Distributed Policy-based Management Enabling Policy Adaptation on Monitoring using Active Network Technology

Kiyohito Yoshihara, Manabu Isomura and Hiroki Horiuchi

KDDI R&D Laboratories Inc.
2-1-15 *Ohara Kamifukuoka-shi Saitama* 356-8502, *JAPAN*

In policy-based management, in addition to deliver and enforce policies in managed systems, it is inevitable to manage policy life-cycle. We mean the policy life-cycle as a sequence of processes involving monitoring to see if the enforced policies actually work at operators' will and adapting them based on monitoring. However, enabling such policy life-cycle management by the current centralized management paradigm such as SNMP may result in poor scalability and reliability. This is typically due to much bandwidth consumption for monitoring and communication failure between a manager and an agent. It may also burden the operators with a heavy load in analyzing management information for the policy adaptation. For a solution to that, we propose a scalable and reliable policy-based management scheme enabling the policy life-cycle management using the active network technology. In the scheme, we provide a new management script describing policies and also how their life-cycle should be managed, and execute the script on the managed systems called active nodes. The scheme can make the current policy-based management more scalable by reducing management traffic, more reliable by distributing management tasks to the managed systems, and more promising by alleviating the operators' burden. We implement a prototype system based on the scheme adopting Differentiated Services as a policy enforcement mechanism, and evaluate the scheme from the following viewpoints: the advantage of policy adaptation on monitoring, the amount of management traffic required and the load on the managed systems executing the management scripts. We also discuss how the prototype system could be integrated with managed systems compliant with standards emerging in marketplace.

Keywords: policy-based management, active network, policy life-cycle management, Differentiated Services

1 Introduction

As commercial and enterprise applications gain widespread use in the Internet, it is getting an urgent need to make the most of limited network resources to guarantee required QoS (Quality of Services) of applications tailored to customers. For a solution to this, policy-based management has recently been developed and standardized by some organizations such as IETF (Internet Engineering Task Force) [IETa, IETb] and DMTF (Distributed Management Task Force) [DMT].

In the policy-based management, in addition to deliver and enforce policies in managed systems such as routers and switches, it is inevitable to manage policy life-cycle. The policy life-cycle means a sequence of processes involving monitoring to see if the enforced policies actually work at operators' will and adapting them based on the monitoring. However, enabling such policy life-cycle management by the current centralized management paradigm such as SNMP (Simple Network Management Protocol) may result in poor scalability and reliability. This is typically due to much bandwidth consumption for monitoring and communication failure between a manager and an agent. It may also burden the operators with a heavy load in analyzing management information for the policy adaptation. Toward more promising policy-based management, how to manage the policy life-cycle in a scalable, reliable and less laborious manner is, therefore, one of the critical challenges.

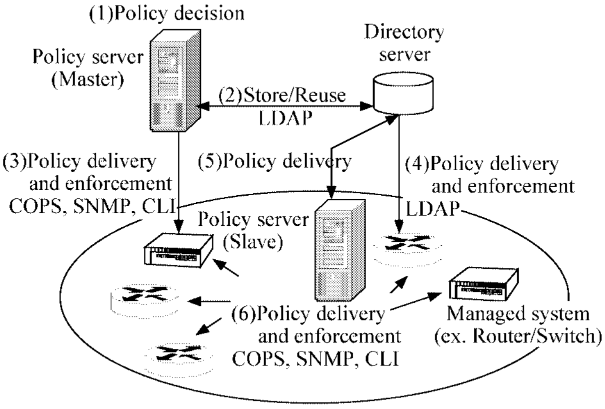

(1)Policy decision
Policy server (Master)
Directory server
(2)Store/Reuse LDAP
(3)Policy delivery and enforcement COPS, SNMP, CLI
(5)Policy delivery
(4)Policy delivery and enforcement LDAP
Policy server (Slave)
(6)Policy delivery and enforcement COPS, SNMP, CLI
Managed system (ex. Router/Switch)

Fig. 1: Framework of current policy-based management.

In this paper, in the context of IETF policy-based management, we propose a new policy-based management scheme enabling the policy life-cycle management. In the scheme, we introduce a new management script describing not only policies but also how their life-cycle should be managed. The script is executed on the managed system assumed to have enough computation resources, as well as in the active network technology [Pso99, RS00], the distributed management paradigm such as MbD (Management by Delegation) [YGY91, GY98], and the management by mobile software agents [ZHG99, GGGO00]. By executing the script on the managed system, the scheme can make the current policy-based management more scalable by reducing management traffic, more reliable by distributing management tasks to the managed systems, and more promising by alleviating the operators' burden. The strength of the paper lies not only in proposing the novel scheme but also in showing how to apply the scheme to a state-of-art policy enforcement mechanism. We implement a prototype system based on the scheme adopting Differentiated Services [BBC$^+$98] as the policy enforcement mechanism. We evaluate the scheme from the viewpoints of: 1) the advantage of the policy adaptation on monitoring, 2) the reduced amount of management traffic required, and 3) the computational resources of a managed system required in the proposed scheme. We also discuss how the prototype system could be integrated with managed systems compliant with the standards being developed.

This paper is organized as follows: In Section 2, we present an overview of the current policy-based management and Differentiated Services. In Section 3, we describe the need for policy life-cycle management and address an issue for the realization. For a solution to the issue, in Section 4, we propose a new policy-based management scheme. In Section 5, we implement a prototype system and, in Section 6, we evaluate the scheme by applying the prototype system to an operational network.

2 Overview of Current Policy-based Management and Differentiated Services

2.1 Overview of Current Policy-based Management

IETF [IETa] and DMTF [DMT] jointly define information model for specifying a policy. IETF [IETa, IETb] also defines a framework of the policy-based management and protocols for policy delivery and enforcement.

In the context of IETF policy-based management, a policy consists of a condition clause and an action clause applied only if the condition clause is evaluated to be true. There are two classes of policies with respect to their purposes: One is of QoS policies for priority and bandwidth control including priority queuing and packet shaping. Another is of security policies for access control including packet filters on a firewall machine and a WWW server. In this paper, we focus on the QoS policies hereafter.

Figure1 shows a framework of the current policy-based management defined by IETF. A network operator, first, decides a policy to be enforced on a managed system such as a router and a switch (Fig.1(1)). The policy can be stored in a directory server and may be reused using LDAP (Lightweight Directory Access

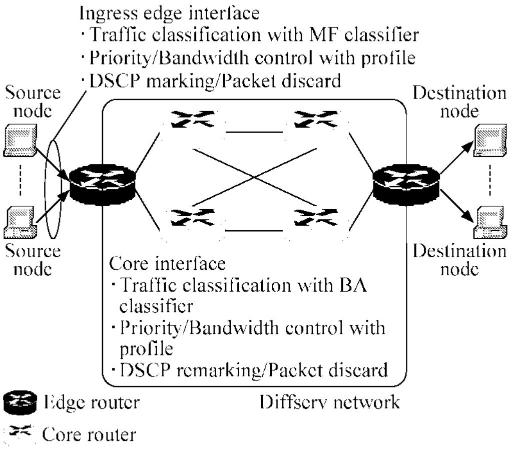

Ingress edge interface
· Traffic classification with MF classifier
· Priority/Bandwidth control with profile
· DSCP marking/Packet discard
Source node
Source node
Destination node
Destination node
Core interface
· Traffic classification with BA classifier
· Priority/Bandwidth control with profile
· DSCP remarking/Packet discard
Edge router
Core router
Diffserv network

Fig. 2: Overview of Differentiated Services.

(a)Policies at ingress edge interface

Policy	MF classifier	Profile		DSCP marked	
		Peak rate	Peak burst size	In-profile	Out-of-profile
A	SrcIP==10.0.0.0/24	2Mbps	2Kbyte	101110	000000
B	Otherwise	None	None	000000	000000

(b)Policies at core interface

Policy	BA classifier	Profile		DSCP remarked	
		Peak rate	Peak burst size	In-profile	Out-of-profile
C	101110	5Mbps	2Kbyte	101110	Discard
D	Otherwise	None	None	000000	000000

```
/*Policy A is interpreted as follows*/
if(SrcIP==10.0.0.0/24) then{
    if(Peak rate<=2Mbps && Peak burst size <=2Kbyte) /*Profile*/
    then{DSCP=101110}  /*In-profile*/
    else{DSCP=000000}} /*Out-of-profile*/
/*Policy C is interpreted as follows*/
if(DSCP==101110) then{
    if(Peak rate<=5Mbps && Peak burst size <=2Kbyte) /*Profile*/
    then{DSCP=101110}  /*In-profile*/
    else{Discard}}          /*Out-of-profile*/
```

Fig. 3: Example of policies for Diffserv.

Protocol) (Fig.1(2)). When the operator enforces the policy on the managed system, the policy is delivered and actually enforced using COPS (Common Open Policy Service) [CSD$^+$01], SNMP, or a vendor-specific CLI (Command Line Interface) (Fig.1(3)). It is also possible to deliver and enforce a policy directly from the directory server (Fig.1(4)), or by way of a slave policy server (Fig.1(5)) located in order for load sharing or fault tolerance among the policy servers (Fig.1(6)).

2.2 Overview of Differentiated Services

Diffserv (Differentiated Services) [BBC$^+$98] is one of the promising policy enforcement mechanisms standardized by IETF. Figure2 shows an overview of Diffserv. Figure3 gives examples of policies for Diffserv that we consider throughout the paper.

At each ingress edge interface, ingress traffic is classified by means of MF (Multi-Field) classifier specifying one or more key-value pairs in a packet, such as source/destination IP addresses and source/destination port numbers. For each classified traffic, a profile providing a rule for determining whether a particular packet is in-profile or out-of-profile, or how the packet should be prioritized or controlled, is applied. For example, a profile in the form of simple token bucket may specify a peak rate and a peak burst size. Depending on the result, the packet is marked with a 6-bit DSCP (Diffserv Code Point), which represents a forwarding treatment of the packet in the Diffserv network, in the IPv4 Type of Service octet or the IPv6 Traffic Class octet. Some out-of-profile packets may be discarded without marking.

At each core interface, the traffic is classified by means of BA (Behavior Aggregate) classifier specifying a DSCP. For each classified traffic, as well as at the ingress edge interface, a profile is applied. Some packets are remarked with another DSCP and others may be discarded depending on the result.

The policy A in Fig.3 classifies the packets with their Source IP Address (SrcIP) 10.0.0.0/24 from others. Each packet in in-profile is marked with the DSCP "101110" representing such a forwarding treatment that the packet is sent with no loss and less delay, called EF (Expedited Forwarding). Other packets in out-of-profile is marked with the DSCP "000000" representing best-effort forwarding treatment.

3 Need for Policy Life-cycle Management

When a network operator tries to decide and enforce a new policy, it is significant to see dynamic nature of network utilization, in addition to the operator's knowledge and experience. Besides, due to the increase in the number of users and traffic, and the deployment of new applications, a policy being enforced does not necessarily work as intended for a long time in general.

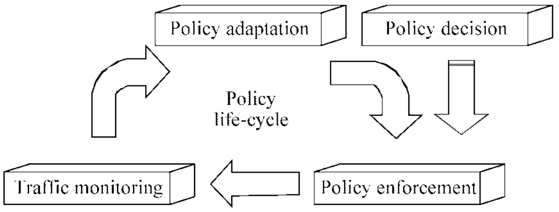


Fig. 4: Policy life-cycle in policy-based management.

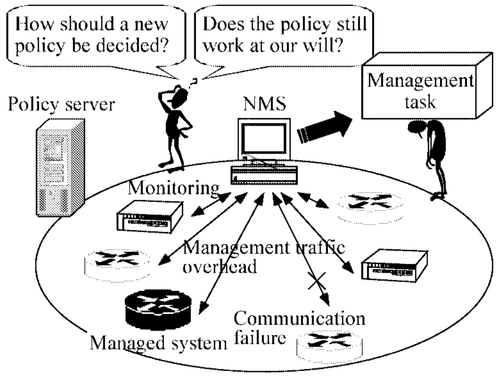

How should a new policy be decided?
Does the policy still work at our will?
Management task
Policy server
NMS
Monitoring
Management traffic overhead
Communication failure
Managed system

Fig. 5: Policy life-cycle management with current products.

The purpose of Diffserv is to establish peak boundaries on traffic values and the boundaries are given in the static forms. So, in order to make the most of limited network resources, a mechanism for adapting and optimizing the boundaries according to the dynamic nature of network utilization is required. Such a requirement is also stated in some early literatures [Wie94, Goh98]. In the policy-based management, therefore, it is inevitable to manage policy life-cycle management. We mean the policy life-cycle as a sequence of processes involving: 1) policy decision and enforcement, 2) traffic monitoring to see if the enforced policy works at operators' will, and 3) policy adaptation (updating the condition and action clauses) based on the monitoring, as shown in Fig.4.

The current products such as policy servers are, however, only capable of delivering a policy and enforcing it on a targeted managed system. If the operator tries to achieve the policy life-cycle management by means of a conventional centralized NMS (Network Management System), this might result in lack of scalability and reliability typically caused by management traffic overhead for the traffic monitoring, and communication failure between a manager and an agent. It may also bring a management task on the operator in analyzing management information for the policy adaptation as shown in Fig.5. RMON (Remote Network Monitoring) is not sufficient either, since the traffic statistics associated with each managed system, such as the number of the degraded and discarded packets at some priority queue of a specific interface caused by a policy, is rather required, than those associated with an entire network obtained by RMON.

Therefore, toward more promising policy-based management, how to manage the policy life-cycle in a scalable, reliable and less laborious manner is one of the critical challenges. For a solution to this, in Section 4, we propose a distributed policy-based management scheme enabling the policy adaptation on monitoring.

4 Proposal on Distributed Policy-based Management Scheme Enabling Policy Adaptation on Monitoring

4.1 Principle

1. For the purpose of policy life-cycle management, we introduce a new management script describing not only policies but also how to manage their life-cycle. As shown in Fig.6, it basically consists of the following three parts each mapped to the component of the policy life-cycle: the policy enforcement part, the traffic monitoring part, and the policy adaptation part.

2. For the purpose of scalability and reliability, we execute the management script on the managed system having enough computational resources, as well as in the active network technology [Pso99, RS00], the distributed management paradigm such as MbD (Management by Delegation) [YGY91, GY98], and the management by mobile software agents [ZHG99, GGGO00]. If we cannot expect such a managed system, the management script should be executed on a management system or a surrogate host. The surrogate host provides enough resources to execute the management script on

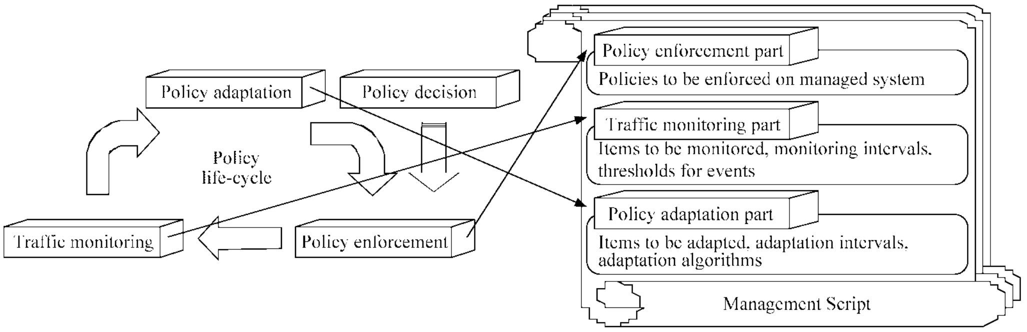
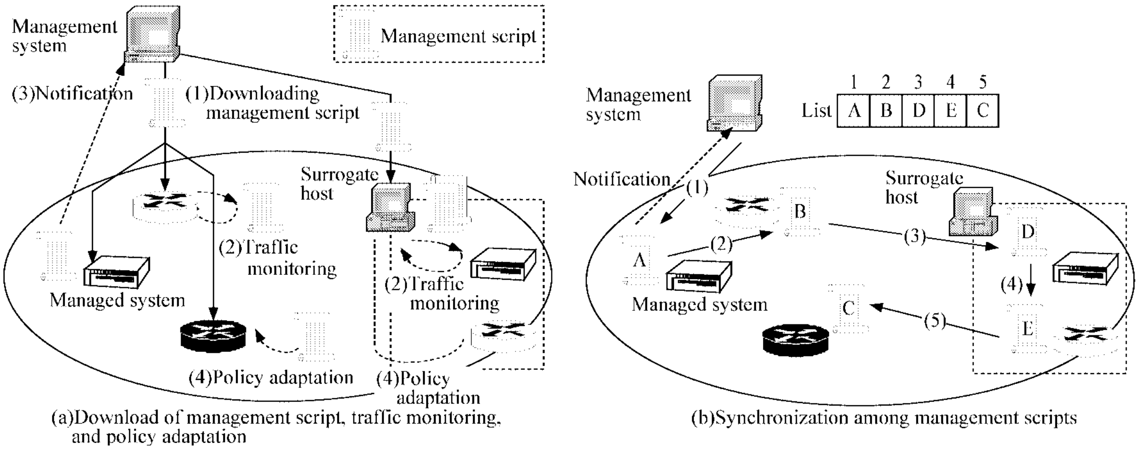


Fig. 6: Management script in proposed scheme.



(a)Download of management script, traffic monitoring, and policy adaptation

(b)Synchronization among management scripts

Fig. 7: Proposed scheme.

behalf of a managed system and a facility to communicate with the managed system in a standardized or proprietary manner.

4.2 Proposed Scheme

Figure7 shows a diagrammatic representation of the proposed scheme. We describe how the proposed scheme can achieve the scalable and reliable policy-based management enabling the policy adaptation on monitoring below.

4.2.1 Policy Enforcement by Downloading Management Scripts

For the purpose of enforcing consistent policies in an entire management domain, the management system downloads management scripts to all managed systems and surrogate hosts simultaneously (Fig.7(a)(1)). If all the downloads have completed successfully, the management system executes the management scripts on the managed systems and surrogate hosts. Otherwise, or if any of downloads have failed, the management system rollbacks the execution context as it was. On executing, each management script enforces the policies specified in the policy enforcement part.

4.2.2 Traffic Monitoring for Policy Adaptation

For the purpose of monitoring to see if the enforced policies actually work as intended, items to be monitored associated with the policies, monitoring intervals, and thresholds for events are provided in the management script. Each management script monitors the items with the given monitoring intervals (Fig.7(a)(2)). When a threshold violation occurs, the management script notifies the management system of the event (Fig.7(a)(3)). The management system may use this event as a trigger for a policy adaptation.

Some examples of the items to be monitored are the number of packets degraded the queuing priority or discarded by the enforcement of the policies. From the viewpoints of enforcing consistent policies in an

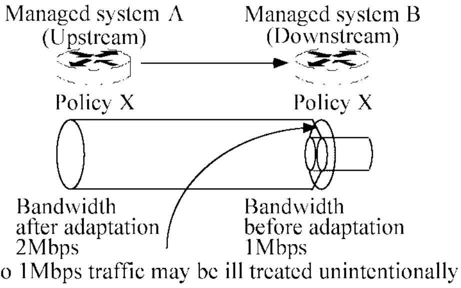


Fig. 8: Need for synchronization among management scripts.

```
/*@param addr    List of IP addresses of managed systems
 * @param ifName List of interface names of managed systems
 * @param pid    Identifier of a management script triggering
                 a policy adaptation
  @param target  List of items to be adapted */
public void syncRequest( Vector addr, Vector ifName,
                         String pid, Vector target)
throws RemoteException{
  syncCtrl( pid, target );
  // For synchronizing this management script
  . . . . .
  syncThread = new Thread(){
  public void run(){
  try {syncRequestNext( tmpAddr, tmpIfName, tmpPid,
                        tmpTarget);
    // For synchronizing the next management script
    } catch(Exception e){e.printStackTrace();}}};
  syncThread.start();}
```

Fig. 9: Skeletal description of `syncRequest` method.

entire management domain, it is ideally desirable and we assume that managed systems support common standard MIB (Management Information Base) like [IET01], allowing for providing a management script in a uniform way. If this MIB is available, we could set a counter at any point on the data path by setting an appropriate `DiffServCountActEntry`. We set this entry when describing the traffic monitoring part of a management script. For example, if the number of packets marked with a certain DSCP should be monitored, then a `diffServCountActPkts` variable in a `DiffServCountActEntry` located after an appropriate `DiffServMeterEntry` on a data path is specified.

It may be often the case where there exists a managed system supporting only a vendor-specific MIB. In order to cope with such a managed system, an additional function for mapping a vendor-specific MIB and a common standard MIB would be required.

4.2.3 Policy Adaptation by Management Script

For the purpose of adapting the policies based on the results of the monitoring, items to be adapted, an adaptation interval, and adaptation algorithms are provided in the management script. The items to be adapted could also be specified in terms of MIB variables as well as the items to be monitored. The adaptation algorithms should be pre-defined and be bounded with a management script in provisioning. The management script performs the policy adaptation based on the adaptation algorithms when a threshold violation occurs or for every adaptation interval (Fig.7(a)(4)).

The adaptation interval is provided in addition to the monitoring interval, in order to detect such a superfluous resource assignment in an initial policy decision and a previous policy adaptation. For example, if a policy assigns the maximum bandwidth, 2Mbps, to a particular traffic, while the actual amount is at most 1Mbps, this could not be detected only by threshold violation.

A typical adaptation algorithm derives the value of the item to be adapted from the weighted average, maximum or minimum value of the items monitored before and the previously derived value for the past adaptation, with some given constants such as the maximum length of a packet.

4.2.4 Synchronization among Management Scripts

When a management script executes a policy adaptation, it also notifies all the other management scripts of that event and synchronizes the policy, so that the policy could still be enforced consistently in an entire management domain. Without the synchronization, as an example shown in Fig.8, up to 1Mbps traffic might be ill treated unintentionally, since the bandwidth assigned to a particular traffic by the policy X on the managed system B is still unchanged, while the bandwidth assigned by the same policy X on the managed system A has been adapted and broadened from 1Mbps to 2Mbps.

For the synchronization, the management system retains information about which management script is executed on which managed system in the form of a list. When the management system receives an event triggered by a policy adaptation, it generates a message for calling the `syncRequest` method of a management script. Figure9 shows a skeletal description of the method. The management script called the method executes the `syncCtrl` method (the 4th line on Fig.9) for synchronizing itself by driving adaptation algorithms with locally monitored items. After that, the management script notifies the management system of the completion, updates the list, and calls the `syncRequestNext` method (the 9th line on Fig.9) for synchronizing the next management script on the list, resulting in sending a message for calling the `syncRequest` method of the next management script.

By iterating the above, all management scripts associated with a policy adaptation can synchronize one another. Figure7(b) shows how the message for calling the `syncRequest` method traverses with the given list as an example.

4.3 Application of Proposed Scheme to Differentiated Services

We show how the proposed scheme can be applied to Diffserv from the Diffserv-specific viewpoint of provisioning of a management script and policy adaptation. The profile is assumed to be in the form of simple token bucket below.

4.3.1 Management Script Provisioning for Diffserv

- Policy enforcement part

  For each MF and BA classifier, the following items are provided. A management script includes one or more policies associated with each classifier.

  - An MF classifier or a BA classifier
  - A profile (a peak information rate and peak burst size)
  - DSCPs to be marked or remarked for in-profile and out-of-profile packets (”discard” is also possible)

- Traffic monitoring part

  For each MF and BA classifier, the following one or more items are provided with their monitoring intervals and thresholds for notification.

  - The number of in-profile packets or bytes
  - The number of out-of-profile packets or bytes
  - The number of remarked packets or bytes caused by the unconformity to the given profile
  - The number of discarded packets or bytes caused by the unconformity to the given profile

- Policy adaptation part

  For each MF and BA classifier, the following one or more items are provided with their adaptation intervals and adaptation algorithms.

  - A peak information rate in a profile
  - A peak burst size in a profile
  - DSCPs to be marked or remarked for in-profile and out-of-profile packets (”discard” is also possible)

4.3.2 Policy Adaptation by Management Script in Diffserv

A number of adaptation algorithms, from simple ones to complex ones, may be possible. Below, we show three simple but practical adaptation algorithms each for the item to be adapted in Section 4.3.1, while an investigation of more suitable one is out of the scope of the paper and is left as a future work.

- A peak information rate in a profile

  The value after an adaptation is derived from the maximum value of the number of bytes conforming to a profile within a monitoring interval multiplied by some constant.

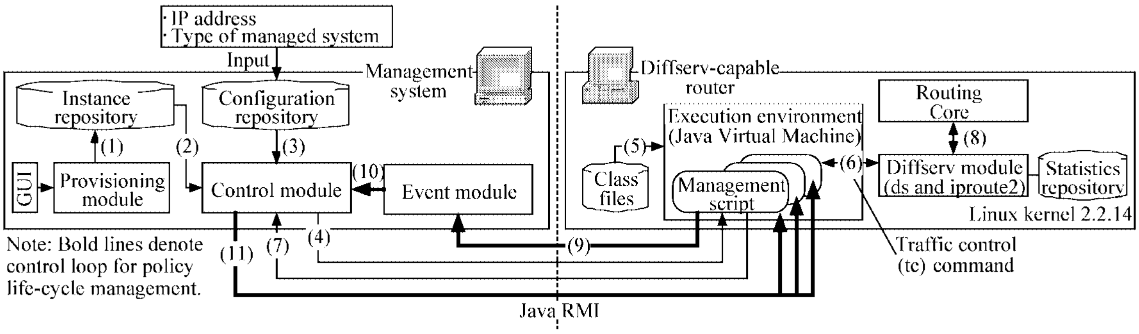


Fig. 10: Prototype system configuration.

- A peak burst size in a profile

  The value after an adaptation is derived from the average value of the number of bytes discarded caused by unconformity to a profile (burst) within an adaptation interval multiplied by some constant.

- DSCP

  The value after an adaptation is determined based on the number of occurrences of a threshold violation within an adaptation interval. If the number of occurrences excesses the given value, the packets is (re)marked with a DSCP representing such a forwarding treatment that each packet is sent with a higher loss probability from $\mathrm{AF}_{*,n}$ (Assured Forwarding) to $\mathrm{AF}_{*,n+1}$, where $*$ (class) $= 1, \ldots, 4$, and n (priority) $= 1$ or 2.

5 Prototyping

5.1 Design Principle

We implement a prototype system on the bases of the following design principles. Currently, the minimum function enough to evaluate the proposed scheme is realized. We show the prototype system configuration in Fig.10.

1. The system is based on the scheme in Section 4.
2. The system prompts values to be adapted and waits for operators' acknowledgment before a policy adaptation is actually executed.
3. Diffserv is adopted as a policy enforcement mechanism.
4. The managed system is a Linux box (Kernel 2.2.14) to emulate a Diffserv-capable router.
5. The "ds" and "iproute2" programs [ASK99] are used to realize the policy enforcement mechanism.
6. The traffic monitoring is performed using the (native) traffic control (tc) commands provided by the "ds" and "iproute2" programs.
7. The system is implemented in JDK v.1.2 (Java Development Kit version 1.2). The download of a management script and the communication between the management system and management script is realized by RMI (Remote Method Invocation).
8. The management script is automatically generated in the form of a Java object after providing the items described in Section 4.3.1 via GUI.

5.2 System Operation

We show how the system operates below. Due to the current system immaturity, we assume that an IP address and the type of a managed system (an edge router or a core router) are input and stored in the configuration repository in advance. In the context of policy-based management, this task should be automated by the integration with a directory server in the future extension.

	OS	CPU	Memory
Management system	WindowsNT4.0	PentiumII 450MHz	128Mbytes
Managed system(A, B, C)	Linux Kernel 2.2.14	PentiumIII 566MHz	64Mbytes

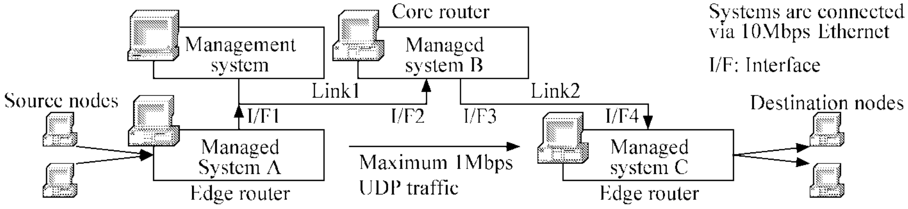


Fig. 11: Network configuration in evaluations.

The provisioning module stores a management script provided by an operator via GUI in the form of instance information in the instance repository (Fig.10(1)). When the download of the management scripts is requested, the control module takes out the instance information from the instance repository (Fig.10(2)). Subsequently, making reference to the configuration information in the configuration repository (Fig.10(3)), the control module downloads the management scripts including a policy associated with an MF classifier to edge routes and also the ones including a policy associated with a BA classifier to core routers (Fig.10(4)).

When a managed system receives the request (Fig.10(4)), the execution environment in the managed system optionally loads the necessary class files (Fig.10(5)), and instantiates the management script. The management script then sets the provided policies to be enforced to the diffserv module (Fig.10(6)), and responses to the management system if the enforcement has completed successfully or not (Fig.10(7)).

The management script monitors using the (native) traffic control (tc) commands provided by the ds and iproute2 programs (Fig.10(6)) that collect statistics from the routing core (Fig.10(8)) and store them into the statistics repository like MIB. If an event of a threshold violation or an expiration of an adaptation interval is detected, the management script identifies the target adaptation algorithms according to the provided items to be adapted and derives the values from the algorithms. After that, the management script notifies the management system of the event including the derived values (Fig.10(9)).

When the event module receives the event from the management script (Fig.10(9)), it prompts the values to be adapted to the operator. The operator makes a decision whether he/she actually adapts or not, and requests the policy adaptation depending on the decision (Fig.10(10)). If actually adapted, the control module indicates that all the associated management scripts synchronize with the adapted policy (Fig.10(11)).

As described above, the system achieves the policy life-cycle management iterating the sequential steps of (9), (10), and (11), drawn in the bold lines in Fig.10.

6 Evaluations

By applying the prototype system in Section 5 to an operational network, we evaluate the scheme from the following viewpoints: 1) the advantage of the policy adaptation on monitoring, 2) the reduced amount of management traffic required, and 3) the computational resources of a managed system required in the proposed scheme. We also discuss how the prototype system could be integrated with managed systems compliant with the standards being developed. Figure11 shows the network configuration in the evaluations.

6.1 Advantage of Policy Adaptation on Monitoring

For the purpose of showing the advantage of the policy adaptation on monitoring, we provide a management script including the parameters shown as in Tab.1, and generate UDP traffic at maximum 1Mbps from the source nodes to the destination nodes in Fig.11.

In consequence, the management scripts for I/F1 and I/F3 prompted the peak information rate from the initial value of 1.5Mbps to a more suitable value of 1.25Mbps. In this evaluation, assuming that the length of each packet is equal to the maximum size of an Ethernet packet, 1,514 bytes, causes the approximated value, while, in theory, the value should be 1.1Mbps (= the maximum 1Mbps × the constant 1.1). Acknowledging the prompt and synchronizing the other management scripts with the prompted value cause

Tab. 1: Parameters provided for management script* in evaluation.

(1)	BA classifier	"101110" (Expedited Forwarding)
(2)	Peak information rate in profile	1.5Mbps
(3)	Peak burst size	2Kbytes
(4)	DSCP marked for in-profile packets	"101110" (Expedited Forwarding)
(5)	DSCP marked for out-of-profile packets	"000000" (Best Effort)
(6)	Item to be monitored	Number of packets
(7)	Monitoring interval	5 seconds
(8)	Threshold for event	3,500
(9)	Item to be adapted	Peak information rate in profile
(10)	Adaptation algorithm	See Section 4.3.2 (Constant is 1.1)
(11)	Adaptation interval	1 hour

Remark: Parameters only for a policy associated with a BA classifier are shown. (1)~(5) for policy enforcement, (6) and (7) for traffic monitoring, and (8)~(11) for policy adaptation.

Tab. 2: Amount of management traffic required for each operation.

	Operation	Amount of management traffic (bytes)
(1)	Management script download request	25,625
(2)	Management script download response	17,929
(3)	Management script execution request	689
(4)	Management script execution response	999
(5)	Threshold violation notification	1,694
(6)	Adaptation interval expiration notification	1,364
(7)	Synchronization request	1,017
(8)	Synchronization response	3,352

Remark: Average values over 10 trials including Ethernet header.

0.25Mbps superfluous bandwidth released from the link1 and link 2 simultaneously. This allows us to make the most of the limited network resources.

The required bandwidth may be different from each link if a managed system has three or more physical interfaces as seen in a large-scale network, although, since all the traffic through the link1 also goes through the link2, the released bandwidth happens to be the same in Fig.11. With the proposed scheme, it is easily possible to provide a scalable and fine-grained policy adaptation, since the management script monitoring at every physical interface basically derives the value of the items to be adapted from what is monitored locally.

Therefore, the proposed scheme would be more advantageous as the growth of a network, and be one of indispensable technologies toward more promising policy-based management enabling the policy life-cycle management in a scalable, reliable and less laborious manner.

6.2 Management Traffic Reduction by Management Script

Table2 shows management traffic required for each operation in the prototype system. The amount of the management traffic required for each RMI-based operation is more than that for an SNMP operation. For example, the amount of the management traffic required for the threshold violation notification (Tab.2(5)) is four times as much as that for an SNMP Trap conveying the variable bind list including 18 variables. However, the proposed scheme can reduce much more management traffic than that of SNMP that relies on polling, since it is sufficient for the management script to notify only when a threshold violation occurs or to notify an event including aggregated statistics. Some quantitative evaluations can be found such as in [ZHG99, GGGO00].

6.3 Management Traffic Required for Management Script

On one hand, it is necessary to manage the execution context of management scripts. In the implementation, except (1) and (2) in Tab.2, the amount of the management traffic for the responses ((4) and (8) in Tab.2) is more than that of the corresponding request initiated by the management system. This is because the responses include additional information about the state transition of a management script, plus the

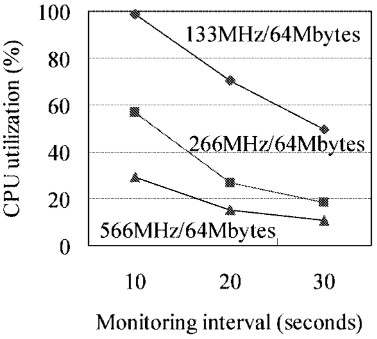


Fig. 12: CPU utilization of managed systems for executing management scripts.

parameters included in the corresponding request. The amount of the management traffic required for the synchronization response is six times as much as that for an SNMP Get Response conveying the variable bind list including 18 variables. However, the overhead can be small since the opportunity of the operations for the execution and synchronization is much less frequent than that for traffic monitoring. Note that the amount of the management traffic for the management script download request ((1) in Tab.2) is more than that for the corresponding response, since the policies in the request is not included in the response.

6.4 Load on Managed System

It is anticipated that a heavy load may be imposed on a managed system caused by the execution of management scripts, while the distributed nature of the scheme can reduce the management traffic required and achieve fault tolerance. Figure12 shows the CPU utilization of managed systems when we change the monitoring interval. We assume that the managed systems have 24 physical interfaces and thus 24 management scripts each monitoring 14 items are executed simultaneously.

The result provides a guideline of the computational resources of a managed system required when we apply the proposed scheme to an operational network. A managed system should be equipped with equivalence of 266MHz CPU and 64Mbytes memory if the monitoring interval is 20 seconds and the CPU utilization for the execution of management scripts should be less than 30%.

6.5 Integration with Managed System Compliant with Standards

It is expected that a managed system capable of enforcing policies, such as a Diffserv-capable router and a switch will soon be widely available, while we emulate a Diffserv-capable router by means of some specific program modules. Also, for the purpose of consistent policy enforcement in an entire management domain, IETF is standardizing SNMP MIB [IET01] and COPS PIB [IET00]. Even in such a situation, the prototype system could still easily be integrated with those emerging managed systems compliant with the standards. Binding the SNMP and COPS Java protocol stacks in generating management scripts can do this. In particular, in case of SNMP, the protocol stack has already been widely available. The prototype system would be of wider application as the diffusion of a managed system compliant with the standards.

7 Related Works and Future Study

The concept of the policy-based management [Wie94, Slo94] is not so new, and there have ever been many dedicated study efforts. The study issues toward the more promising policy-based management range from the policy specification and analysis to the architecture and realization. The most recent research results can be found in such as [SLL01]. In [HB99], an "active policy" in the form of a mobile and intelligent software agent is introduced, for more scalable policy-based management. In [KS00], another scalable policy-based management architecture base on the active network technology is presented. The main purpose of the study is to reduce the management traffic required for delivering the policy. Both of these works stay at presenting the architecture, and the policy life-cycle management is not explicitly considered.

The followings are planned for the future study among the immature aspects of the proposed scheme: 1) integration with a directory server allowing operators free from an initial configuration settings, 2) investigation of the more suitable adaptation algorithms, and 3) deployment in a large-scale operational network.

8 Conclusions

In this paper, in the context of IETF policy-based management, we proposed a new policy-based management scheme enabling policy life-cycle management in a scalable, reliable and less laborious manner. The policy life-cycle management in this paper means a sequence of 1) policy enforcement, 2) traffic monitoring to see if the enforced policy works at operators' will, and 3) policy adaptation (updating the condition and action clauses of the policy).

For the purpose of the policy life-cycle management, we introduced a new management script describing not only policies but also what is to be monitored and how to adapt the policies. In addition, for the scalability and reliability, we execute the management script on the managed system having enough computational resources, as well as in active network technology, the distributed management paradigm, and management by mobile software agents.

We implemented a prototype system based on the proposed scheme adopting IETF Differentiated Services as the policy enforcement mechanism, and evaluated the system from the following four viewpoints: 1) the advantage of the policy adaptation on monitoring, 2) the reduced amount of management traffic required, 3) the load on the managed systems executing the management scripts, and 4) the integration with managed systems compliant with the IETF standards.

The results shows that the proposed scheme allows us to alleviate the load on operators in analyzing management information and to make the most of the network resources by the fine-grained policy adaptation based on the monitoring. The results also shows that the required amount of the management traffic is much less than that in SNMP and provides a guideline of the computational resources of a managed system required when we apply the proposed scheme to an operational network. As the managed systems capable of policy enforcement and compliant with the standards penetrate the market, the scheme would be one of the essential technologies toward more promising policy-based management.

Acknowledgment

We are well indebted to Mr. Tohru Asami, President, Chief Executive Officer of KDDI R&D Laboratories Inc., and Dr. Yuichi Mastushima, Executive Vice President, for their continuous encouragement to this research.

References

[ASK99] W. Almesberger, J. H. Salim, and A. Kuznetsov. Differentiated Services on Linux. http://diffserv.sourceforge.net/, June 1999.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services*. IETF, RFC 2475, December 1998.

[CSD+01] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith. *COPS Usage for Policy Provisioning (COPS-PR)*. IETF, RFC 3084, March 2001.

[DMT] DMTF. http://www.dmtf.org/.

[GGGO00] D. Gavalas, D. Greenwood, M. Ghanbari, and M. O'Mahony. Advanced network monitoring applications based on mobile/intelligent agent technology. *Computer Communications*, Vol.23, No.8, 2000.

[Goh98] C. Goh. Policy Management Requirements. Technical Report HP-98-64, HP Laboratories, 1998.

[GY98] G. Goldszmidt and Y. Yemini. Delegated Agents for Network Management. *IEEE Comm. Mag.*, Vol.36, No.3, 1998.

[HB99] T. Hamada and D. Blight. Active Policy in Knowledge Hyperspace. In *Proc. of* 1999 *Asia-Pacific Network Operations and Management Symposium (APNOMS'*99*)*, September 1999.

[IETa] IETF Policy Framework Working Group. http://www.ietf.org/.

[IETb] IETF Resource Allocation Protocol Working Group. http://www.ietf.org/.

[IET00] IETF draft-ietf-diffserv-pib-00. *Differentiated Services Quality of Service Policy Information Base*, March 2000.

[IET01] IETF, draft-ietf-diffserv-mib-11. *Management Information Base for the Differentiated Services Architecture*, August 2001.

[KS00] K. Kato and S. Shiba. Designing Policy Networking System Using Active Networks. In *Proc. of IFIP IWAN* 2000, October 2000.

[Pso99] K. Psounis. Active Networks: Applications, Security, Safety, and Architectures. *IEEE Comm. Surveys*, 1999.

[RS00] D. Raz and Y. Shavitt. Active Networks for Efficient Distributed Network Management. *IEEE Comm. Mag.*, Vol.38, No.3, Mar. 2000.

[SLL01] M. Sloman, J. Lobo, and E. Lupu, editors. *Policies for Distributed Systems and Networks*. Springer-Verlag, 2001.

[Slo94] M. Sloman. Policy Driven Management for Distributed Systems. *Journal of Network and System Management*, Vol.2, No.4, 1994.

[Wie94] R. Wies. Policies in Network and System Management - Formal Definition and Architecture. *Journal of Network and System Management*, Vol.2, No.1, 1994.

[YGY91] Y. Yemini, G. Goldszmidt, and S. Yemini. Network Management by Delegation. In *Proc. of IFIP ISINM* '91, 1991.

[ZHG99] M. Zapf, K. Herrmann, and K. Geihs. Decentralized SNMP Management with Mobile Agents. In *Proc. of IFIP/IEEE IM* '99, 1999.