **Policy division**
  - Transformation of a HL policy into two or more LL policies.



- **Policy fusion**
  - Tansformation of two or more HL policies into a LL policy.



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



Simple policy division

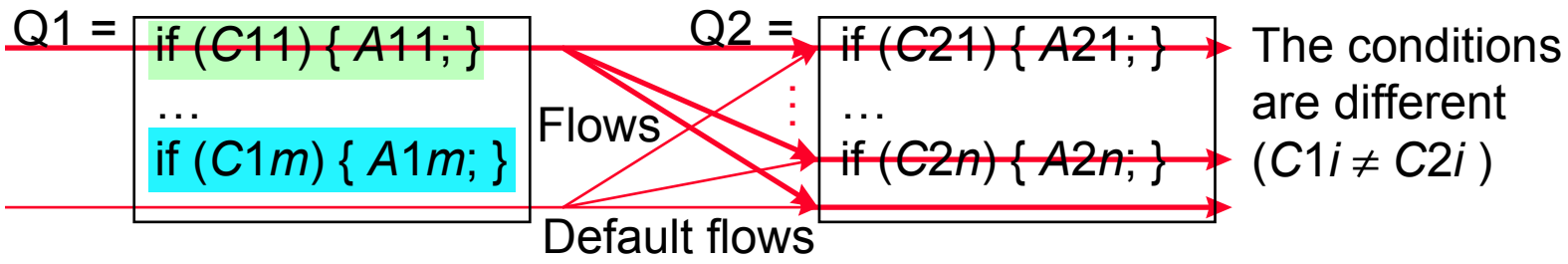
- **Input:** a HL policy P
  - P = { …, if (*Ci*) { *Ai*1; …; *Ain*; }, … }
    — a rule with *n* actions (various types of actions in a policy).
- **Outputs:** LL policies P1, …, P*n*
  - P1 = { …, if (*Ci*) { *Ai*1; }, … }, — limited types of actions in a policy.
    …,
    P*n* = { …, if (*Ci*) { *Ain*; }, … }.
- **Notes**
  - Every rule in P is divided into *n* rules.
  - The number of rules in P, P1, …, P*n* are the same.

Simple policy fusion (Case 1)

- **Inputs:** HL policies P1, …, P*n*
  - P1 = { …, if (*Ci*) { *Ai*1; }, … },
    …,
    P*n* = { …, if (*Ci*) { *Ain*; }, … }.
- **Output:** a LL policy P
  - P = { …, if (*Ci*) { *Ai*1; …; *Ain*; }, … } — a rule with *n* actions
- **Note**
  - The number of rules in P1, …, P*n*, and P are the same.
  - *I*-th rules of P1, …, P*n* have the same condition *Ci* — a very rare case.

Simple policy fusion (Case 2)

◆ **Inputs:** HL policies Q1 and Q2



◆ **Output:** a LL policy Q

Q = { if (C11 AND C21) { A11; A21; },
…,
if (C11 AND C2n) { A11; A2n; },
if (C11) { A11; }, — n+1 rules
…,
if (C1m AND C21) { A1m; A21; },
…,
if (C1m AND C2n) { A1m; A2n; },
if (C1m) { A1m; }, — n+1 rules
if (C21) { A21; },
…,
if (C2n) { A2n; } }.

The number of rules in Q:
(m+1)(n+1) – 1 — *too many!!*

2. Examples and problems in Diffserv

Example policy types for Diffserv

■ Definition of HL policy types

◆ Edge policy

- A type of policy that *classifies*, *meters*, and/or *marks* packets.
- E.g., if (Source_IP == 192.168.0.1) {
  if (Information_rate < 1M bps) { DSCP = 46; } else { drop; };
  }

◆ Core policy

- A type of policy that *queues*, *schedules*, and *randomly drops* packets.
- E.g., if (DSCP == 46) {
  Queue_number = 6;
  Scheduling_algorithm = "Priority_scheduling";
  }

Example policy types for Diffserv (cont'd)

- **Each LL policy type may correspond to a command.**
- **Definition of LL policy types**
  - Filtering policy
    - A type of policy that *classifies*, *filters*, and/or *marks* a DSCP on packets.
    - E.g., if (Source_IP == 192.168.0.1) { DSCP = 46; },
  - Metering and scheduling policy
    - A type of policy that *meters*, *queues*, and/or *schedules* packets.
    - E.g., if (DSCP == 46) {
      if (Information_rate < 1M bps) {
      Queue_number = 6;
      Scheduling_algorithm = "Priority_scheduling";
      } else {
      drop; }; }.
- **A restriction on the LL policies**
  - The conditions may not contain "OR" (flow aggregation).
    - if (… OR …) { … } — not allowed (must be divided into two rules)

Why are policy division/fusion required?

- **Because the functional correspondence between HL and LL policies is not one-to-one.**
  - E.g., Policy A has function f1 and f2, but policy A1 only has f1 and policy A2 only has f2.
- **Because of functional restrictions of the network nodes.**
  - Policy servers should implement HL policies that are standardized and device-independent.
  - LL policies may be restrictive because they may be implemented in hardware, or high performance is required.
    - The LL policies and the policy divison/fusion are implemented in the PolicyXpert agent for Hitachi's gigabit router GR2000.

Policy division example 1: with metering

- This example is similar to the simple one given before.
- **Input:** edge policy E
  - E = {…,
    if (Source_IP == a*i*) {
      if (Information_rate < 1M bps) { DSCP = d*i*; } else { drop; }; },
    …
    }.

Metering action **Marking action**

- **Outputs:** filtering policy F and metering and scheduling policy MS
  - F = { …,
    if (Source_IP == a*i*) { DSCP = d*i*; },
    …
    },
  - MS = { …,
    if (Source_IP == a*i*) {
      if (Information_rate < 1M bps) {} else { drop; }; },
    …
    }.

Marking action

Metering action

Policy division example 2: with aggregation

- **Input:** edge policy E'
  - E' =

```
{ if ( Source_IP == a1 OR
    … OR
    Source_IP == an) {          — The flows are aggregated here.
    if (Information_rate < 1M bps) { DSCP = 10; } else { drop; }; }
}.
```

- **Outputs:** filtering policy F' and metering and scheduling policy MS'
  - F' =

```
{ if (Source_IP == a1)  { DSCP = 10; },
  …,
  if (Source_IP == an)  { DSCP = 10; }  }.
```

— The flows are aggregated by marking DSCP 10.

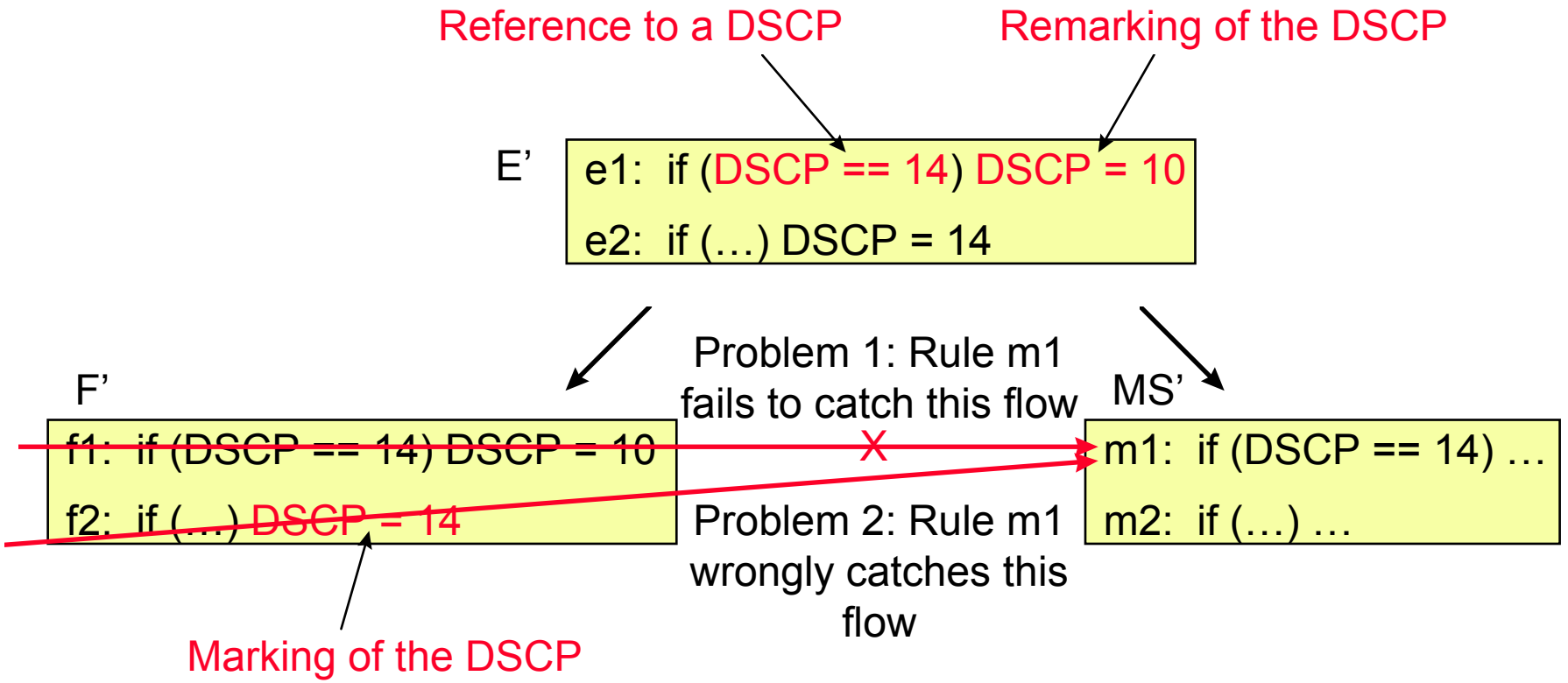
  - MS' =

```
{ if (DSCP == 10) {
    if (Information_rate < 1M bps) {} else { drop; }; }  }.
```

Restrictions on policy division

■ Restrictions on DSCP reference and marking

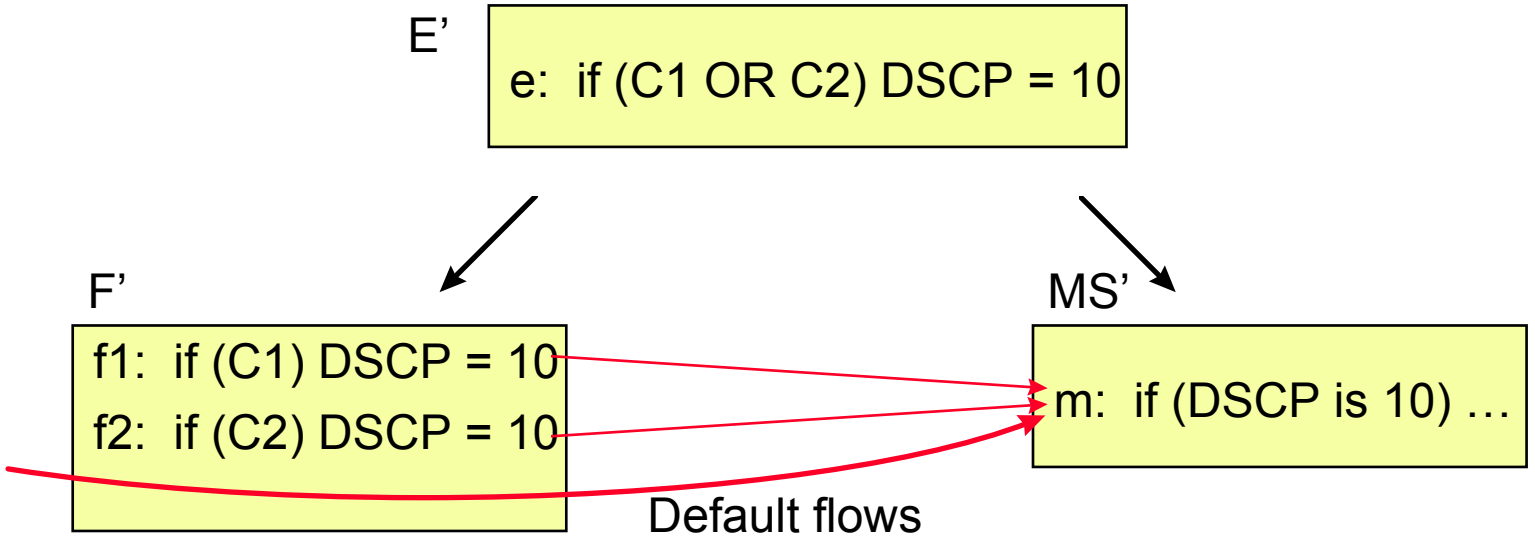
- ◆ If rules in the HL policy refer to a DSCP, and the rule remarks or another rule marks the DSCP, the naive transformation must be inhibited.



Restrictions on policy division (cont'd)

Restrictions on flow aggregation

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



Policy fusion example: typical Diffserv policies

- This example is similar to the Case 1.
- **Inputs:** edge policy E1 and core policy C1
  - E1 = { …,
    if (Source_IP == a*i*) {
    if (Information_rate < 1M bps) {DSCP = d*i*;} else {drop;}; },
    …
    }.
  - C1 = { …,
    if (DSCP == d*i*) {
    Scheduling_algorithm = "WRR"; Min_BW = 512 kbps; },
    …
    }.

Metering and marking actions

Conditions are different

Scheduling action

- **Output:** Metering and scheduling policy MS1
  - MS1 = { …,
    if (Source_IP == a*i*) {
    Scheduling_algorithm = "WRR"; Min_BW = 512 kbps;
    if (Information_rate < 1M bps) {DSCP = d*i*;} else { drop; }; },
    …
    }.

Scheduling action

Metering and marking actions

Restrictions in policy fusion

- **The transformation in Case 2 is not restricted, but practically unacceptable because it generates too many rules.**
- **In the transformation in the Diffserv example (Case 1'), the following conditions are required not to increase the number of rules:**
  - Each rule in HL policy E1 must mark or check a DSCP.
    - E.g., if (…) DSCP = …; — marking action
      if (DSCP == …) …; — checking condition
  - A default flow may not exist.
    - The following rule may have to be added:
      if (true) drop;

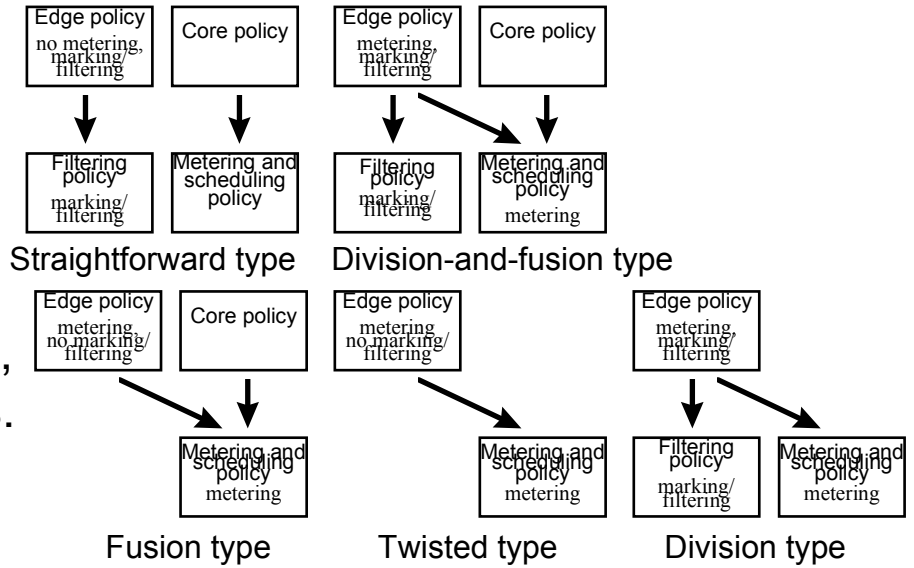
3. A Method of policy division and fusion

A method of policy division and fusion

- **Restrictions (to avoid complexity)**
  - This method only applies to the edge and core policies.
  - All the restrictions described before (may) apply.
  - The core policy must have DSCP-only conditions (a BA classifier).
  - See details for the paper.

A method of policy division and fusion (cont'd)

■ Outline of the algorithm



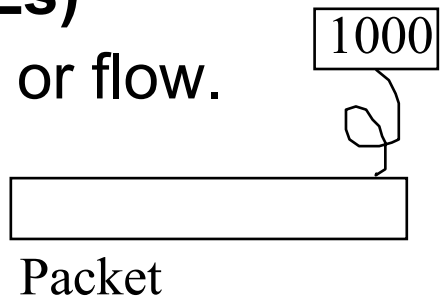
- ◆ Edge policy pass 1
  - ▮ The transformation type (TT) is determined.
    - – Five TTs: straightforward, division-and-fusion, fusion, twisted, and division types.
- ◆ Core policy pass
  - ▮ If the TT is the straightforward type, a metering and scheduling policy is generated from the core policy.
  - ▮ Otherwise, a core policy table (DSCP-to-action mapping table) is created.
- ◆ Edge policy pass 2
  - ▮ The HL policies are transformed into LL policies according to the TT.

4. Resolution of the problems

Resolution of the problems

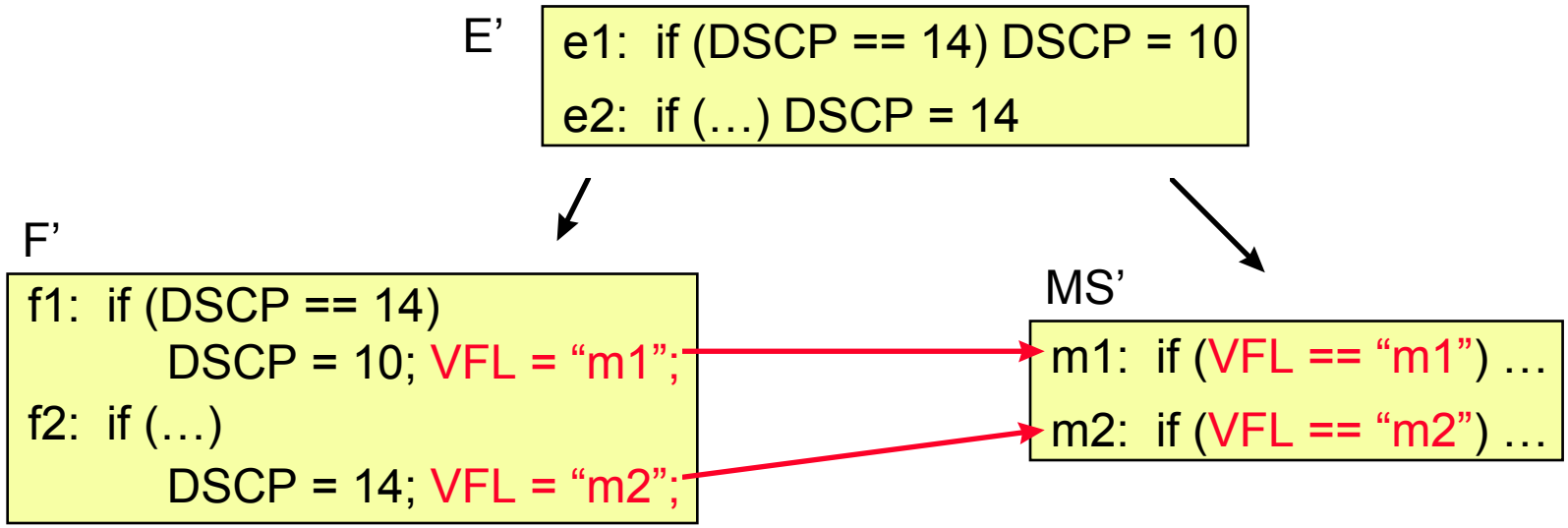
■ Introduction of virtual flow labels (VFLs)

- ◆ 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.



Conclusion

Conclusion

- **If the forms of HL policies are properly constrained, they can be translated into LL policies automatically by using policy division/fusion.**
- **However, policy division/fusion should be avoided if possible because**
  - the forms of HL policies are ristricted, and
  - the transformation may be too much complicated.
- **The ristrictions on policy division can be eliminated by introducing VFLs.**