NAIST-IS-DD0361005

Doctoral Dissertation

Studies on Policy Based Route Management for Intra-domain and Inter-domain networks

Masashi Eto

January 31, 2006

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Masashi Eto

Thesis Committee:

Suguru Yamaguchi, Professor
Hideki Sunahara, Professor
Jun Murai, Professor
Youki Kadobayashi, Associate Professor

Studies on Policy Based Route Management for Intra-domain and Inter-domain networks*

Masashi Eto

Abstract

Since the Internet became a crucial social infrastructure, network is required to be more stable and flexible for users' demands. The rapid advances in network technology led an explosion of the number of nodes, and enabled dynamic changes of networks. In this situation, operational costs are increasing considerably whereas network operators are expected to reflect users' requirements to their network as quickly as possible. One of the reason is that, operation technologies have not been investigated enough, yet the communication technologies have been developed actively. Particularly in route management, operators are manually maintaining their networks based on a set of general rules which is called "management policy". In the implementation, they need to build up network behavior by combining basic functions of each network component. Though operators expect to automate this process, the management policy is too abstract to generate configurations for each individual component automatically. Because of the gap of abstraction level between policies and configurations, the policies are distorted in the conversion process of router configurations, then an unintentional network behavior occurs.

We assume that a stepwise refinement process from management policy to actual router configurations is the missing link of the current network management. In response to these issues, "*Policy Based Route Management (PBRM)*" method is

* Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0361005, January 31, 2006.

proposed in this dissertation, PBRM consists of a description method of route management policies, and a conversion process from the policy to the actual router configurations. According to the current routing management architecture, we split our efforts into two categories: inter-domain and intra-domain networks, and the goal of this dissertation is to deploy PBRM method into the two domains.

Since there are a network operation method which employs Routing Policy Specification Language (RPSL) and Internet Routing Registry (IRR) for inter-domain network, a dissemination of this method is efficient for the deployment of PBRM. Unfortunately, this method is not commonly used because of a low reliability and low availability of information in IRR databases. To solve this problem, this dissertation proposes two systems which urge IRR users to register correct policies. On the other hand, for intra-domain network, there have been few tools to establish PBRM so far, so operators have to manually decide link costs for route selection. This is the primary cause of misconfiguration and instability of networks, therefore an automation tool to ease operators' burden is strongly required. In this research, a definition of policy description is proposed, and a system is implemented, which automatically generates link costs with a consideration of mutual relations among all routers in the network.

In this dissertation, background and needs of the PBRM are introduced, then the details of our proposals are discussed. Through the deployment of the systems, the value and usefulness are confirmed. In the later chapter of this dissertation, several issues are discussed as the future work.

Keywords:

Policy Based Network Management, BGP, IRR Consistency, Routing Registry, OSPF, link cost assignment

Contents

List of Figures

List of Tables

1. Introduction

The current network management method is strongly required to be scalable so that the Internet keeps the rapid growth. In fact, the number of hosts in intra-domain network is ever-increasing, and the purpose of network use varies due to the diversification of services which require high speed, large bandwidth or high reliability circuits. Moreover, in inter-domain network, interconnections of Autonomous Systems (AS) across the world become possible by the improvement of the Layer-2 technology. Therefore, the number of interconnections will increase, and then establishment / abolishment of interconnections will happen frequently.

According to the growth of the Internet, operational costs are increasing considerably. In intra-domain network, operators spend a lot of time to allocate appropriate routes to each service, because they have to configure routers with considerations of mutual relations among all routers in the network. Even in inter-domain network, it is difficult for operators to maintain a number of interconnections concurrently. In a process of operation, they have to decide a priority of peer connections for optimal route selection, and apply the decision to multiple routers. They also have to clear unused connections as needed. If the Internet keeps growing within the current network operation, unintended route selections will occur, which induce inefficient network use or disconnection of end nodes.

To figure out the problem of the current network operation, we classified the process of the network operation as follows.

1. defining administrative requirements
2. designing network topology
3. deploying network equipments and physical links
4. configuring nodes based on the administrative requirements
5. checking and debugging the behaviours of the nodes

Currently, each process is executed manually, therefore we assume that it is efficient to provide automated method which supports the processes of network operations. This research focuses on the processes 4 and 5, which allow to take in an automated technology. The purpose of this dissertation is to provide an integrated operation method, which expedites the process and improves the effectiveness of network management as a result.

1.1 Stepwise Refinement of Management Policy

The problem of the current network operation is the lack of a stepwise refinement of administrative requirements. In the management process, operators need to build up network behavior by combining basic functions of each network component. A set of general rules based on operators' preferences and a capacity of the network is called *management policy*.

Since the management policy is too abstract, it is difficult to generate configurations for each individual component automatically. Because of the gap of abstraction level between policies and configurations, the policies are distorted in the conversion process of router configurations, then an unintentional network behavior occurs. In order to fill the gap, a stepwise refinement method is required, which has a capability to express a management policy and a function of policy conversion.

We can recognize that the essential problems of network operation are classified as the two categories.

1. Nonexistence of a Policy Level Description
   One of the problems of the current network operation is that, operators can not intuitively configure routers in an abstract description which has a capability to express their policies. When operators intend to reflect the policy to routers, they have to translate the policy to a router configuration language which consists of basic functions of routers. Router configuration language is defined by every router vendors, so that operators have to acquire as various syntax as the kinds of vendors. In the process of translation, the policy is deflected from its original meaning.

2. Lack of Consideration to Mutual Relations of Nodes
   Moreover, it is important to configure routers with a consideration of mutual relations among all entities in the network rather than managing them one-by-one. In inter-domain network, if an AS doesn't export a route information which is required by its peer AS, there is a possibility that the peer AS loses a connectivity to a certain destination AS. In intra-domain network, an operator needs to set appropriate link costs to a number of routers to enable expected route selection. If one cost value on a router is incorrect, it causes an unexpected route selection. Thus, when we configure a behavior of a network node, we need to examine whether the configuration is consistent among other nodes.

In conclusion, a policy level description is required, which is designed with a consideration of mutual relations among all entities in the network.

1.2 Policy Based Route Management Method (PBRM)

In response to the classified problems, we propose *Policy Base Route Management (PBRM)* method, which realizes a more stable network operation. PBRM consists of following functions.

1. a definition of policy description:
   For the problem of the nonexistence of a policy level description, the route management description should be constructed from abstract rules which reflects the policy intuitively. The configurations should be described in more abstract expressions than actual router configurations, and it should be able to satisfy operators' genuine requirements.

2. a function of policy conversion:
   After the policy description is defined, a system which converts a policy to actual router configurations is required, and the system also needs to generate router configurations for a number of routers at a time. Furthermore, the system is required to be constructed with a consideration of mutual

relations among all entities in the network to enable an unified network configuration.

Based on these definitions, this dissertation presents two proposals to cover missing pieces for the deployment of PBRM method in each field of inter-domain and intra-domain networks. In the next section, we discuss the current status of each network with a focus on the route management.

1.3 Applying PBRM to the Internet

This section presents an overview of the current network operations and clarifies the definition of policy for both inter-domain and intra-domain networks. Then some problems of deploying of PBRM method are introduced.

1.3.1 Deployment to Inter-Domain Network

Border Gateway Protocol (BGP), the most common routing protocol in inter-domain network, is basically designed to manage route selections arbitrarily to reflect operators management policy.

In BGP network, operators explicitly specify peer routers from all adjacent BGP speakers, and also manually specify route information which is exchanged between the peer routers to make up connectivities to another ASs. Because each management entity of AS is independent of other ASs, such arbitrary routing management was strongly required in inter-domain network. The route selection is decided with a reference of delay, bandwidth and monetary costs between each AS, thus arbitrary routing based on management policy is realized in BGP network. However, in BGP network, the number of peer AS is increasing yearly, so that it is difficult to apply configurations of route selection to all external routers one-by-one. To solve this problem, RPSL was proposed. RPSL has a capability to express route management policies in not only router level but also AS level by treating an AS as a network entity. With RPSL, an abstract policy description, and IRRToolSet, a policy conversion tool, operators are able to generate configurations for each individual router at a time.

The problem of the management method with RPSL and IRRToolSet is that this method focuses on only an individual AS. Since RPSL treats an AS as a network entity, the generated configurations have to define consistent route information exchanges among peer ASs. However, this method doesn't have a function to check the consistency between peer ASs. As a result, an unintended route selection may occur between BGP routers that are applied configurations generated by this method.

1.3.2 Deployment to Intra-Domain Network

OSPF, the most common routing protocol in intra-domain networks, is NOT basically designed for arbitrary route selection which reflects management policies based on delay limit, usable bandwidth, link stability and so on. As a historical background, OSPF routers are regulated to form a neighbor relationship and exchange route information autonomously, to manage many routers as simply as possible in intra-domain network. To manage route selection arbitrarily, operators need to configure OSPF link costs on each router, but because of the complexity of the configuration, it is difficult to apply management policies to the whole network. In other words, to realize autonomous route selections, the functionality of arbitrary route management was sacrificed in OSPF network. In today's network, although the necessity of an arbitrary route management method is increasing, there are few studies on policy based route management method for OSPF network.

1.3.3 Steps of PBRM Deployment

Realization of more stable and less labor-intensive network operation is the subject of this dissertation. Design and implementation of PBRM method is one step toward this goal. The states of network operation in the deployment process of PBRM are classified by several levels. The following items point out which level the current inter-domain or intra-domain network operation is in, and then describe challenges of this dissertation.

1. Nonexistence of Policy Description and Conversion Tools
   In today's OSPF network, operators manually decide link costs for each router one-by-one to reflect their routing policy to the network, so that we take in a concept of arbitrary routing management to OSPF network operation. In particular, we propose a definition of policy description and a policy conversion tool for OSPF network.

2. Existent of Policy Description and Node-level Conversion Tool
   In BGP network, although a route management method with RPSL and IRRToolSet have been proposed, this method is lacking a consideration of mutual relations among each network entity. Therefore, we propose a system which checks the consistency of policies among ASs.

3. Existent of Policy Description and Network-level Conversion Tool
   Finally, PBRM aims to realize this state of network operation. In this state, an abstract policy description is provided, so that operators can manage route selections intuitively. Moreover, a mechanism is also provided, which generates actual router configurations from policies with a consideration of mutual relations.

So far, we observed the current state of both inter-domain and intra-domain network operations and challenges we should solve. This dissertation introduces some proposals so that we apply PBRM method to both inter-domain and intra-domain networks.

1.4 Dissertation Overview

This dissertation is organized as follows. Chapter 2 surveys related works in fields of Policy Based Network Management. Chapter 3 produces the *Policy Check Server* which investigates and reduces the observed inconsistencies in public IRR databases. Then chapter 4 describes a system which is reasonable than manual calculation of each parameter of a machine's configuration, namely to generate link costs automatically from an abstract and minimum policy which indicates a user's management rules for a network. Chapter 5 provides some discussions of PBRM, and chapter 6 suggests the future directions of research and concludes this dissertation.

2. Related Work

The PBRM has two aspects: one is to define operation rules in an abstract description language, and the other is to generate router configurations for many routers collectively. These issues have been studied for many years in the category of network management.

From these aspects, this chapter provides an overview of related works in the fields of inter-domain and intra-domain network.

2.1 Route Management Methods in Inter-Domain Network

In inter-domain network operation, Routing Policy Specification Language (RPSL) and Internet Routing Registry (IRR) is deployed as a standard. The main role of IRR is to collect and publish the routing policies of ASs. Although there is a method to generate actual router configurations from a policy in the IRR with a tool called IRRToolSet, this method is not commonly employed.

The reason is that, there are no mechanisms to check the consistency of an AS's policy among policies of peer ASs. Therefore, when an operator applies the router configurations generated from inconsistent policy to their router, the exchange of routing information fails and the connectivity may be lost between the routers.

2.2 Route Management Methods in Intra-Domain Network

2.2.1 Policy Based Network Management (PBNM)

In the research field of Policy Based Network Management (PBNM), there are some proposals[2]. Resource reSerVation Protocol (RSVP)[3] and Common Open Policy Server (COPS)[4, 5] are designed to deliver defined routing policy to routers. Those protocols define how to exchange policies but not how to describe the policy. The objective of PBRM is to facilitate the high-level description of routing policy and its conversion to router configuration. We do not address the deployment issues of routing policy in this dissertation.

Some policy description languages are proposed in the area of PBNM, such as, PIB, PCIM and QPIM. However, those protocols only define the format of rules but not how to use the policy description for route management. Furthermore, there are no practical implementations of those protocols for route management.

2.2.2 Routing Optimization

There are some studies of routing optimization, which objective is to average out the network utilization. In response to this objective these proposals finally generate link costs for OSPF routers. In terms of the collective configuration of multiple routers, their approaches and PBRM have similarities.

Fortz [6] provided an approach of heuristic algorithm and Mulyana [7] applied a genetic algorithm for link cost assignment. These researches are also discussed in Chapter 4 in more detail. In those researches, traffic engineering techniques are used to generate the values of link cost for the optimization of the network traffic. Although we have the similar goal to achieve with the issues on unified network management, their approaches do not provide any flexibility for the administrators to deploy their own policy.

2.3 Summary

As mentioned in Chapter 1, the PBRM consists of two functionalities such as a definition of policy description, a function of policy conversion. For the inter-domain network management, the policy management method with RPSL and IRR and the policy conversion system called IRRToolSet, are commonly implemented and deployed. However, since a mechanism of the inconsistency resolution is not provided, it is difficult to widely deploy the route management method.

Although various approaches are proposed in the field of PBNM, most of them are mainly focused on a framework with the policy description format and the communication protocol among network entities, so that few studies are not practical implementation of route management.

In the research of routing optimization, although functions of the inconsistency resolution and the policy conversion are proposed, the policy description language is not defined, so that it is difficult to reflect an operator's management policy to the network with these researches. As a conclusion, in this paper we propose two solutions. In inter-domain network, a system which has a function of inconsistency resolution of policies is proposed. In intra-domain network, a system is proposed, which consists of a definition of policy description, a function of inconsistency resolution and policy conversion as a different way from the PBNM and routing optimization algorithms.

3. Improving the Reliability of IRR Database

As the deployment of PBRM to inter-domain network, we employ a route selection method with Internet Routing Registry (IRR). Then we figure out problems of this method which are induced by low reliability and low availability of IRR databases. In this chapter, efforts to solve these problems are described.

In inter-domain network, ASs often establish peer connections between multiple ASs and exchanges their route information each other. By the decline of connection fees and the advancement of Layer-2 technology in recent years, organizations are increasing, which obtain AS numbers and establish connections between the other organizations. Border Gateway Protocol (BGP), the standard routing protocol for inter-domain network, requires to specify the other AS to exchange route information. In other words, when an AS intends to make a connection to another AS, the operator needs to explicitly configure a transit AS to go through the destination AS. In inter-domain network, the selection criteria which is based on connection fees, bandwidth, hop count to the destination AS and so on, is commonly called "*Policy*". In this dissertation, we define the "Policy" as "*a selected transit AS to the destination AS*".

Although an operator's policy is expressed as AS level specification, the operator needs to manually interpret the policy into a router configuration language in the actual operation process. In particular, instead of the AS number, all route information which is exchanged with the peer AS have to be described in a router configuration. However, because vast amount of route information is exchanged in recent inter-domain network, it is difficult to correctly maintain all route information for each peer AS. Actually, there were some large scale network troubles which is caused by human-induced misconfigurations in the past. For example, in April 1997, AS7007 accidentally announced the route information to most of the Internet and disrupted connectivity of the global network for more than two hours [8]. In April 2001, AS3561 propagated more than 5000 of improper route announcements from one of its downstream customers, again leading to global connectivity problems [9, 10, 11].

As an alternative proposal of such an existing network management scheme, an operation method was proposed, which employs Routing Policy Specification Language (RPSL) and IRR. Whereas the past operation method requires to describe all route information for peer AS into the router configurations, RPSL allows AS level description instead of describing route information, and IRR and IRRToolSet enable collective settings of multiple border routers in own network. In other words, the method with RPSL and IRR provides a policy level description and a function of policy conversion, so that it is adequate for deploying PBRM in inter-domain network.

However, as we mentioned in the top of this dissertation, the operating method with IRR is not spread in the global inter-domain network. The main reason of this problem is the low reliability of IRR databases. If policies in current IRRs are applied to border routers automatically, unintended route selections or connection breakdowns may occur [12]. Therefore, technologies to improve the reliability of IRR databases are highly required. Eventually, these efforts lead to the realization of PBRM method in inter-domain network.

In this chapter, three approaches to improve the reliability of IRR databases are described. At first, we propose *Policy Check Server*, which is a system to point out inconsistencies between ASs' route selection policies and urges operators to correct them. Second, *IRR object garbage collector*, a mechanism to decrease unsound policies which are not updated in long term, is explained. We deployed these systems in JPIRR as a field work of our efforts in JPIRR, so a discussion of this field work is described at last.

3.1 Operation Method with IRR

IRR is a global Internet resource database that stores routing policies such as AS numbers and prefix information. IRR stores of several objects (Route object, aut-num object, Maintainer object, etc.). Policies registered in IRR are written in RPSL, which is designed to describe specific route information by import and export sentences in aut-num object. Operators can generate the vendor-specific router configurations from the policy data [13].

3.1.1 IRR Operation

Unlike Domain Name Server (DNS), the organization who operates IRR server is not regulated. Therefore some Regional Internet Registries (RIRs), National Internet Registries (NIRs) and many Internet Service Providers (ISPs) operate their own IRR servers in the world. Especially, RADB, RIPE and APIRR operated by Merit Network Inc, RIPE NCC and APNIC[14] respectively are known as the representative IRR services. They are classified roughly *Public IRR* and *Private IRR*. Public IRR makes registered information available to the public so that everyone can use the information to check some ISPs' route information. Private IRR holds nondisclosure information such as route information of ASs who form private peerings between them.

Public IRRs form data mirroring each other and in current operation, data of most IRRs are accumulated to the representative IRR servers described above.

3.1.2 Registering to IRR

Operators of ASs can register information about their ASs to one or more IRR(s) at any time. On the other hand, some Local Internet Registries (LIRs) force their customer ASs to register information to their own IRRs, and some LIRs substitute registering information for the customers.

3.1.3 Problem of Current IRR Operation

However, in its current operation, it is difficult to keep IRR database consistent for the following reason. That is, to register correct route information to IRR, operators need to check the consistency of them between each peer AS. However, in the recent Internet, an AS holds quite many peerings between many other ASs, so that operators have to bear a higher burden to check all of the consistency.

As a result, the database will contain many inconsistencies, and when router configurations are generated from this database, peer connectivity between ASs will be lost. Otherwise, an unintended link selection may occur.

On the other hand, IRRs do not hold all of the AS objects on the Internet, because operators of ASs are not forced to register their information in an IRR. This issue makes operators view an IRR as an incomplete database, causing a vicious circle which prevents the increased of utilization of IRRs [15].

3.2 Classification of the Inconsistency

In this section, we discuss the definition of inconsistencies that could prevent peer connectivity between ASs.

The inconsistencies are roughly classified into the following two types:

- inconsistency in route information import
  There are less description of route information in export sentence or there are too much description of route information in import sentence.
- inconsistency in route information export
  There are less description of route information in import sentence or there are too much description of route information in export sentence.

In the following subsections, we first explain each type of inconsistency, and then describe classified inconsistencies in more detail.

3.2.1 Inconsistency in Route Information Import

When a peering connectivity is held between a source AS and a transit AS, the transit AS has to export the route information of the destination AS, and the source AS has to import the route information. If this route information is missing in the policy of the source AS or the destination AS, the source AS cannot get connectivity to the destination AS. We focus on and explain a case *inconsistency in route information import* in the following example, which the export sentence for the destination AS is missing in the transit AS's policy.

As shown in Figure 1, AS 1 and AS 2 operate under the contract that AS 2 provides transit for the traffic from AS 1 to AS 3 and AS 4. According to this contract, the operator of AS 1 registered the policy shown in Figure 2 which is configured to import the routes for AS 2, AS 3 and AS 4 from AS 2. On the other hand, the policy registered by the operator of AS 2 is shown in Figure 3; in this policy, the route of AS 4 is missing by accident. The router configurations generated from these policies by IRRToolSet [16, 17] are shown in Figures 4 and 5. In these configurations, each network address presents the route of each AS. Note that in Figure 4, AS1 imports routes of AS 2, AS 3 and AS 4. On the other hand, in Figure 5, AS 2 exports only AS 2 and AS 3. If the operators apply these configurations to their routers, the router of AS 1 cannot receive the route of AS 4 so that AS 1 cannot establish connectivity with AS 4.

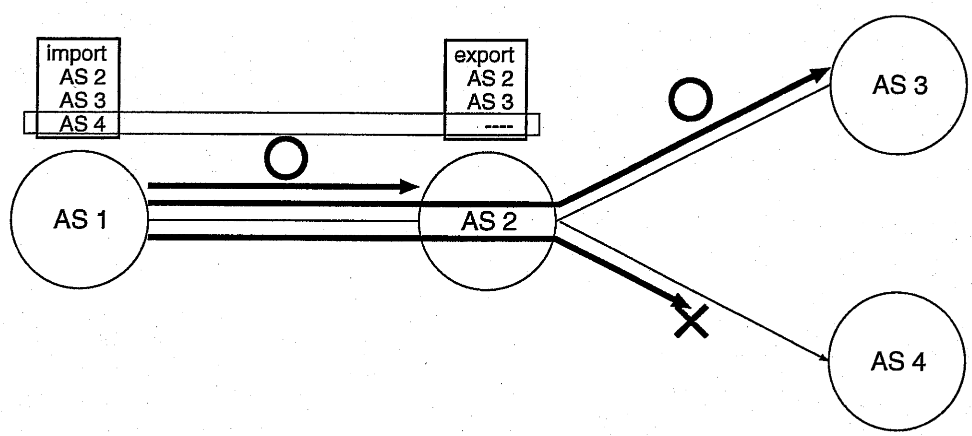

Figure 1. Inconsistency in route information import

3.2.2 Inconsistency in Route Information Export

In contradiction to above example, we explain a case, which the import sentence is missing in the source AS's policy. If a source AS does not import the destination AS's route information from the transit AS, the source AS cannot establish

```
aut-num: AS 1
as-name: EtoNet
...

import: from AS 2
        accept AS 2, AS 3, AS 4
...
```

Figure 2. Policy registered by AS 1

```
aut-num: AS 2
as-name: SaiNet
...

export:  to AS 1
         announce AS 2, AS 3
...
```

Figure 3. Policy registered by AS 2

```
import proto bgp as AS 2 {
  192.168.2.0 masklen 24 exact;
       //route information of AS 2/
  192.168.3.0 masklen 24 exact;
       //route information of AS 3/
  192.168.4.0 masklen 24 exact;
       //route information of AS 4/
  all restrict;
}
```

Figure 4. Configuration on a router in AS 1

connectivity between the destination AS. In the following example, we describe this problem as the *inconsistency in route information export.*

```
proto bgp aspath .* origin any {
  192.168.2.0 masklen 24 exact;
       //route information of AS 2/
  192.168.3.0 masklen 24 exact;
       //route information of AS 3/
  all restrict;
}
```

Figure 5. Configuration on a router in AS 2

Assume that AS 1 and AS 2 registered the policies shown in Figures 7 and 8. In this case, the transit provider (AS 2) registered the proper policy according to the contract. However, in the policy of AS 1, the sentence required to import the route of AS 1 is missing by accident. The router configurations generated from these policies are shown in Figures 9 and 10. In this case, although AS2 exports routes of AS2, AS3 and AS4, AS1 imports only AS2 and AS3. As a result, AS 1 cannot establish the connectivity with AS 4.

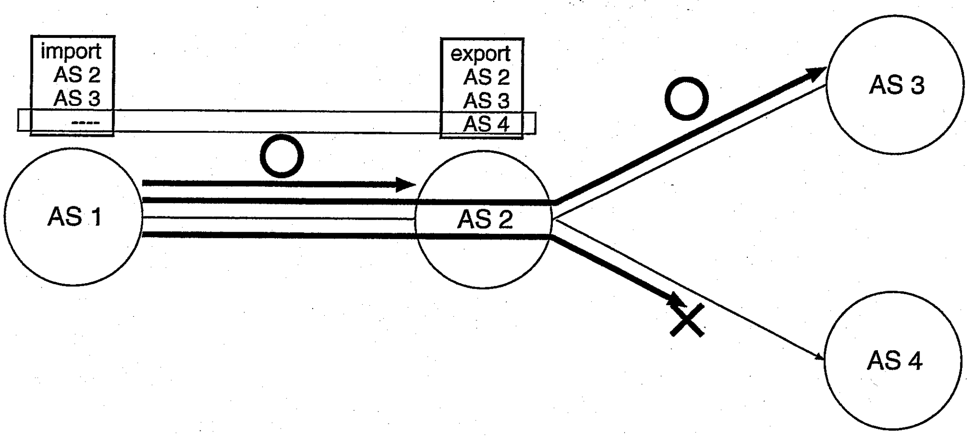

Figure 6. Inconsistency in route information export

Based on these premises, we have classified more details of these inconsistencies as shown in Table 1, in which *AS-SET* object is an aggregate of aut-num

```
aut-num: AS 1
as-name: EtoNet
...
import:  from AS 2
         accept AS 2, AS 3
...
```

Figure 7. Policy registered by AS 1

```
aut-num: AS 2
as-name: SaiNet
...
export:  to AS 1
         announce AS 2, AS 3, AS 4
...
```

Figure 8. Policy registered by AS 2

```
import proto bgp as AS2 {
  192.168.2.0 masklen 24 exact;
      //route information of AS 2/
  192.168.3.0 masklen 24 exact;
      //route information of AS 3/
  all restrict;
}
```

Figure 9. Configuration on a router in AS 1

object. Like an AS specifying another AS as a peer, an AS can also specify an AS-SET as a peer.

3.2.3 Other Attributes

Although there are several other attributes and parameters in an aut-num object, we focus on inconsistencies classified in Table 1 for this research.

```
proto bgp aspath .* origin any {
  192.168.2.0 masklen 24 exact;
       //route information of AS 2/
  192.168.3.0 masklen 24 exact;
       //route information of AS 3/
  192.168.4.0 masklen 24 exact;
       //route information of AS 4/
  all restrict;
}
```

Figure 10. Configuration on a router in AS 2

Table 1. Classification of inconsistencies

Inconsistencies in route information import	peer AS-SET does not exist on IRR database
	peer AS does not exist on IRR database
	peer AS does not export any routes to the AS
	peer AS does not export route which the AS imports
Inconsistencies in route information export	peer AS-SET does not exist on IRR database
	peer AS does not exist on IRR database
	peer AS does not import any routes from the AS
	peer AS does not import route which the AS exports

In import and export attributes of an aut-num object, RPSL defines *action* attributes that enable more detailed configurations, such as *MED*, *Local Preference* and *community* parameters, which are described in Figure 11.

```
import:  from AS 2
   action pref = 10 ; med = 0 ;
   community.append(10250, {3561,10}) ;
   accept AS 2, AS 3, AS 4
```

Figure 11. import sentence with action attribute

In this case, an action attribute sets a constraint on imported route prefixes for each parameter. Of course, it is possible that these parameters are inconsistent with the specification of the peer AS's action attribute.

One of inconsistencies that may disturb connectivity between peering ASs is the inconsistency caused by description of Well-Known Community in COMMUNITY path attribute.[18] When an AS sets *NO_EXPORT* COMMUNITY path attribute to a certain export route, the route will not be announced to the peer AS. On the other hand, if the peer AS configured the policy to import the route, their policies are inconsistent and the connectivity may be disturbed.

In this research, to simplify the classification of inconsistencies, we focus on only the inconsistencies described in Section 3.2.1 and 3.2.2.

Therefore, we focused on inconsistencies classified in Table 1 in this research.

3.3 Discussion of Required Solution

In this research, we aim to establish a mechanism to conduct accurate inspection on a global scale with high efficiency. To accomplish this goal, we need the following three systems: a system for a-priori inspection of a policy's consistency before it is registered; a system to aggregate all IRR databases in the world; and a system to perform a comprehensive inspection of consistency in the aggregated IRR databases.

3.3.1 A-priori Inspection of Consistency

We propose and implement a system to investigate the consistency of AS policies in IRR databases globally. When an operator intends to register his/her AS's policy, it is difficult to check whether the policy is consistent between peering ASs. This fact may cause inconsistencies to arise between policies of the operator's AS and peering ASs, and we therefore need a system to inspect the consistency of the policy before it is registered with the IRR database.

3.3.2 Aggregation of IRR Databases

We decided to collect and aggregate all policies of IRR databases in the world, to inspect consistency in more detail.

When any organization gains an AS number from the RIR, it needs to register its policy with the IRR database. However, the IRR server is managed by any organizations as we mentioned in Section 3.1.1. Therefore, the policies of each AS are distributed to each of the IRRs. To inspect the consistency of AS policies, we have to collect and store them in one place. Because the databases are open to the public, we decided to collect all 55 of the accessible IRR databases including RIPE, RADB and APNIC. In this paper, we call the collected databases the *Unified IRR Database.*

Unified IRR Database

Now, we describe the algorithm of the Unified IRR Database. Basically, we extract all of aut-num objects from each IRR database and store them in the Unified IRR Database. aut-num object in the Unified IRR Database consists of AS number, import sentences and export sentences. At this moment, we should notice that there are duplicate aut-num objects registered into multiple IRRs. In this case, we merge them based on the following rules.

1. Restructure the duplicate aut-num objects as a aut-num object and store import and export sentences in it.

2. If multiple sentences import *different* routes from *same* peer AS, store both sentences in separate. Treat multiple export sentences as same.

3. If multiple sentences import *same* routes from *same* peer AS and they have *different* actions, check the updated date and adopt the most up-to-date sentence.

4. In the above step, if the updated dates of both sentences are same, determine they are inconsistent and discard them. Because we can not decide automatically which sentence is the valid direction.

We have considered another algorithm that is to adopt the most up-to-date object and discard the other instead of merging them. However, we decided to merge them because the operator might divide the AS's information into several parts and register them into multiple IRRs intentionally.

3.3.3 Comprehensive Inspection

By aggregating IRR databases as described above, we can perform a comprehensive inspection of all the accessible IRR databases currently in operation. Then, as described in section 3.3.1, we can make clear the need for a-priori inspection of the policies.

3.4 Designing the Inspection System

Before implementing these systems, we have to discuss the adequacy of our methodology compared with alternative proposals for aggregation of IRR databases. Then we have to consider the synchronization of the Unified IRR Database and the Public IRRs.

3.4.1 Using Whois Query

As an alternative proposal, we could use whois[19] query to perform a-priori inspection. IRRd has whois interface to provide us registered ASs' information. When we send a query that specifies a particular AS number to an IRR server, the IRRd sends back to us a response that includes the specified AS's information [13]. Using this function, we can implement the following system: when the system receives a query of policy inspection, it sends whois queries to IRRs, then it can collect peer ASs' policies.

On the other hand, as described in Section 3.3.2, each AS's policy is deployed in IRRs that are also repeated throughout the world. Besides that, there is no appropriate way to know which IRR the required policy is in, so that it is difficult to perform a-priori inspection if all the IRR databases are not integrated.

For this reason, we decided to construct the Unified IRR Database instead of using a whois query.

3.4.2 Database Synchronization

We decided to update the Unified IRR Database every thirty minutes for the following reason. In current operation of IRR, many IRRs mirror their information each other every thirty minutes. On the other hand, there are some IRRs that mirrors once a day. Therefore, the Unified IRR Database updates its database every thirty minutes in accordance with the shortest cycle of the other IRRs. In this case, the time lag between the Unified IRR Database and a certain IRR may be one hour at a maximum. Considering that there are IRRs that mirrors once a day, we determine that this time lag does not lead to critical problems of inspection.

3.5 Proposed System and Implementation

To investigate and prevent the inconsistencies, we propose *Policy Check Server*, which consists of three main components as follows.

- To inspect consistency between ASs, we need the complete set of policies for all the accessible IRR databases in the world. Therefore, we constructed the Unified IRR Database by *DBGenerator*.
- To make clear the necessity of inspecting consistency of policies, *Database Checker* is produced, which inspects how many inconsistencies exist on the Unified IRR Database.
- *Policy Checker* gives an opportunity to an AS's operator to inspect whether his policy is consistent with the policies of its peering ASs.

3.5.1 DBGenerator

DBGenerator collects policies from IRRs, and aggregates them as the Unified IRR Database. Although ASs' policies are dispersed across a number of independent public IRRs, most of public IRRs are mirrored from RIPE, RADB and APNIC, and available for everyone with constraints about redistribution. DBGenerator then injects the AS objects into a database, which is constructed by PostgreSQL database. In the aggregation process, if there are an AS's duplicated objects in several IRRs, DBGenerator aggregates and reconstructs them into an object rather than discard either one of them. Thus, DBGenerator tries to collect an AS's policies as much as possible.

3.5.2 Database Checker

Database Checker inspects how many inconsistencies exist on the Unified IRR Database. It inspects all the policies in the Unified IRR Database according to the algorithm shown in Figure 12.

The algorithm consists of three phases.

1. Database Checker specifies the peer AS by import or export sentences, and holds the peer AS as an AS object. If the peer AS does not exist on the Unified IRR Database, Database Checker records this fact as an inconsistency.

2. Database Checker compares import sentences of the input AS and export sentences of the peering AS. If the peering AS does not have an export sentence which specifies the input AS as a peer like this:

   - export: to *input AS* announce AS 3 ... ,

   Database Checker records this fact as an inconsistency. Otherwise, Database Checker determines whether the peering AS exports the route prefix which the AS intends to import from. If it does not, Database Checker records the fact as an inconsistency. In the next phase, Database Checker compares the export sentences of the input AS and the import sentences of the peering AS.

3. Database Checker outputs the collected inconsistencies to a log file.

```
[ specify peering AS ]
    extract import, export sentences from input AS object ;
    if (the peering AS (or AS-SET) exists on database) {
        create "Autnum" object as a peering AS ;
    } else {
        warn as an inconsistency "Peer AS (AS-SET) doesn't exist on IRR database" ;
    }
[ inspection of import sentence ]
    for (number of import sentence of input AS) {
        for (number of export sentence of peer AS) {
            if (the export sentence of peer AS specify input AS as a peer) {
                if (the sentence doesn't export required routes) {
                    warn as an inconsistency  "Peer AS doesn't export route which the AS imports ;
                }
            } else {
                warn as an inconsistency "Peer AS doesn't export any routes to the AS ;
            }
        }
    }
[ inspection of export sentence ]
    for (number of export sentence of input AS) {
        for (number of import sentence of peer AS) {
            if (the import sentence of peer AS specify input AS as a peer) {
                if (the sentence doesn't import required routes) {
                    warn as an inconsistency "Peer AS doesn't import route which the AS exports ;
            } else {
                warn as an inconsistency "Peer AS doesn't import any routes from the AS" ;
            }
        }
    }
```

Figure 12. Inspection Algorithm

As we explained in Section 3.1, IRRs do not hold all of the AS objects in the Internet. Regarding this issue, the following situation can be assumed.

Since an AS imports 50 route prefixes from its peer AS, if the peer AS is not registered with any IRR, it is assumed that Policy Checker issues 50 warnings for every route prefix. However, the inconsistencies are based on a single factor: the peer AS is not registered with any IRR. To eliminate these redundant warnings, we bind up these inconsistencies in one inconsistency, which is "peer AS does not exist on IRR database". Policy Checker is designed to detect this inconsistency using the algorithm shown above (Figure 12).

3.5.3 Policy Checker

Policy Checker gives an operator the opportunity to inspect the policy which he/she intends to register with an IRR database. Policy Checker keeps all of the latest entries of an IRR database as the Unified IRR database, which is made by DBGenerator, so that it is suitable for Policy Checker to be managed inside an IRR server.

The flow of the process is as follows.

1. The policy input by the operator is transmitted to Policy Checker.
2. Policy Checker creates an AS object from the input policy and transmits it to Database Checker.
3. Database Checker then inspects the consistency between input policy and the peer AS's policy, as described in section 3.5.2.
4. Database Checker returns the collected inconsistencies to Policy Checker.
5. Policy Checker generates an HTML document from the result of the inspection, and displays it on the operator's web browser.

The process of inspection starts on a web-based interface which is deployed as a Java Servlet on TOMCAT (Web application server). If any inconsistencies are detected, Policy Checker displays warnings on the operator's web browser. The operator of the AS may then correct the corresponding entries and register the consistent policy. The basic composition is shown in Figure 13.

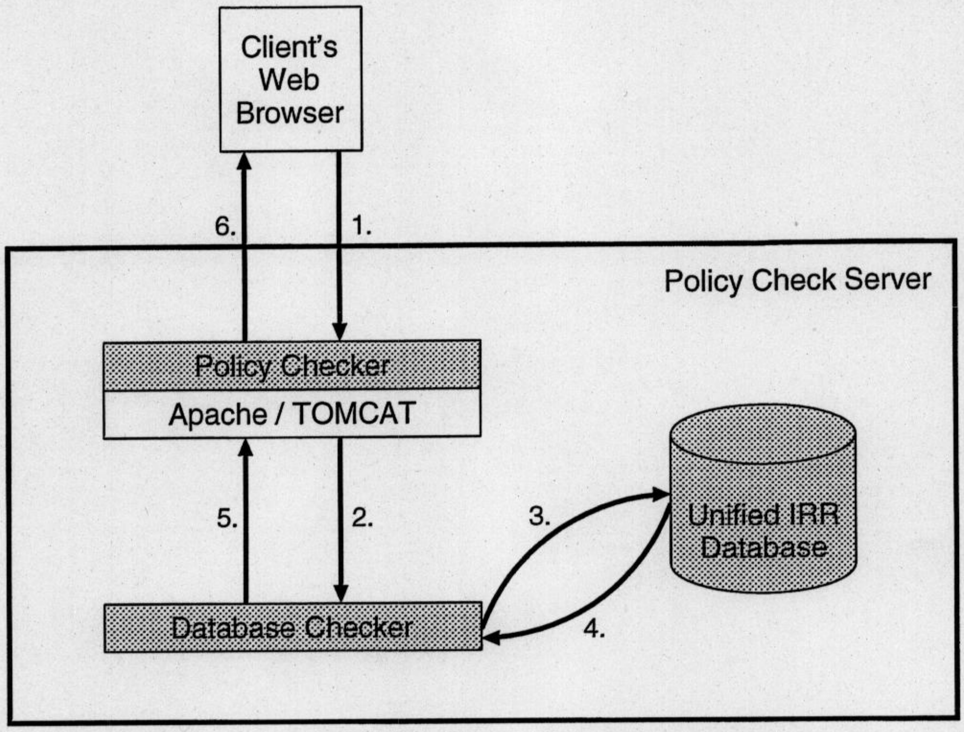

Figure 13. Basic components of Policy Checker

3.6 Analysis of Inspection Results

We built up the Unified IRR Database from 55 public IRRs such as RIPE, Level3, RADB, Cable and Wireless, APNIC, Verio and so on. Most of these databases are mirrored by RADB, so that we obtained them from RADB IRR server. The list of the collected main objects and attributes in the Unified IRR Database is shown in Table 2.

As a result of our investigation, we have found that 71.2% of the 12,582 registered ASs in IRRs have at least one inconsistency which is shown in Table 1. These results are shown in more detail in Figure 14, which indicates that there is variation in the number of inconsistencies according to the AS number. In other words, inconsistency decreases as the AS number becomes larger.

Table 2. List of Objects and Attributes

Objects / Attributes	Number
aut-num object	12,583
as-set object	4,807
route object	296,035
import sentence	52,986
export sentence	51,107

We assume that ASs who have smaller AS numbers have many inconsistencies because of following reason. AS numbers are assigned in an order from smaller one to larger one by RIRs, therefore, smaller AS numbers are assigned earlier time than larger AS numbers. The difference of the assigned time between smaller and larger AS numbers may affect the numbers of the inconsistencies. In other words, it is thought that an AS which has a smaller AS number tends to have many peers, so that the AS has many import or export sentences and many inconsistencies.

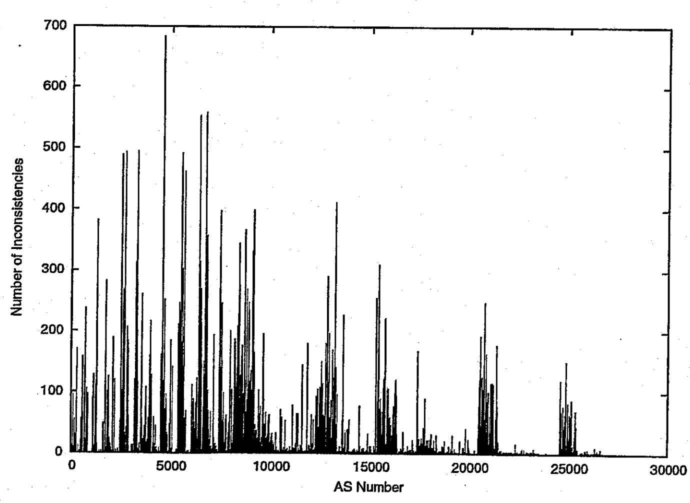

Figure 14. Number of inconsistencies in each AS

In Figure 14, we divided the AS numbers into every 1000 numbers and calculated the average of the number of inconsistencies in each slot. As the result, Figure 15 shows the averages of inconsistencies per 1000 ASs. By this figure, it became clear that old ASs (ASs which have smaller AS numbers) have much inconsistencies than newer ASs.

Figure 15 also shows the rate of each type of the inconsistencies classified in Table 1. One of the notable features of this figure is that inconsistency of *Peer AS doesn't import/export route which the AS exports/imports* increases as the AS number becomes smaller. This fact means that although many old ASs specify peer ASs in their routing policy, they do not import or export necessary routes. We suppose that old ASs may have many peer ASs, so that it is difficult to describe the correct route information.

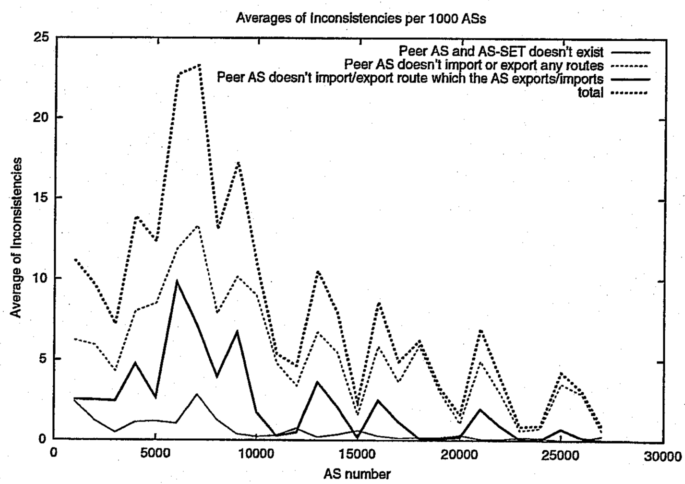

Figure 15. Averages of Inconsistencies per 1000 ASs

Details of the inconsistencies in all registered AS policies are shown in Table 3. In Table 3, the "Rate" column shows the rate of ASs which have any of classified inconsistencies in all 12,582 ASs. As a significant result, we found that the rate of two categories *Peer AS does not export any routes* and *Peer AS does not import* *any routes* are particularly higher than the other categories. This result means that, although a particular peer AS exists in the IRR database, the peer AS does not specify the AS as a peer. RPSL is designed to describe the routing configuration, particularly for import and export sentences. Although operators can increase their efficiency of operation on the BGP network by generating router configurations from IRR database automatically, we found out that this functionality is hardly used at all.

Table 3. Details of inconsistencies

	Classification	Inconsistent ASs	Rate (in 12,582 ASs)
1	Peer AS-SET does not exist	64	0.5%
2	Peer AS does not exist	4,876	38.6%
3	Peer AS does not export any routes	5,553	44.1%
4	Peer AS does not import any routes	5,437	43.2%
5	Peer AS does not export expected route	444	3.5%
6	Peer AS does not import expected route	728	5.8%
	Total of Inconsistent AS	8953	71.2%

3.7 Performance Evaluation

Database Checker spent 2100 seconds to inspect the whole database. Since the number of registered aut-num objects is 12,582 entries, it takes an average of 0.17 seconds to inspect each aut-num object. However, this figure is just an average for each inspection, and actually the time for each inspection increases linearly with the number of import and export sentences. It is thought that the number of policies in IRR databases will increase, so that it is necessary to reduce the time for the inspection.

Because Database Checker sends several queries to PostgreSQL database for every import and export sentence while it inspects the policy, it is clear that the majority of the time required for the inspection process is spent for disk I/O. In future work, therefore, we will improve the performance of Database Checker by optimizing the algorithm to reduce the number of physical disk I/Os.

3.8 Related Work

The Routing Registry Consistency Check (RRCC) project [15, 20] reports inconsistency between IRR databases and the real Internet. Tools which detect inconsistencies are available on their web site.

The definition of RRCC's *"inconsistency"* is a gap between an AS's policy and actually advertised route prefixes such as: route prefixes which are not advertised on the real Internet although they are registered on the IRR database, and route prefixes which are not registered on the IRR database although they are advertised on the real Internet.

On the other hand, our research detects inconsistency among the registered policies. Since both of RRCC and our research aim to improve the integrity of IRR databases by correcting their inconsistencies, these two studies complement each other.

3.9 Discussion

In this section, we will consider the detail classification of inconsistencies such as Well-Known Community problem as we mentioned in Section 3.2.3.

We also need to consider the aggregation algorithm of the Unified IRR Database. In this paper, we discarded duplicated sentences on multiple IRR databases. We discuss the other algorithms that takes in both of duplicated sentences.

3.9.1 Consistency Chain

By correcting inconsistencies between peering ASs, we believe that it is possible to exchange route information between ASs that are not directly peering. Eventually, it will be also possible to improve the consistency of all IRR databases.

For example, consider the situation shown in Figure 16. In this situation, the requirement is to give AS 1 connectivity to AS 2 and AS 3. To complete this requirement, each AS has to declare that it will import or export expected routes.

1. At the initial state (Figure 16(a)), AS 2 does not export the route of AS 3 to AS 1. Furthermore, AS 3 does not export the route of AS 3 itself to AS 2. At this state, the route of AS 3 is never transmitted to AS 1, so that AS 1 cannot establish connectivity to AS 3.

2. At this state, if operators use Policy Checker, it tells them that AS 2 does not export the expected route to AS 3, so that the operator of AS 2 would be able to correct the corresponding entry (Figure 16(b)).

3. However, at the next state (Figure 16(c)), Policy Checker tells the operator of AS 3 that AS 3 does not export any routes to AS 2. On this warning, the operator of AS 3 would be able to subjoin an entry properly.

4. As a result, the policies of each AS are corrected and AS 1 is able to receive the route of AS 3 (Figure 16(d)).

Finally, connectivity between AS 1 and AS 3 is established. This connectivity is established only when operators use Policy Checker and correct the corresponding entries. Thus, by using Policy Checker, it is possible to check consistency between ASs that are not directly peering.

In the near future, we intend to investigate this functionality of *Consistency Chain* in all accessible IRR databases.

3.10 IRR Object Garbage Collector

As we mentioned at the beginning of this chapter, we have designed and implemented "IRR Object Garbage Collector", which forces owners of objects in IRR database to update their objects in a certain period. The main purpose of this effort is also to improve the reliability of IRR database. IRR database stores various kinds of objects that are registered by maintainers of each AS so that IRR database reflects practical state of current inter-domain network. However, IRR accumulates objects which are not updated for a long-term after they are registered. These objects lose touch with the practical network operation, as a

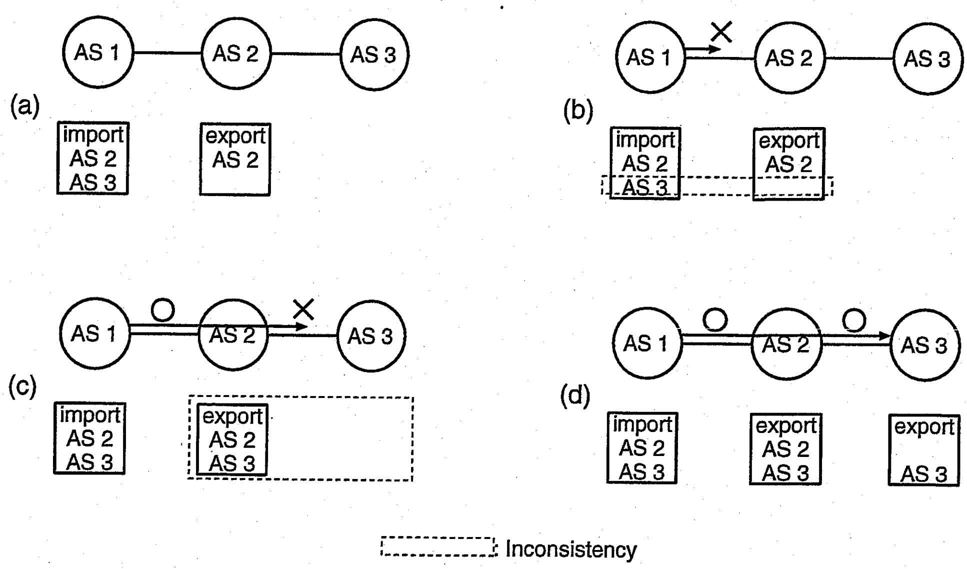

Figure 16. Consistency Chain

result, they impair the reliability of IRR database. Although this issue is well known as the *IRR's garbage object problem* for a long time, no countermeasures have been taken for this issue.

As a solution for this issue, we have designed and implemented *IRR object garbage collector*, which is an attempt to reduce garbage objects in an IRR database. IRR object garbage collector classifies objects into following four categories based on *changed* field of each object.

- initial state: objects that are newly registered or updated
- expired: objects that are not changed for a certain period after registered or updated
- access precluded: objects that are not changed for a certain period after expired

  In this state, the objects are removed from the database and stored in a temporary file.

- deleted: objects that are not requested to recover by their owner for a certain period after accesses to them are precluded
  In this state, these objects are removed from the repository file and they are completely disposed from IRR.

According to these states, IRR object garbage collector executes following processes.

- check and prenotification of expiration Based on changed dates, the system checks whether each object is expired or not. If it is expired, the system sends a prenotification of the expiration to the maintainer of the object by an e-mail.
- access preclusion and notification After the expiration date, if an object is not updated in a certain period, the system temporarily removes the object from IRR database, and sends a notification to the maintainer. The object becomes hidden from users of IRR after this process. and stored in a repository file for a certain term.
- deletion The objects in the repository file are deleted in this process unless the maintainers of those objects request to restore them in a certain period.
- display a list of expiration dates of objects For the maintainers of each object, IRR object garbage collector features a function to display the expiration dates of objects with the web browser.

After the initial test operation, this system is deployed and in operation on the IRR server of JPNIC. In the initial operation, the system published notifications of expiration for the 180 objects that is not updated for over twelve months, and these notifications were sent to 24 maintainers of each object. After three months of the initial operation, in the access preclusion process, nine objects that is maintained by three maintainers were removed from database to the repository file. Accordingly, it is clear that most objects were updated after they are expired by the day of access preclusion process. By long-term operation of IRR object garbage collector, it will keep high integrity IRR database.

3.11 Field Work at JPIRR

As a proof of our activities as JPNIC IRR Planning Team, we show the current status of the consistency in JPIRR database. Table 4 represents the comparison of the rate of ASs which have one or more inconsistencies.

Table 4. Comparison of the Inconsistent ASs

	Classification	Inconsistent ASs	Rate (in 37 ASs)	Rate of Unified IRR
1	AS-SET does not exist	0	0%	0.5%
2	AS does not exist	2	5.4%	38.8%
3	AS does not export any routes	15	40.5%	44.1%
4	AS does not import any routes	15	40.5%	43.2%
5	AS does not export expected route	1	2.7%	3.5%
6	AS does not import expected route	2	5.4%	5.8%
	Total of Inconsistent AS	15	40.5%	71.2%

In every type of inconsistency, ASs in JPIRR showed lower rate of inconsistency than ASs in Unified IRR. Although the number of the stored AS policies in JPIRR is 37, fewer than Unified IRR, this result indicates JPIRR users' high awareness of registering correct policies than the other IRRs' users.

However, the rates of inconsistencies "*AS does not export / import any routes*" are much higher than the other inconsistencies in JPIRR. Therefore, it is still necessary to appeal the importance of keeping consistency of AS policies to enable PBRM method.

As another aspect of our field work, we compared the update status of objects in Unified IRR and JPIRR. Figure 5 shows that most objects in JPIRR are updated within two years whereas there are many objects which are not updated since 1990's in Unified IRR. In Unified IRR, there are even many objects which are not updated since 1995, and most of these objects do not reflect current routing policies. On the other hand, although JPIRR has been in operation for only four years since 2002, objects in JPIRR have been updated at least one time. IRR Object Garbage Collector permits an object's expiration term to be maximum two years, so even the oldest objects in JPIRR have been updated in 2003.

Table 5. Comparison of Changed Date

Year	Unified IRR	Rate	JPIRR	Rate
(before) 1999	10175	8%	0	0%
2000	5690	4%	0	0%
2001	11376	9%	0	0%
2002	14680	11%	0	0%
2003	19632	15%	10	1%
2004	28236	21%	28	2%
2005	39478	30%	1356	96%
2006	2318	2%	13	1%
total	131585	-	1407	-

In this way, AS operators obtain precise notifications, which are the opportunities to check whether their registered policies are correct or not. In this regard, we did not employ a way to correct AS policies automatically because a modification which is contrary to the operator's intention would cause enormous influence to the global Internet. To advance the use of IRR for the deployment of PBRM method, tools are required, which help AS operators to use IRR more conveniently such as Policy Check Server and IRR Object Garbage Collector.

3.12 Summary

In this chapter, a deployment of PBRM method in inter-domain network operation was discussed. Since there are RPSL, IRR and IRRToolSet which are a set of operation tools to support BGP network operation, we decided to follow and advance the use of them. As our efforts, Policy Check Server and IRR Object Garbage Collector were proposed to improve the reliability of IRR database.

We defined and classified inconsistencies in IRR database into two categories, "*inconsistency in route information import and export*", which can potentially disrupt the connectivity between peering ASs.

Based on this classification, we proposed Policy Check Server which enables to inspect a management policy in advance of a registration. Policy Check Server consists of three components, *DBGenerator*, *Policy Checker* and *Database Checker*. Database Checker is a system to investigate the consistency of AS policies in all accessible IRR databases in the world. As a result of the investigation by Database Checker, we have found that 71.2% of ASs have one or more inconsistencies. From this result, we advocate that AS operators should take the consistency between the other ASs' policies into consideration before registering it to IRR. To complement this concept, Policy Checker was implemented, which gives a web interfaces to check mutual relations of AS policies.

As another activity, we implemented IRR Object Garbage Collector which put a term limit of 12 months to objects in IRR database. In every month, IRR Object Garbage Collector sends notification messages of objects' expiration to users, and then deletes expired objects. It also gives operators a web interface to check their objects' status.

Currently, Policy Check Server and IRR Object Garbage Collector are implemented and in operation at JPNIC as its first step of deployment. After three years of the operation, the higher consistency of JPIRR database is indicated in comparison with the other IRRs.

4. Policy-Based Cost Assignment Algorithm for Intra-Domain Network

This chapter describes a proposal of a link cost assignment algorithm and system for OSPF routers, as an adoption of PBRM method in intra-domain network. The proposal defines the policy as "*a route selection criteria based on capability of links between objective nodes, such as delay, bandwidth and stability*", then a system is proposed, which employs a multiple dimensional constraints and linear programming method.

As same as inter-domain network, the number of routers in intra-domain network is increasing year by year. In recent years, there is a wide variety of services which involve real-time and bursty traffic such as Voice over IP (VOIP) and video streaming service as well as WWW and e-mail. To provide these services concurrently, operators have to assign appropriate routes to each service. Although routers in an intra-domain network autonomously form connections with neighbor routers, it is difficult to configure the route selection for every services by manually. A route selection is implemented by configuring cost of each link in OSPF, the common routing protocol in intra-domain networks. In the process of the configuration, operators have to assign the link costs in the whole network by fine-tuning, whereas BGP allows to specify a transit route explicitly. In the current network operation, operators have to assign link costs manually. Since configuring link cost becomes difficult according to increase of network nodes, operators have to spend much time and human resources for router configurations.

On the other hand, there are several tools which apply the Simple Network Management Protocol (SNMP) [22], and are commonly used to support network operation. These tools reduce the burden of configuring routers by hiding differences between vendor-specific commands with Graphical User Interfaces (GUI) or more user-friendly commands. Although operators use these tools, they still have to assign the actual link costs. In other words, these tools apply link costs to routers, but they do not assign the appropriate link costs. In addition, there are some proposals of link cost generation which employ heuristic algorithm and genetic algorithm as we describe from the next section. Because the objective of these proposals is just the equalization of network utilization, they are unsuitable for the route selection based on the type of network usage.

The PBRM aims to realize a more simple management process of route selections than existing network management method. We assume that there are multiple dimensional policies on each pair of objective nodes. Therefore we define the policy in intra-domain networks as "*a route selection criteria based on capability of links between objective nodes, such as delay, bandwidth and stability*".

In this chapter, we define a policy description method firstly, and then an algorithm is proposed, which automatically generates link costs for multiple routers from a policy concurrently.

4.1 Related Works

In this section, we will review previous work on link cost determination algorithms based on network conditions. In IGP protocols such as Open Shortest Path First (OSPF) [23] and IS-IS, primary and secondary route is determined based on link costs configured by operators. Although configuring link costs has a significant impact on network management, it is more difficult as network becomes larger.

To reduce operators' burdens, some link cost determination algorithms have been proposed.

4.1.1 Heuristic Algorithm

The link cost determination algorithm proposed by Fortz and Thorup [6] adopted a heuristic algorithm for traffic engineering to derive semi-optimal route selection. In this approach, a penalty is applied to the link when the flow rate exceeds a certain threshold, and a new cost is then set which prevents subsequent overflow of the link. By applying this rule to all links, network usability was improved. Since link costs before and after adoption of this approach may differ vastly, it may take a long time for convergence. Pham [24] modified Fortz and Thorup's approach by limiting the range of the change, and reduced the convergence time.

4.1.2 Genetic Algorithm

Mulyana [7] applied a genetic algorithm (GA) for cost determination. This approach applies the number of routers, number of links and their bandwidth, utilization rate and its average, and current link costs to a genetic algorithm for cost determination. Link costs were compared before and after adoption of the algorithm as a constraint for fast convergence.

4.1.3 Problem of Related Works

The purposes of these approaches are to automate network operation. Link costs are determined and configured dynamically, without the operator's burden, and in this respect, their approaches satisfy the requirements of automated operation.

In current standard network operation, operators seldom use these techniques, and they still determine link costs manually. We assume this is because the existing approaches do not reflect operator's genuine requirements; operators do not really want to equalize the rate of flow across the network, but just to apply their own organizations' policies to particular routes.

We discuss this issue in more detail, and show our proposed solution to this problem, in the following section.

4.2 Policy-Based Cost Assignment Algorithm

The purpose of our approach is to determine link costs automatically based on a policy. In our approach, operators need only to specify their routing policy, that is, their preferences such as delay, bandwidth and stability. Our system selects the most preferred route which satisfies the given policy, and then assigns appropriate costs for each link. In this section, we show the basic algorithm used in our approach.

4.2.1 Definition of Policy

Before describing our approach, we need to define the word *policy.* Policy is an operator's selection of routes based on delay, bandwidth and stability. The operator can choose the most preferred parameter from delay, bandwidth and stability on each route between a certain source and destination.

The meanings of these parameters are as follows.

- delay: one-way transmission delay
- bandwidth: bandwidth of each interface on a router
- stability: the number of times that a link is down for a certain period

Preferences of Routes

How are parameters selected on a real network? We have to consider both *the local preference* and *the global preference.*

The local preference is a preference for links on certain source and destination nodes. For example:

- An operator may prefer a wider band link on a route between a WWW server and an egress router, on the assumption that collisions by many accesses and attacks may otherwise occur.
- An operator may prefer a more stable link on a backbone route.
- An operator may prefer a low-delay link on a route for applications that are sensitive to delay, such as VoIP and streaming.

The global preference is a single policy that is applied across the network. For example:

- An operator may choose between delay, bandwidth or stability as the preference for the whole network except on routes where local preferences are applied.

- If a network failure occurs on some links, an operator may switch to more stable links on an alternative route to prevent network flapping.

Operators do not need to apply local preferences to every route in the network, but only to certain routes.

A set of these preferences is defined as the *policy*. Policy is not based on utilization of a network, but on management preferences which reflect operators' actual intentions flexibly.

We propose a link cost determination algorithm in which operators just need to specify local preferences and a global preference. The basic algorithm is shown in the next section.

4.2.2 Assignment Algorithm

The algorithm consists mainly of three phases: *primary and secondary route selection*, *constraints determination* and *link costs determination*. The purpose of each phase is as follows.

1. *primary and secondary route selection*
   In this phase, an operator can specify a preference for a route between certain source and destination nodes, and also specify a preference for an alternative route in the case of network failure between the nodes. Based on these preferences, our system determines primary and secondary routes between all nodes in the whole network.

2. *constraints determination*
   From the selected routes, our system determines constraints to derive appropriate costs by using linear programming [25, 26, 27] in phase 3.

3. *link costs determination*
   Finally, our system determines the cost for each link using linear programming.

The specific processes of each phase are described below.

Primary and Secondary Route Selection

In this section, we describe the algorithm to select primary and secondary routes between source and destination nodes in the whole network. In this phase, we decide primary and secondary routes for each pair which consists of all nodes in the network.

At first, our system decides *pseudo-costs* that are temporary parameters to derive primary and secondary routes from specified local and global preferences. Besides these preferences, we use some more parameters (D_{ij}, B_{ij}, S_{ij}) that are given from network conditions.

These parameters are defined as follows:

- delay:

  When the result of an Round Trip Time (RTT) measurement between two routers i and j (X_{ij}(msec)) is given, the value of delay (D_{ij}) parameter is defined as

  $$D_{ij} = X_{ij} \qquad (X_{ij} > 0)$$

- bandwidth:

  When the bandwidth of a link between routers i and j (Y_{ij}(Mb/s)) is given, the value of bandwidth (B_{ij}) is defined as

  $$B_{ij} = \frac{1}{Y_{ij}} \qquad (Y_{ij} > 0)$$

- stability:

  The stability value is determined by the frequency of Link State Advertisement (LSA) type-4 packets' appearance. When the number of frequencies on router i (Z_i) and j (Z_j) are given, the stability value of the link between the routers (S_{ij}) is defined as

  $$S_{ij} = Z_i + Z_j \qquad (Z_i, Z_j \geq 0)$$

For a route between certain source and destination nodes, the local preference or the global preference means the priority of these parameters. The higher the priority become, the more the operator should define a greater coefficient. The preferences are expressed as coefficients (α, β, and γ) in the following pseudo-cost (PC) calculation formula:

$$PC = \frac{1}{SD(D)}\alpha D_{ij} + \frac{1}{SD(B)}\beta B_{ij} + \frac{1}{SD(S)}\gamma S_{ij}$$

In this formula, $SD(D), SD(B), SD(S)$ are the standard deviation of each value of delay, bandwidth and stability on the all links in the network. We used the reciprocal of the standard deviation to correct the disproportion of the value that is caused by the difference of the measure. In the next step, we apply these pseudo-costs and link parameters to Dijkstra algorithm to derive the primary and secondary routes.

Fig 17 is an example which shows how pseudo-costs are derived from given parameters (D_{ij}, B_{ij}, S_{ij}) and the operator's preferences (α, β, γ). It also shows that the primary route is selected based on the pseudo-costs.

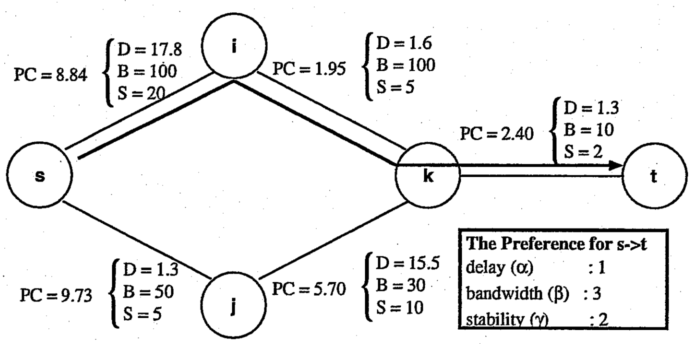

Figure 17. Primary Route between s, t

By the way, in our approach, there will be multiple costs on a link, because primary and secondary routes are set for each pair which consists of the all nodes in the network. As a result, there is a possibility that our algorithm determines a closed path with the already derived primary or secondary routes. In Fig 18, the route from node s to k via node i is selected as the primary route between nodes s and t. On the other hand, the route from node s to k via node j is also selected as the primary route between node s and k. These routes form a closed path, so that it is difficult to decide which route should be selected.

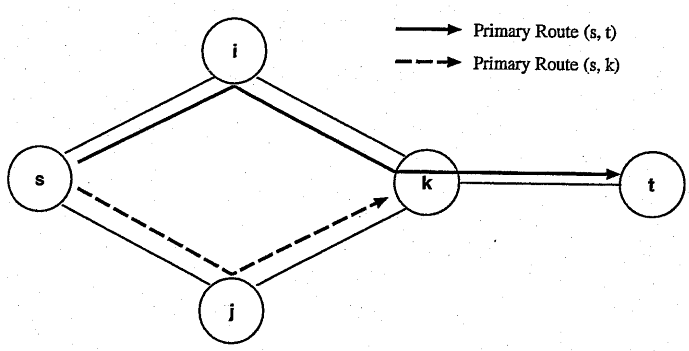

Figure 18. Closed Path

To avoid this problem, we made a following change to Dijkstra Algorithm: when our system detects a route which comprises a closed path on another route, it does not update the shortest path with the route.

Now we explain the determination algorithm of a secondary route. When a certain link is assumed to be down, our system first detects pairs of source and destination nodes whose primary routes are influenced by the link down. It then determines secondary routes for each detected pair of nodes using the same algorithm as the primary route.

Constraints Determination

Having determined primary and secondary routes for all nodes in the network, our system now derives the constraints to determine the *Unified Link Costs*, which are the actual link costs of a router configuration. This is done by using the following rules:

- cost of primary route $<$ cost of secondary route

- cost of primary route < cost of each route except the primary route
- cost of secondary route < cost of each route except primary and secondary routes

By applying these rules to all routes, we can obtain the constraints for linear programming determination of Unified Link Costs. However, it costs a vast amount of computational time efforts to search all routes between source and destination nodes and derive constraints for them. To reduce the computation, we searched only routes consisting of two paths: the shortest path between the source node and the node adjacent to the destination node; and the path between the adjacent node and the destination node. This practice reduces the computational complexity to $O(n^3)$.

This mechanism is illustrated in Fig 19.

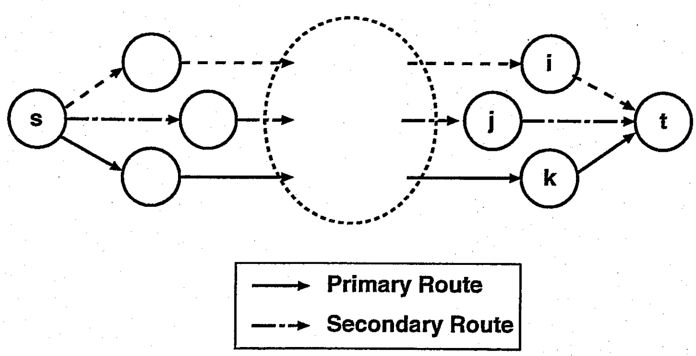

Figure 19. Constraint Route

Let us assume that $X_{src,dst}$ is the total cost of the primary route between the source and destination nodes, $Y_{src,dst}$ is the total cost of the route between source and destination nodes, and $WT_{src,dst}$ is the cost of the link between an adjacent node and the destination node.

With these assumptions, the following constraints are formed with the adjacent nodes i, j, k.

$$X_{st} < Y_{st}$$

$$Y_{st} < X_{si} + WT_{it}$$

Hereafter, we call the route the *constraint route* which is expressed by $X_{si} + WT_{it}$.

According to the algorithm described below, we can now derive the constraints for the primary, secondary and constraint routes.

Link Cost Determination

Having determined the constraints, we now explain how to derive the Unified Link Costs from those constraints.

When the following definitions are given:

- adjacency matrices of primary, secondary, and constraint routes, respectively: $Cp_{ij}, Cs_{ij}, Cc_{ij}$.
- a set of link cost of each route: X_{ij}
- any positive integer: ξ

we can express the constraints as follows.

$$\sum_{i=1}^{n}\sum_{j=0}^{n}(Cs_{ij} - Cp_{ij})X_{ij} < \xi$$

$$\sum_{i=1}^{n}\sum_{j=0}^{n}(Cc_{ij} - Cs_{ij})X_{ij} < \xi$$

$$X_{ij} \geq 1, \ (i = 1, 2, 3..., n), \ (j = 1, 2, 3..., n)$$

These are linear inequalities of X_{ij} that have adjacency matrices as coefficients. Consequently, we can solve these constraints with linear programming by giving an objective function. Optimization with the objective function is not so important, however, because our goal is not the optimization but rather to derive costs which just fulfill operators' policy.

Operators do not usually set each cost in increments of one, because if they want to add new links in the network, or if they change their policy later, they need slack between each cost value to avoid having to change all the costs on each link. To derive costs that have appropriate slack between each other, we hold costs as low as possible, and multiply them depending on operators' needs. Under this assumption, we define the objective function as follows.

$$min \sum_{i=1}^{n} \sum_{j=0}^{n} X_{ij}$$

We can now obtain the Unified Link Costs by applying these constraints and objective functions for linear programming.

4.3 Implementation

We implemented a prototype of our proposed algorithm. The architecture of this system is shown in Fig 20.

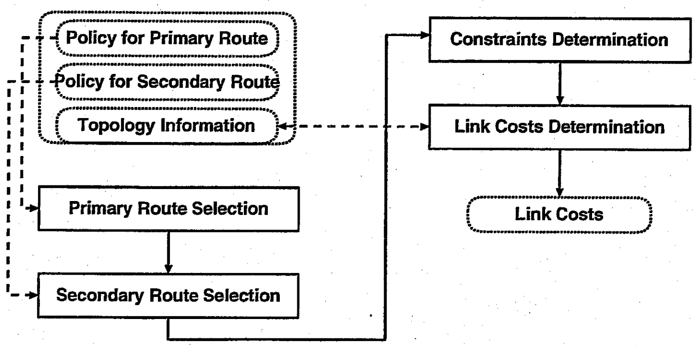

Figure 20. System Architecture

At first, the system receives topology information about the network, and policies to decide primary and secondary routes. It then generates primary and secondary routes from the topology and policies in *primary route determination phase* and *secondary route determination phase*, respectively. With the derived routes, the system next generates constraints in *constraint determination phase*, and the system finally generates the Unified Link Costs with a free tool named "lp_solve" in the *link cost determination phase.*

4.3.1 Primary Route Selection Phase

In this phase, the system receives the policy as user input, which consists of topology information about the network expressed by adjacent matrices, and operator preferences such as delay, bandwidth and stability.

We devised a determination algorithm for the primary route based on the theory described in section 4.2.2. All primary routes are determined by applying this algorithm to each source and destination node of the whole network. These routes are stored and passed to other phases.

4.3.2 Secondary Route Selection Phase

In this phase, the system receives topology information and operator preferences as in the primary route selection phase. Unlike the previous phase, the system changes the dead link in the topology information, then searches the primary routes for routes that go through the dead link. The system applies the determination algorithm to these routes, and also does a closed path check by comparing the current route with the primary and secondary (if it already exists) routes. Thus, the system determines a secondary route between source and destination nodes that is invoked by a certain link's failure.

4.3.3 Constraint Determination Phase

This phase consists of the following three components.

- objective function determination
- constraint determination
- finite limitation and counting number limitation of link costs

The results from each component are formatted as constraints for linear programming.

Objective Function Determination

This component receives the topology information and searches pairs of nodes which have finite route, in other words, which have at least one route between them. The results are written in a format such as "min : $x_i x_j + ...$"to a file named *constraint.dat.*

Constraints Determination

Based on section 4.2.2, this component generates constraints of primary and secondary routes.

This component specifies the adjacent nodes of a destination node, then it checks whether the routes are redundant or not.

Limitation of Link Costs

The values of link costs in OSPF are limited to a range from 1 to 65535, so that this component generates a constraint to keep the costs within the appropriate range. This component also generates a constraint to make costs integer values.

4.3.4 Link Cost Determination

In this phase, the system determines the Unified Link Costs by using the free tool "lp_solve"as a solver for linear programming. An example of objective function and constraints is shown in Fig 21. In this figure, the objective function "min: x1x2 + ..."implies minimization of the sum of the costs between source and destination nodes. The entries bracketed with /* */ imply nodes adjacent to each source and destination node.

These constraints are put into lp_solve to obtain Unified Link Costs which satisfy the objective function and constraints, as shown in Fig 22.

4.4 Evaluation

To examine whether the determined Unified Link Costs reflect the operator's policy effectively, and whether this system reduces the operator burden in comparison with manual configuration, five graduate students participated in the

```
min: x1x2 + x1x3 + x2x3 + x2x5 + .. + x8x9;
x1x2 = x2x1;
x1x2 > 1;
x2x1 > 1;
x1x2 < 65535;
x2x1 < 65535;
x1x3 = x3x1;
    :
    :
x9x8 < 65535;
    :
    :
/* src 1 dst 2, neighbor 1 */
/* src 1 dst 2, neighbor 4 */
/* src 1 dst 2, neighbor 5 */
x2x3 + x3x1 + 1 < x2x5 + x5x3 + x3x1;
    :
    :
x8x6 + 1 < x8x5 + x5x3 + x3x6;
/* src 6 dst 8, neighbor 7 */
/* src 7 dst 8, neighbor 5 */
x8x7 + 1 < x8x5 + x5x7;
/* src 7 dst 8, neighbor 6 */
int x1x2;
int x2x1;
    :
    :
int x8x9;
int x9x8;
```

Figure 21. Objective Function and Constraints

```
Value of objective function: 53

Actual values of the variables:
x1x2                    3
x1x3                    1
x2x1                    3
   :
   :
x8x9                    4
x9x7                    3
x9x8                    4
```

Figure 22. Determined Unified Link Costs

experimental test of the system. In addition, we examined the scalability of the system, for use in a large scale network with a topology data of a real network and constraints, and measured the time spent for the cost determination with the system.

4.4.1 Comparison with Manual Configuration

The network topology and its specification for the experiment are shown Figs 23 and 24. In addition, we defined the following policy to determine the primary route:

1. prefer the widest bandwidth route
2. prefer a less delay route if the routes have the same bandwidth

For the secondary route, we defined the following policy.

1. prefer the most stable route
2. prefer the widest bandwidth route if the stability of those routes have the same value

In these conditions, we carried out the experiment whose result is shown in Table 6.

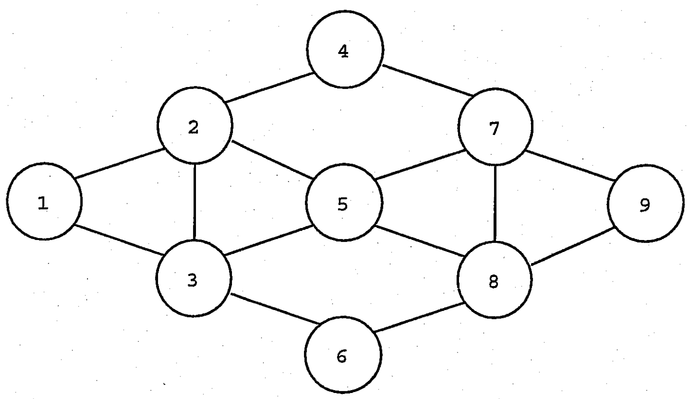

Figure 23. Topology

route	delay (msec)	bandwidth (Mb/s)	stability
1 - 2	1.6	10	0
1 - 3	0.8	100	5
2 - 3	0.6	100	6
2 - 4	15.5	10	10
2 - 5	17.8	10	20
3 - 5	1.3	1000	15
3 - 6	113.2	10	15
4 - 7	122.4	100	10
5 - 7	125.1	10	0
5 - 8	123.4	10	0
6 - 8	10.3	100	0
7 - 8	5.2	10	2
7 - 9	8.3	1000	2
8 - 9	96.4	100	20

Figure 24. Topology Data

Table 6. Result of Experimentation

	manual (min)	proposed system (min)
A	30	2
B	76	8
C	60	54
D	27	6
E	53	20

Although the result of the manual experiment varied widely, Table 6 shows that it takes more than twenty seven minutes to determine the link costs manually. On the other hand, almost all examinees finished their work within about twenty minutes with our proposed system. This result shows the efficiency of our proposed system.

Examinee C spent more than fifty minutes completing his work because of mistakes in using the system. This result does not revoke our proposition because it is caused by a problem of the system's user interface.

4.5 Summary

We aimed to reduce operators' burden and misconfigurations in current standard network operation.

On the assumption that the existing approaches do not reflect an operator's genuine requirements, we showed the necessity of a policy-based link cost determination algorithm for OSPF. We then designed and implemented an algorithm and a system which focuses on applying constraints to each pair of source and destination nodes, with Dijkstra algorithm and linear programming. Finally, we demonstrated experimentally that operator burden was reduced, and we showed the effectiveness of the system for a large scale network. As the future work, we will prove the validity of derived costs by comparing the performance with costs derived by the existing approaches.

5. Discussion

In this chapter, we discuss the adequacy of our proposal, PBRM. Then we also discuss about the necessity of each proposal for both inter-domain and intra-domain networks.

The contribution of this research is to indicate the efficiency of a network management method based on policy. This dissertation defined operator's genuine requirements for network management as "management policy", and then proposed PBRM method which consists of a definition of policy description and a function of policy conversion. The conversion system concurrently generates configurations for multiple routers with a consideration of mutual relations of routers. In the implementation process of PBRM method, we took care not to make much changes to existing protocols or management methods for widespread deployment of proposed system.

5.1 Necessity of Each Proposal

With the motivation that PBRM should follow existing network architecture rather than replacing them, we have firstly reviewed the current network operation of inter-domain and intra-domain networks.

The route selection method of BGP has been called "*policy routing*" in inter-domain network, because operators select routes based on "management policy" which is a set of general rules such as delay, bandwidth and monetary cost between peer ASs. To support policy routing in more efficient way, there have been several proposals: RPSL is a policy description language which has a capability to express routing policy by treating AS as an network entity. IRR and IRRToolSet are used to convert policies written in RPSL to actual router configurations for multiple border routers in the AS concurrently.

In other words, we realized that there is a network management method which is compatible with PBRM method. Therefore, we tried to improve the efficiency of the existing method to promote the widespread use of it. One of problems of the method with IRR is inconsistencies of policies among AS objects registered in

IRRs as mentioned in chapter 3. Based on a fact that the inconsistencies detract the reliability of IRR, *Policy Check Server* have been proposed, which is a system to prevent the increase of those inconsistencies. In addition, as a solution of IRR's garbage object problem, we provided IRR Object Garbage Collector which urges operators to correct objects outdated.

On the other hand, in intra-domain network, there are some studies of QoS which enables arbitrary route selection. However, sometimes the adoption of QoS mechanism accompanies additional equipment investment and large scale change of network topology.[28, 29, 30, 31] To avoid such modification of network, we aimed to take in a concept of arbitrary routing management to OSPF network which is commonly used for intra-domain network. In OSPF network, operators need to manipulate values of link cost on each routers in the network to realize certain route selections. Therefore, we defined a definition of policy description and implemented the policy conversion system which generates actual link costs for OSPF routers as mentioned in chapter 4.

5.2 Contribution to the Global Internet

It is important to improve reliability of whole IRRs in the world. To check consistency among policies in only one IRR is not efficient for the stability of the global Internet. In other words, investigating consistency of policies across the IRRs in the world is one step toward the improvement of reliability of the global Internet.

In chapter 3, a mechanism to prevent increase of inconsistencies in IRR database was introduced. Policy Check Server collected and aggregated all IRR databases in the world for total inspection of consistency of not a single IRR but across IRRs. In this way, every operator in the world can use Policy Check Server with his web browser wherever he is. In the future, we will release and promote Policy Check Server to more users.

5.3 Synchronization of IRR Database

Policy Check Server has a possibility to output inconsistent inspection results because of time-lag between update of objects in IRR and reflection of Unified IRR Database. It is difficult to avoid this problem because of following reasons.

- As mentioned in Section 4, most of IRRs are mirroring their data to major IRRs such as RADB, RIPE and APIRR. When Policy Check Server checks inconsistency of policies between distinct IRRs, each IRR has to reflect up-to-date information to the major IRRs in advance. The mirroring needs to be done as fast as possible because the inspection results might be based on inconsistent database. Figure 25 shows the interval time of mirroring between RADB and JPIRR via APIRR. From this figure, we recognized the results were dispersed from five minutes to three hundreds minutes.

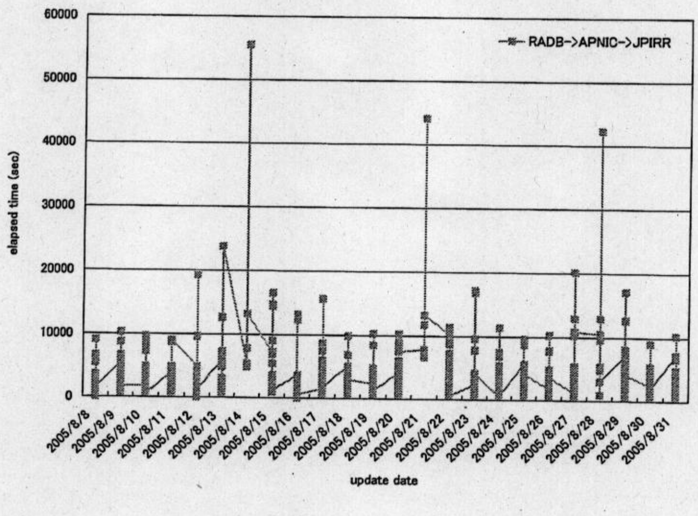

Figure 25. Interval time of mirroring

In response to the result, as JPNIC IRR planning team, we are working on measurements and cooperation with other IRRs to enable speedy mirroring.

- In case the objective ASs of policy check are existing in a single IRR, Policy Check Server can inspect without data of other IRRs. As mentioned in Section 3, Policy Check Server firstly collects data of other IRRs and builds up Unified IRR Database which is used for every inspection. Therefore, the inspection result might be incorrect because the data is not up-to-date for about two hours at minimum until the update process of Unified IRR Database finishes. Since a misconfiguration in inter-domain network has a significant impact on the global Internet, the consistency of IRR database is very important, which supports the reliable route information exchange. Through our activities as JPNIC IRR planning team, we know that the most of update frequencies of objects are from several days to several months. Therefore, Policy Check Server can avoid this problem by encouraging users to check their policies several times at intervals of several hours.

6. Conclusions and Future Work

The main theme of this dissertation is how to realize more stable and less labor-intensive network operation. Design and implementation of the PBRM method is one step toward this goal. With the review of current network operation, we found several problems and figured out the problems are classified essentially two issues: *Nonexistence of a Policy Level Configuration Description* and *Nonexistence of a Consideration Process of Mutual Relations of Nodes.* To solve these problems, *Policy Based Route Management (PBRM)* method was proposed, which consists of a definition of policy description, a function of inconsistency resolution and a function of policy conversion. Based on the definition of PBRM, this dissertation presented some proposals to cover missing pieces for the deployment of the PBRM method in each field of inter-domain and intra-domain networks. In comparison with previous works, two solutions were proposed. The proposals for each network were presented, then the design and implementation of the PBRM method were produced.

In the future, a more abstract description language is required for inter-domain network operation. As mentioned in Chapter 3, an operator's policy is described in RPSL. Although RPSL is designed to express a management policy, we have to describe actual route behaviours in specific syntax such as exporting and importing route information, and configuring the values of local_preference and MED. To control router behaviours intuitively, a more abstract policy description language is required.

6.1 Contribution to Inter-Domain Network

For inter-domain network operation, *Policy Check Server* is proposed, which investigates the consistency of routing policies in IRR databases, and it provides a method for operators to check whether the policy is consistent with policies of other ASs. As a key finding of investigation of IRR servers, a significant proportion of the AS policies are either specified incorrectly or outdated. In addition, IRR Object Garbage Collector was proposed, which is a mechanism to decrease unsound policies.

Policy Check Server and IRR Object Garbage Collector were implemented and currently in operation at JPNIC as its first step of deployment. As a result of our activities as JPNIC IRR Planning Team, we found that inconsistencies in JPIRR were less than other IRRs in the world, and most objects were updated in one year at least.

As the future work of Policy Check Server, a little consideration of the aggregation algorithm of the Unified IRR Database is required. Currently, Database Checker discards duplicated entries on multiple IRR databases, therefore other algorithms which take in both of duplicated entries should be provided.

6.2 Contribution to Intra-Domain Network

In intra-domain network, on the assumption that there is no previous work which reflect an operator's genuine requirements such as the PBRM, we presented *policy-based link cost determination algorithm for OSPF.* We designed and implemented a system which has functions of inconsistency resolution and policy conversion. The system employs the Dijkstra algorithm and linear programming resolution algorithm to resolve multidimensional policies of each pair of object nodes. Finally, we demonstrated experimentally how the operator burden was reduced, and the scalability of the system for a large scale network was shown.

The main objective of this proposal was to implement a concept of arbitrary routing management in OSPF network operation. This research derived an objective function which solves multidimensional requirements from multiple objective nodes. As a result, operators can change the configuration of route selection only by providing simple *Policy.*

We should notice that this proposal involves a certain measure of rounding error to satisfy multidimensional requirements in the process of policy conversion. As the future work of this research, it is required to improve the integrity of derived costs.

Acknowledgements

I wish to express my gratitude to professor Suguru Yamaguchi, my advisor and committee chairman, for his support, advice and encouragement. Without his kind cooperation, I could not complete this dissertation.

I would also like to thank professor Jun Murai of Keio University, for his helpful support on my committee. His useful comments helped me to finish my work.

I wish to thank professor Hideki Sunahara for his long sustained support to my work. His kind support encouraged me to make my research a fruitful one.

My special thanks go to associate professor Youki Kadobayashi for his continuous support and assistance with his great patience. There are no words to express how grateful I am for his help.

I also thank associate professor Katsuyoshi Iida, assistant professor Takeshi Okuda and assistant professor Shigeru Kashihara for their friendly assistant. Their continuous stimulation pushes me to the goal.

I also would like to offer Professor Ian R. L. Smith my hearty thanks for his constant assistance. An encounter with him is an essential part of my research life, and his smile makes me comfortable.

I would like to thank all members of IPLAB for their friendships, supports and comments. Graduate school and life in general would not be the same without them.

Specially, I would like to offer Mr. Teruaki Yokoyama, Mr. Hiroaki Hazeyama and Mr. Ryo Kaizaki of Keio University my soulful thanks for their help. As brothers in my doctoral life, moreover, I am thankful for their precious comments.

I also thanks Mr. Mio Suzuki, Mr. Omar Ismail and Mr. Toshio Shimojo who are doctoral candidates in the laboratory, and also Ms. Natsue Tanida, the secretariat of the laboratory.

Furthermore, I would like to thank Mr. Kuniaki Kondo, Mr. Tomoya Yoshida, Mr. Katsuyasu Toyama, Mr. Junichi Matsumoto, Dr. Kengo Nagahashi, Mr. Taiji Kimura and Mr. Hiroki Kawabata who are the members of JPNIC IRR

Planning Team for their help. Without their presence, I could not persevere in my research up to the present.

I also would like to thank Ms. Keiko Nakano, my fiancee who gave me great help for public and private affairs.

Finally, I wish to thank my family who always be there for me and cheer me up every time I am exhausted and want to give up. Their sincere encouragement, their loves and support enable me to go through my degree.

References

[1] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpastra. Routing policy specification language (rpsl).
http://www.ietf.org/rfc/rfc2622.txt, June 1999.

[2] P. Flegkas, P. Trimintzios, G. Pavlou, I. Andrikopoulos, and F. Cavalcanti. On policy-based extensible hierarchical network management in qos-enabled ip networks. *Proceedings of the IEEE Workshop on Policies for Distributed Systems and Networks (POLICY ' 01), Bristol,UK, M. Sloman, J. Lobo, E. Lupu, eds., Springer, January 2001, pp. 230-246.*

[3] S.Herzog, J.Boyle, R.Cohen, D.Durham, R.Rajan, and A.Sastry. Cops usage for rsvp. *RFC 2749*, January 2000.

[4] J.Boyle, R.Cohen, D.Durham, S.Herzog, R.Rajan, and A.Sastry. The cops (common open policy service) protocol. *RFC 2748*, January 2000.

[5] K.Chan, J.Seligson, D.Durham, S.Gai, K.McCloghrie, S.Herzog, F.Reichmeyer, and A.Smith R.Yavatkar. Cops usage for policy provisioning (cops-pr). *RFC 3084*, March 2001.

[6] Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing OSPF weights. In *INFOCOM (2)*, pages 519–528, 2000.

[7] U. Killat E. Mulyana. A hybrid genetic algorithm approach for ospf weight setting problem. In *Proceedings of the 2nd Polish-German Teletraffic Symposium PGTS 2002*, Gdansk Poland, September 2002.

[8] S. A. Misel. Wow, AS7007!. NANOG mail archives.
http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html, April 2001.

[9] J. Farrar. C&W Routing Instability. NANOG mail archives.
http://www.merit.edu/mail.archives/nanog/2001-04/msg00209.html
, April 2001.

[10] R. Mahajan, D. Wetherall, and T. Anderson. Understanding bgp misconfiguration. *SIGCOMM'02*, August 2002.

[11] T. G. Griffin and G. Wilfong. On the correctness of ibgp configuration. *SIGCOMM'02*, August 2002.

[12] Japan Network Information Center (JPNIC). Jpirr bof the 3rd, internet week 2005.
http://internetweek.jp/bof/03.html.

[13] Y. Hori, Z. Ikenaga, Y. Kadobayashi, and S. Goto. *Interconnection of Networks.* Iwanami Shoten, 2001. (in Japanese).

[14] Asia Pacific Network Informatin Centre. *http://www.apnic.net/.*

[15] E. Gunduz, S. Kerr, A. Robachevsky, and J. L. S. Damas. Routing registry consistency check. Technical report, RIPE NCC, December 2001.

[16] Routing Arbiter Project. Ratoolset.
http://www.isi.edu/ra/RAToolSet/.

[17] IRRToolSet. *ftp://ftp.ripe.net/tools.IRRToolSet.*

[18] R. Chandra P. Traina. *BGP Communities Attribute.* RFC 1997, August 1997.

[19] Silva Damas Joao Luis and Andrei Robachevsky. *RIPE Database Reference Manual.* ripe-251, July 2002.

[20] Antony Antony and Henk Uijterwall. *Routing Information Service R.I.S. Design Note.* ripe-200, October 1999.

[21] E.Davies H.Berkowitz and L.Andersson. *An Experimental Methodology for Analysis of Growth in the Global Routing Table.* Internet-Draft, Working in Progress, 2001.

[22] J. Case, M. Fedor, M. Schoffstall, and J. Davin. A simple network management protocol (snmp).
http://www.ietf.org/rfc/rfc1157.txt, May 1990.

[23] J. Moy. Ospf version 2.
http://www.ietf.org/rfc/rfc2328.txt, April 1998.

[24] B. Lavery H. Pham. An improved method for determining link weights for optimising ospf routing. In *WITSP 02 Proceedings*, pages 112–116, New South Wales, Australia, 2002.

[25] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, 1963.

[26] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 1985.

[27] D Baraff. Curved surfaces and coherence for non-penetrating rigid body simulation. In R. J. Beach, editor, *SIGGRAPH '90 Proceedings*, pages 19–28, Dallas, Texas, 1990. ACM, Addison-Wesley.

[28] Appan Ponnappan, Lingjia Yang, and Radhakrishna Pillai.R. A policy based qos management system for the intserv/diffserv based internet. *Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (POLICY ' 02). IEEE*, 2002.

[29] T. Ebata, M. Takihiro, S. Miyake, and M. Koizumi. Inter-domain qos provisioning and accounting. *INET2000 Proceedings*, Internet Society (ISOC).

[30] H.Zhang. Service disciplines for guaranteed performance service in packet switching networks. *Proc. IEEE vol.83 pp.1374-1396*, Oct. 1995.

[31] R.Yavatkar, D.Pendarakis, and R.Guerin. A framework for policy-based admission control. *RFC 2753*, January 2000.

[32] Kengo NAGAHASHI. *Reseach into the the Hieratical Registry System.* Under graduated Thesis, Keio University, 2000.

Appendix

A. List of Publications

A.1 Journal

1–1. Masasi Eto, Youki Kadobayashi and Suguru Yamaguchi, "Improvement of Consistency among AS Policies on IRR Database", IPSJ Journal, Vol. 46, No 6, pp. 1456–1465, June 2005.

1–2. Masashi Eto, Youki Kadobayashi and Suguru Yamaguchi, "A Proposal and Implementation of Policy-Based Cost Assignment Algorithm for Link State Protocols", IEICE *Special Section on Internet Technology VI*, (Submitted).

1–3. 下條敏男, 衛藤将史, 門林雄基: "ポリシによる複数ファイアウォールの一括設定方式の提案と実装", 電子情報通信学会論文誌, Vol. J87-B, No. 10, pp. 1616–1625, 2004年10月.

A.2 International Conference

2–1. Masasi Eto, Youki Kadobayashi, and Suguru Yamaguchi, "Improvement of Consistency among AS policies on IRR Database", In *Proceedings of TERENA Networking Conference 2003*, Zagreb, Croatia, May 2003.

2–2. Masashi Eto, Youki Kadobayashi and Suguru Yamaguchi, "A Proposal and Implementation of Policy-Based Cost Assignment Algorithm for Link State Protocols", In *Proceedings of TERENA Networking Conference 2005*, Poznan, Poland, June 2005.

2–3. Omar Ismail, Masashi Eto, and Youki Kadobayashi, "A Proposal and Implementation of Automatic Detection/Collection System for Cross-Site Scripting Vulnerability", In *Proceedings of the 18th International Conference on Advanced Information Networking and Applications (AINA2004)*, Fukuoka, Japan, March 2004.

A.3 Technical Report

3–1. 衛藤将史, 森島直人, 門林雄基: “自律システム相互接続点における経路制御運用の自動化の為の技術要素の提案”, 電子情報通信学会 インターネットアーキテクチャ研究会, 2002 年 7 月.

3–2. 夏山京大, 衛藤将史, 門林雄基, 山口英: “ポリシに基づく IGP コスト決定手法の提案”, 電子情報通信学会技術研究報告, Vol. 103, No. 354, IA2003-33, pp. 55–60, 2003 年 10 月.

3–3. 太田政宏, 横山輝明, 衛藤将史, 門林雄基, 山口英: “遠隔授業における共同レポート作成のための議論支援システムの実装と評価”, 電子情報通信学会技術研究報告, Vol. 103, No. 605, IA2003-38, pp. 13–18, 2004 年 1 月.

3–4. 中村真也, 衛藤将史, 門林雄基: “情報の二次利用ポリシを記述可能なメールシステムの提案と実装”, 電子情報通信学会情報セキュリティ研究専門委員会 暗号と情報セキュリティシンポジウム, 2005 年 1 月.