Experiences in Emulating 10K AS Topology with Massive VM Multiplexing

Shinsuke Miwa
NICT*
4-2-1 Nukuikita-machi,
Koganei, Tokyo, Japan
danna@nict.go.jp

Mio Suzuki
NICT*
mio@nict.go.jp

Hiroaki Hazeyama
NAIST†
8916-5 Takayama,
Ikoma, Nara, Japan
hiroa-ha@is.naist.jp

Satoshi Uda
JAIST‡
1-1 Asahidai,
Nomi, Ishikawa, Japan
zin@jaist.ac.jp

Toshiyuki Miyachi
NICT*
miyachi@nict.go.jp

Youki Kadobayashi
NAIST†/NICT*
youki-k@is.aist-nara.ac.jp

Yoichi Shinoda
NICT*/JAIST‡
shinoda@nict.go.jp

ABSTRACT

New technologies that will be introduced to the Internet should be practically tested for effectiveness and for side effects. A realistic environment that simulates the Internet is needed to experimentally test such technologies, which will be widely deployed on the Internet.

To support experimentation in a realistic, Internet-like environment, we are now trying to construct an Internet on a testbed. We describe our method of constructing an Internet-like environment on the testbed using a virtualization technology and estimation of the inter-AS network on StarBED with Xen and our prototype system. We stably constructed a 10,000-AS network using 150 testbed nodes and estimated its performance and feasibility.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: [Network management, Network monitoring, Public networks]

General Terms

Experimentation, Verification

Keywords

Network Emulation, Testbed, virtualization, Internet, BGP

*National Institute of Information and Communications Technology, Japan.
†Nara Institute of Science and Technology, Japan.
‡Japan Advanced Institute of Science and Technology.

VISA'09, August 17, 2009, Barcelona, Spain.

1. INTRODUCTION

When a new technology is introduced to the Internet, which is a very large scale distributed environment, it must be thoroughly tested using an actual implementation of its for precision to make sure that it operates as designed and that it has no adverse effects on other parts of the Internet. These tests require an experimental environment similar to the real Internet. The Internet currently consists of about 500 millions hosts[1], and links between each host and router. The Internet can be divided into operational units called autonomous systems (ASs), and the network parts of the Internet consist of intra-AS networks and inter-AS networks. There are over 30,000 advertised ASs[2] on the Internet.

To emulate the Internet for an experiment, the target hosts of the experiment and the intra-AS networks that include target hosts and the required parts of the inter-AS network, which include all the routes between target ASs, must be emulated. Using this model, to thoroughly test an implementation, we tested the construction of the emulated Internet environment on a testbed, which is used instead of the real Internet in experiments.

In this paper, we discuss a method of constructing a emulated inter-AS network environment on a testbed. We then report experimental construction of the inter-AS network on our testbed and estimate its performance and feasibility. We use the following terminology in this paper: elements of the testbed are referred to as *node*s or *testbed node*s; elements of a emulation in our experiment are referred to as *experimental node*s; networks using experimental communications such as emulated inter-AS communications are referred to as *experimental network*s; and networks used for management are referred to as *management network*s.

2. RELATED WORK

Large-scale inter-AS network simulations using network simulators such as the ns-2[3] and the GTNetS[4] have been carried out. Also, to experiment using a real implementation of the BGP router in network simulators, previous researchers have tried simulated routers such as the BGP++[5], which is based on the Zebra[6] `bgpd`. Some BGP-related experiments have also been conducted using large-scale inter-AS network emulation[7] on the DETER testbed[8], which is

based on ns-2 and NSE. However, these environments using simulators or emulators were not suitable for our experiment, which requires verification of actual implementation and observation of conversations with actual packets because network simulators and emulators can only provide abstract networks with naturally low fidelity.

We have been developing **StarBED**[9] as a testbed for practical network system-related experiments. StarBED consists of 830[1] physical PC servers being used as nodes. These are connected to other nodes via Ethernet,and the network topology of nodes are capable of flexibly configuring that using Ethernet VLAN. Moreover, some experiments using virtualization technologies to increase the scale of the experiment have been conducted[10] on StarBED. This shows that StarBED is suitable for our requirements, and we will discuss below how to construct the emulated inter-AS network on StarBED.

3. CONSTRUCTING THE INTER-AS NETWORK

This section discusses automatic construction of the emulated inter-AS network on StarBED by modeling AS and the inter-AS network, architecture of our method.

3.1 Modeling AS and Inter-AS Network

An AS on the real Internet usually consists of an intra-AS network, which includes many hosts and routers, some AS border gateway routers and some network links to other ASs. On our emulated inter-AS network, the AS is simply emulated by a single quagga[11] router and network links on an experimental node.

For the topology of the inter-AS network to resemble that of the real Internet, topology information derived from the real Internet must be used to make topology of the emulated inter-AS network. We have selected AS Relationships files, which are referred to as *as-rel*, of the CAIDA AS Ranking project[12] to make the topology of our emulated inter-AS network. We chose them because as-rel files of the CAIDA AS Ranking project are recorded as RouteViews[13] BGP AS links, which have been observed in the real Internet, and are annotated with inferred relationships. As-rel files show lists of pairs of AS numbers and their relationships, such as customer, provider, peer, and sibling.

It is not feasible for 30,000 physical testbed nodes to be used to emulate each AS on the inter-AS network emulated on a physical testbed node, and simulators do not provide a high degree of fidelity for verifying actual implementations. Therefore, a virtual testbed node should act as an experimental node on our emulated inter-AS network, using virtualization technologies such as hardware and OS virtualizations because many experimental nodes should be emulated on a physical testbed node.

Furthermore, each network link between an AS and another AS on the real Internet has performance characteristics such as bandwidth, delay, jitter, and packet loss rate. Moreover, an AS often has multiple network link to the another AS for redundancy. It should be noted that we could not address these characteristics in the current prototype, because we have no method of inferring performance characteristics and redundant links.

[1]At present, StarBED has 1,070 physical PC servers.

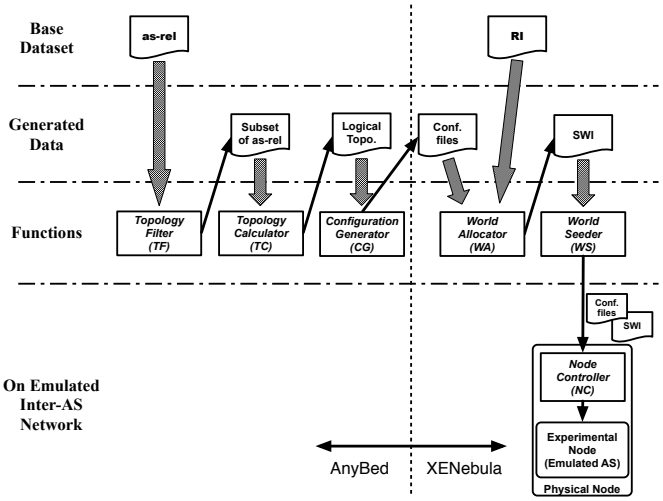

Figure 1: Architecture

3.2 Architecture

Figure 1 shows the architecture used to automatically construct the emulated inter-AS network on a testbed. The function of each element is discussed below:

Testbed Nodes Resource Information (RI) is a list of resource information of a testbed node on a testbed such as processor specifications, memory capacity, hard disk drive specifications, network interface specifications, etc.

AS Relationship Information (as-rel) is a list of relationships of one AS to other ASs, such as providers, customers, peers, and siblings.

Topology Filter (TF) shapes a subset from AS relationship information according to a topology filtering rule.

Topology Calculator (TC) extracts logical topology information from the subset of the AS relationship information. The logical topology information must bind a emulated AS and an experimental node.

Configuration Generator (CG) generates configuration files for each experimental node, such as network configurations and router configurations with route filters, based on the logical topology information.

World Allocator (WA) allocates all experimental nodes to testbed nodes according to an allocation policy and the testbed nodes resource information. A set of experimental nodes, which is within the testbed node, is called a *small world*. **WA** also records allocation to small world information (**SWI**).

World Seeder (WS) initializes all testbed nodes, dispatches small worlds to the bound testbed nodes, starts up **NC** for small worlds, and controls them.

Node Controller (NC) initializes all experimental nodes on small worlds, configures all of them, starts up experimental nodes on small worlds, and controls them.

4. PROTOTYPE SYSTEM

We have been implementing a prototype system according to this architecture on StarBED with our developed tools, which are called **AnyBed**[15] and **XENebula**. **TF**, **TC**, and **CG** have been implemented using AnyBed tools. **WA**, **WS**, and **NC** are called XENebula tools.

We already conducted and reported some experiments of emulated inter-AS network on physical testbed nodes using AnyBed[16] without the virtualization technology and XENebula tools. So, this section focused on XENebula tools.

4.1 Prototyping

This subsection describes some details of implementation of the prototype system.

4.1.1 *Base Dataset*

A base dataset is inputted from a user on a prototype system consisting of a **RI** file, a **vRI** file, and an **as-rel** file.

The **RI** file must be formatted according to the format of XENebula resource information, which is a list of resource information of StarBED physical nodes: the IP address of management network interface, identifier number, memory capacity, hard disk device name, host name, experiment network interface name, and management network interface name. The **RI** file was transformed from StarBED resource information.

The **vRI** file is a resource information file of virtual testbed nodes, which must be formatted according to the format of AnyBed facility information for XENebula: virtual node name, guest OS type, network interface specifications and name, and IP address of management network. The **vRI** generator generates a **vRI** file based on the required number of virtual nodes.

The **as-rel** file must be formatted according to the format of CAIDA as-rel file, which is a list of pairs of AS numbers (below, the former AS is AS1 and the latter AS is AS2) and their relationship. The relationship is a number that can be taken to be -1 if AS1 is a customer of AS2 or 0 if AS1 and AS2 are peers, 1 if AS1 is a provider of AS2, 2 if AS1 and AS2 are siblings.

4.1.2 *Route Filter by* **CG**

A route filter on the `bgpd.conf` is simply inferred from AS relationships by **CG** as:

- When there is a provider relationship, a route from a related AS (a provider) is marked as a default local preference when importing the route, and routes that are originate from it and are imported from customers are exported to the provider.
- When there is a customer relationship, a route from a related AS (a customer) is marked as the highest local preference when importing the route, and all routes are exported to the customer.
- When there are peer and sibling relationships, a route from a related AS (a peer) is marked as a higher local preference when that route is being imported, and routes that are originate from it and are imported from customers are exported to the peer.

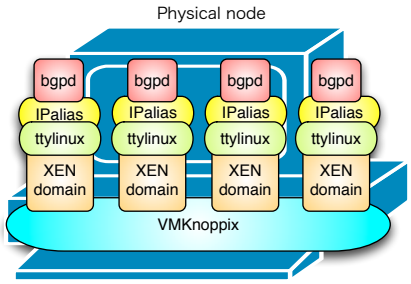

Figure 2: Structure of Physical Testbed Node

4.1.3 *Allocation Policy on* **WA**

The current allocation policy on **WA**:

0. counts neighbor ASs on `bgpd.conf` for each experimental node,
1. selects an experimental node, which has most neighbor ASs, and a physical StarBED node, which has the most remaining memory capacity,
2. calculates the required memory of the experimental node, which is designated base memory size (default value is 24 MB) divided by designated ratio (default value is per 25 neighbor ASs),
3. allocates the experimental node to the physical node and subtracts the required memory from the remainder of the memory capacity of the physical node,
4. if the required memory exceeded designated maximum memory allocation size of single experimental node (default value is 1024MB) which for a performance arrangement, the remainder of the memory capacity of the physical node sets to 0 to avoid more allocation,
5. repeats the process from step 1 excluding the allocated experimental nodes.

4.1.4 *Experimental Node*

Figure 2 shows overview of each physical testbed node. On the prototype system, modified VMKnoppix[17], which is a live CD package based on Debian Linux and includes many virtualization technologies, has been modified to be network bootable for the OS of physical nodes. Physical nodes can boot up the modified VMKnoppix via PXEboot with a NFS without a hard disk-installed OS. Each virtual testbed node for emulating an experimental node will be virtualized as a ttylinux[18] host with `bgpd`, `ospfd`,[2] and `zebra` of the quagga and some tools using Xen[19].

Manipulations of a physical testbed node and a virtual testbed node can be provided via a proper network interface for management network, which must be prepared in a manner different from that used to prepare a network interface for emulating network links of the AS.

[2] `ospfd` could not be used in the current trial because an AS only consists of an experimental node, but the **CG** could generate routing daemon configurations on the AS, which consist of many border gateway routers. Therefore, `ospfd` can be used for intra-AS propagations.

Each network link is allocated to an IEEE802.1q VLAN for early implementation, but this method cannot accept enough network links because the maximum number of IEEE802.1q VLAN tags is 4,096. Therefore, all network links will be allocated on a single flat VLAN, and each network link is separated by IP address ranges using an IP alias on the current implementation.

4.2 Construction Steps on the Prototype

In the prototype system, an emulated inter-AS network can be automatically constructed on StarBED in the following steps:

0. A base dataset must be inputted from a user.
1. The **TF** shapes a subset of the as-rel file based on topology filtering rules. The prototype system provides three types of filtering rule: the top N (designated number) of number of linked neighbors, number of hops from designated ASs, and ASs in Japan using the JPNIC[20] repository.
2. The **TC** calculates logical topology information, which must be formatted according to the format of topology information of AnyBed, from the subset and the vRI file. The logical topology information keeps two bindings which are between an emulated AS and an experimental node, and between the experimental node and a virtual testbed node.
3. The **CG** generates `bgpd.conf` which includes route filters, `zebra.conf`, and `rc.local` which includes an initializing script for all network interfaces, for each experimental node based on the logical topology information.
4. The **WA** allocates each experimental node to a physical StarBED node based on current allocation policy using the `bgpd.conf` and the RI file, as mentioned above, and records the allocation to a SWI file by each physical node.
5. The **WS**: 1) boots up required physical StarBED nodes that were bound to a small world according to the RI file and SWI files, and initializes a VLAN configuration of network switches, which connects them to a single flat VLAN, 2) corrects `ssh` keys, which are used to manage physical nodes, from each physical node via management network, 3) initializes them by mounting a disk device, which is used to keep NC tools and the SWI file for the node, and synchronizing time of day using NTP, and deploying NC tools and configuration files for each experimental node (i.e. the virtual testbed node) to them, 4) and when the initialization is finished, starts up NC for each small world on each physical node via `ssh`.
6. The **NC**: 1) initializes `xend`, which is the control daemon of Xen, and the network bridges of Xen for communication between virtual testbed nodes and other virtual nodes on other physical nodes, 2) when the initialization is finished, generates a Xen disk image file for each virtual testbed node, inserts configuration files for each bound experimental node to the disk image, generates a Xen configuration file (.xm) based on a record into the SWI file for each bound experimental node, 3) and then, starts up each virtual node on the physical node and starts up `bgpd`, `zebra`, and `rc.local` on each virtual node.

Table 1: Specifications of Physical Testbed Node on StarBED

Group		Spec.
F	CPU	Intel Pentium4 3.2GHz ×2 (HT)
	Memory	8GB
	NIC	1000Base-T×6
	HDD	SATA 80GB×2
G-1	CPU	AMD Opteron 146 HE 2.0GHz
	Memory	8GB
	NIC	1000Base-T×2
	HDD	(none)

After these steps, all emulated BGP routers communicate with other BGP routers for propagating inter-AS routing information. In other words, the emulated inter-AS network has been constructed.

5. EXPERIMENT AND EVALUATION

We also conducted experiments to construct an emulated inter-AS network using the prototype system. All experiments were conducted using Group F and Group G-1 as physical testbed nodes on StarBED. Specifications for each node are shown in Table 1.

In this section, we discuss the following three experiments:

- "Feasibility Test" was conducted to confirm feasibility of our method.
- "Maximum Scale" is the latest maximum AS emulated inter-AS network.
- "Merging into the real Internet" was conducted to estimate stability and performance of feasible size of an emulated inter-AS network.

5.1 Feasibility Test

We attempted to construct an emulated inter-AS network, consisting of the top 250 ASs in terms of number of linked neighbors from the CAIDA as-rel file for April 30, 2007 using the prototype system. We successfully emulated the top 250 ASs on five physical StarBED Group F nodes. To confirm its feasibility, the performance and throughput of the prototype system was measured on the emulated inter-AS network.

Round trip time (RTT) and bandwidth were measured to evaluate the difference between the performance and throughput of an actual PC router environment and that of the emulated inter-AS network. Performance and throughput were measured in three different environments (Figure 3): physical node A connected to physical node B,

1) without any routers.
2) via five actual PC routers.
3) via five emulated routers on the emulated inter-AS network.

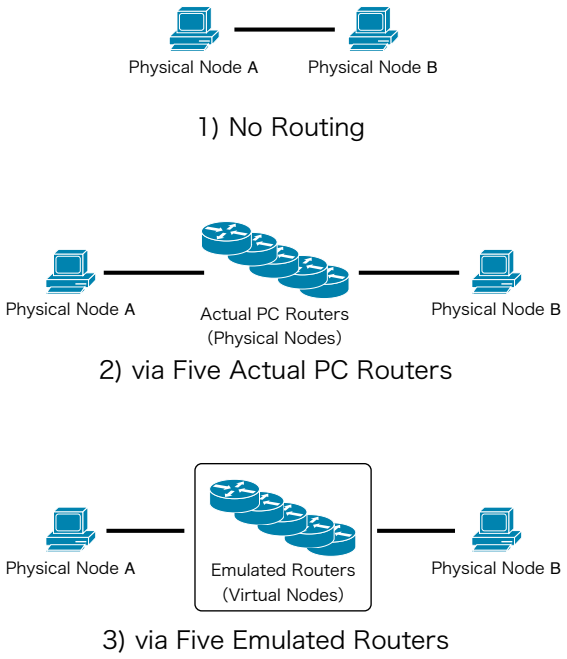

Figure 3: Measurement Environments

Table 2: Measurement Results

Environment	`ping` Avg.(ms)	`iperf` (Mbps)
No Routing	0.122	961
via five Actual PC Routers	0.936	961
via five Emulated Routers	1.489	160

RTTs were measured using `ping` commands from physical node A to physical node B, and bandwidths were inferred by `iperf`[21] from physical node A to physical node B. Table 2 shows the results of measurements. The `ping` results are averages of 100 trials each, and the `iperf` results are averages of 10 trials each.

These results show that RTT is about 0.1 ms in environment 1, about 1.0 ms in environment 2, and about 1.5 ms in environment 3, which shows that RTT is affected by the virtualization of the prototype system. We thought that this was not a very large difference for inter-AS communications. Bandwidth in environment 1 is about 960 Mbps, in environment 2 is about 960 Mbps, or about the same as in environment 1, and in environment 3 is about 160 Mbps. This shows that bandwidth can be impacted by virtualization. We thought that impacts on RTT and bandwidth were caused by software implementation of network bridge which used by network interfaces of Xen domU. We also think that we can satisfactorily conduct experiments that require bandwidth of 100 Mbps using physical nodes with a 1000 Mbps network interface.

5.2 Maximum Scale

We also attempted two experiments which would have emulated the maximum scale of the inter-AS network, which consists of the top 10,000 ASs from the CAIDA as-rel file of October 29, 2007. We successfully emulated the top 10,000 ASs. This is the largest scale of the emulated inter-AS network using our prototype system.

On the first experiment, the top 10,000 ASs were emulated on 140 physical StarBED Group F nodes with non-default values for the **WA**: base memory size is 72MB per 25 neighbor ASs and dom0 memory size is 1024MB, because the top 10,000 ASs emulation with default values could not be successfully conducted. We successfully emulated in the experiment, though unstable behaviors were observed in some virtual nodes. According to our analysis using log files of `xend`, we found that unstable behaviours were caused by two problems. First problem is no more loopback device for `vbd`. `Vbd`, which is the back-end device node of Xen, on some testbed nodes could not respond to domU because number of loopback device were exceeded its limit, so `xend` shutdowns the domUand re-create the domU, repeatedly. Second problem is unsuitable allocation to a domU on some dom0. The domU could not handle its `bgpd` because allocated memory size is not enough, so domU were crashed and `xend` re-create the domU, repeatedly.

According to this result, we modified the kernel module for loopback device and the **WA**. The kernel module for loopback device were modified its limit of number of device, which restricted to 256 on normal module, to unlimited. The **WA** were modified its memory allocation policy to memory allocation size is multiplied a normal memory allocation size by 1.5. So, we attempted second experiment. On the experiment, the top 10,000 ASs were emulated on 150 physical StarBED Group F nodes with the modified loopback module and the modified **WA** with same values in the first experiment. We successfully emulated in the experiment, and we also observed no unstable behaviors concerning the virtual node. Although, some unstable behaviors of bgp routing such as inexplicable dissipation of routes and route flapping were observed. So we have to stabilize them by developing health check mechanism for bgp routing.

5.3 Merging into the real Internet

We also attempted to operate a network that emulated the inter-AS network of Japan, which consists of 445 ASs on the JPNIC AS Numbers file of August, 2008 and on the CAIDA as-rel file of July 21, 2008 on ten physical StarBED Group G-1 nodes with a NFS server that provides Xen disk images of virtual nodes, and the network merging into the real Internet environment at WIDE Project[22] Autumn 2008 Camp, which was held for 3-days.

Figure 4 shows topology of the experiment. The emulated inter-AS network was merging between the entry point (WIDE-BB in the figure) of the Internet of WIDE Project and WIDE camp network (ESSID WIDE in the figure), so the network had to carry all incoming and outgoing traffic between WIDE camp network and the Internet. Figure 5 shows incoming and outgoing traffic graph of the emulated inter-AS network. Left side of graph shows traffic from edge AS, which directory connected to WIDE camp network (ESSID WIDE), and right side of graph shows traffic to the other edge AS, which connected to the Internet (WIDE-BB). The graph said that all traffic stably passed through the emulated inter-AS network without significant loss, because incoming traffic of left side similar to outgoing traffic of right side, and outgoing traffic of left side also similar to incoming traffic of right side in the graph.

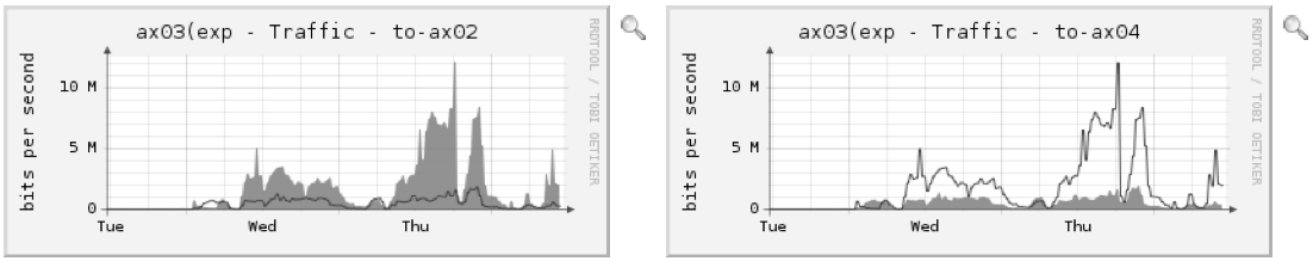

Figure 5: In/Out Traffic of Emulated inter-AS network (3-days)

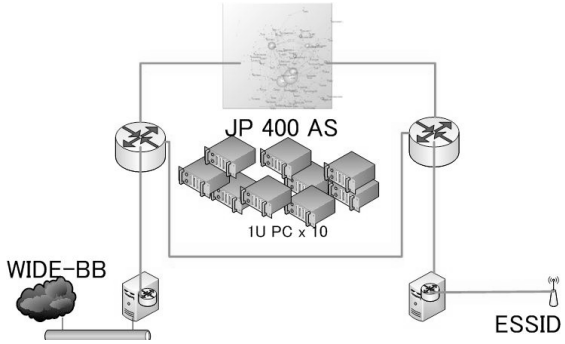

Figure 4: Topology of the experiment

We successfully and stably operated the network on 3-days without any troubles, then we can conclude emulated inter-AS network using our method is feasible in the case of a nation size.

6. FUTURE WORK

In this section, we describe our plans for future work on the prototype system.

6.1 Single Flat VLAN

All network links on physical nodes, which involve all inter-AS links, are connected into a single flat VLAN, and are separated by IP address ranges using IP aliases on virtual nodes in the current version of the prototype system. Therefore, we discovered below issues that have to be resolved.

Issue 1: An enormous number of ARPs can flood the flat VLAN because ARP requires broadcast communication on a network segment, i.e. the flat VLAN.

Issue 2: It is not easy to emulate inter-border gateway communication on an AS and on intra-AS router communication using IP aliases and the flat VLAN because inter-router communications could not be separated on each link in the flat VLAN when another routing protocol using multicast or broadcast of the link layer, such as OSPF, has to be used.

Issue 3: The performance characteristics of each link cannot be emulated, because `netem`[14], which we plan to use to emulate the characteristics of each link, cannot handle IP alias interfaces.

The Issue 1 could be simply solved by configuring static ARP entries, which are list of pairs IP address and MAC address of all neighbor nodes, to all experimental nodes. So, we plan to improve our **CG** to resolve the static ARP entry of each neighbor node from the vRI file.

Issue 2 and Issue 3 could be solved by using more VLANs, but this method cannot accommodate enough network links because of the limitation of number of IEEE802.1q VLAN tags as mentioned in section 4.1.4. To solve this network links issue, we have to reduce number of bridging AS network links between all physical nodes. Therefore, we plan to develop an algorithm for **WA** which is capable of partitioning inter-AS topology according to requirements, that are: reducing cutting edges, limiting maximum size of subgraphs, and limiting maximum number of vertex into subgraphs, based on some graph partitioning algorithm[23].

6.2 Allocation of Nodes

Our default allocation policy for the **WA** cannot always efficiently allocate virtual nodes to physical nodes because the allocation policy is based on a law learned by experience from results of some preliminary experiments without guarantees. Moreover, a physical node that has leftover memory capacity would be simplistically selected as a candidate for the next allocation. Therefore, characteristics such as the average processor load and the bandwidth usage of the network interface of the physical node could not be regarded.

We are now discussing two approaches to improve the **WA** allocation mechanism. First approach is precisely estimating resource requirements of each experimental node. Second approach is dynamically allocating and relocating an unstable experimental node to a physical testbed node which might be able to afford to conduct the experimental node.

6.3 Model of AS

The Model of AS was simplified on our emulated inter-AS network as consisting of a single AS border gateway router and some network links to other ASs. This simplification would make it difficult to emulate ASs connected by multiple links via multiple border gateway routers. Multiple border gateway routers on a single AS have to be emulated when a target experiment would require high fidelity emulating about network links, because they are popular for redundantly connecting to other ASs.

Although the **CG** could generate routing daemon configurations on an AS which consist of many border gateway routers, the **TF** and the **TC** could not address multiple border gateway routers on the AS because the **as-rel** file from CAIDA AS Ranking project, which is a base dataset for the

topology of our emulated inter-AS network, is only recorded relationships of each AS and other ASs but the file is not also listed relationships of each border gateway router and other routers.

Furthermore, the routing policy of each border gateway router on the emulated inter-AS network might not be accurately emulated, because a route filter on each `bgpd.conf`, which was generated by the **CG** for implementing a routing policy on the border gateway router, is simply inferred from AS relationships.

We think that the original RouteView dataset and the Routing Assets Database (RADb)[24] could help us to infer relationships of border gateway routers and their policies.

6.4 Emulating Other Part of the Internet

The goal of constructing the emulated Internet is to emulate entire of elements on the real Internet on our testbed, though we have been trying on constructing emulated inter-AS network. As mentioned above, the network part of the Internet were composed by an inter-AS network and many intra-AS networks. We are aware that major reminder part of our Internet emulation is the intra-AS network emulation.

The intra-AS network emulation would require many observations of the real intra-AS networks such as Rocketfuel[25] to imitate an emulated intra-AS network as the real one, and capable of emulating more complicated network, because intra-AS networks have their respective topologies and various constructions. We have to discuss on how we can get to the topology of a typical intra-AS network, and how we can emulate it.

Furthermore, we are aware that other reminder part of our Internet emulation is the emulation of fundamental services such as DNS. We are now trying to develop "fake DNS service" to provide DNS service on the emulated Internet.

7. DISCUSSION

In this section, we discuss fidelity, scale and tractability of emulation of the Internet. Moreover, we also discuss coverage and target of the emulated Internet.

7.1 Fidelity of Emulation

Though almost all actual implementations on actual PCs can be accurately executed by virtualization technologies, time transition issues and minor differences in environment between actual PCs and using virtualization technologies might cause differences of behavior in the implementation. If higher fidelity were required, some tricks, such as a physical node allocated as an experimental node instead of allocating to a virtual node, must be provided.

We expect that behaviors of a router on the prototype system are different from behaviors of actual router instruments because a virtual node with routing software was implemented as the router on the prototype system. If higher fidelity were required, higher fidelity of router implementation would have to be provided using a technology such as CISCO 7200 Simulator[26], which simulates a router using an actual router OS (CISCO IOS), and Shooner[27], which prepares more actual router instruments for experiments.

We plan to emulate performance characteristics of each network link using `netem`, so the fidelity of the network link emulation will depend on the fidelity of `netem`. Therefore, we have to estimate its fidelity, and if it does not have high enough fidelity, we must somehow provide high fidelity emulation, for example, by connecting actual networks into our emulated inter-AS network to add fidelity. We also have to estimate what degree of fidelity is required in experiments on the emulated Internet.

7.2 Scale of Emulation

The goal of constructing the emulated Internet is to emulate entire hosts, network instruments such as routers, and network links on the real Internet. Therefore, because the emulated Internet is as big as the real Internet, the scale of the emulation may cause some problems.

Emulating a huge-scale environment requires the use of some virtualization technologies for multiplying physical nodes because it is not feasible for the same number of physical nodes as are present in the real Internet to be used in experiments. If the number of physical nodes were fixed, a high degree of abstraction of experimental nodes must be provided to increase the scale of the emulation. Therefore, we will be confronted with a difficult trade-off between increasing the scale of the emulation and decreasing the fidelity of the emulation. Therefore, we will also be confronted with another difficult trade-off between increasing the scale of the emulation and degrading performance because performance depends on the number of virtual nodes on a physical node.

7.3 Tractability of Emulation

To do valid experiments, the experimental environment must be managed based on the requirements for each experiment. However, it is very difficult to manage and observe a distributed system of the scale of the real Internet. To resolve this issue, all physical nodes and virtual nodes should be directly managed and observed via a management network on the prototype system. This will enable us to acquire information about nodes and to operate the nodes. However, managing the emulated Internet is still difficult because of the huge number nodes. Tractability is a very important issue in constructing the emulated Internet.

7.4 Coverage and Target

Though the emulated Internet has some issues as mentioned above, it could be a useful tool to various experiments when fits its construction to targets and purposes of the experiment. For example, we had conducted some experiments[16], which estimates performance of the IP traceback technology against DDoS attacks, on the emulated Internet using no virtual nodes for improving fidelity of emulation, but less scale.

If target instruments or software could be installed on the edge or in the middle of the emulated Internet, many experiments in impossible situations on the real Internet could be conducted on the emulated Internet, because it could be avoid any impact to the real Internet, which is a serious social infrastructure. Therefore, we think that the emulated Internet is suitable to investigating new technologies except technologies in the step of concept verification, which has no implemented instance.

8. CONCLUSION

We described our method of constructing the emulated inter-AS network on StarBED, which will be used instead of the real Internet for experiments that require the precision of actual implementation. We also presented the results of our experiments on the prototype system, which showed that it

can construct an emulated inter-AS network consisting of 10,000 ASs on 150 physical testbed nodes and that it can satisfactorily conduct experiments that require 100 Mbps links.

We are now working to improve our method and have plans for verifying the method with larger-scale experiments. We also plan to emulate intra-AS networks, and hope to emulate the entire Internet using our method of emulating inter- and intra-AS networks.

9. ACKNOWLEDGEMENTS

The authors give special thanks to Ken-ichi CHINEN, an assistant professor at Japan Advanced Institute of Science and Technology (JAIST) for his great implementation of supporting tools (SpringOS) for StarBED, and Satoshi OHTA, a technical expert at National Institute of Information and Communications Technology (NICT) for his support of our experiments.

10. REFERENCES

[1] Internet Systems Consortium, Inc., "ISC Domain Survey: Number of Internet Hosts", `http://www.isc.org/index.pl?/ops/ds/host-count-history.php`.

[2] G. Huston, "The 32-bit AS Number Report", `http://www.potaroo.net/tools/asn32/`, Mar. 2009.

[3] "The Network Simulator - ns-2", `http://www.isi.edu/nsnam/ns/index.html`.

[4] G. F. Riley, "Using the Georgia Tech Network Simulator", `http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/`.

[5] X. A. Dimitropoulos and G. Riley, "Efficient Large-Scale BGP Simulations", Elsevier Science Publishers, Elsevier Computer Networks, Special Issue on Network Modeling and Simulation, vol. 50, num. 12, 2006.

[6] IP Infusion Inc., "GNU Zebra – routing software", `http://www.zebra.org/`.

[7] G. Carl, G. Kesidis, S. Phoha, B. Madan, "Preliminary BGP Multiple-Origin Autonomous Systems (MOAS) Experiments on the DETER Testbed", *In proceedings of DETER community workshop 2006*, Jun. 2006.

[8] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, S. Schwab, "Design, Deployment, and Use of the DETER Testbed", *In proceedings of DETER community workshop 2007*, Aug. 2007.

[9] T. Miyachi, K. Chinen, Y. Shinoda, "StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software", *In proceedings of International Conference on Performance Evaluation Methodlogies and Tools (Valuetools) 2006*, ACM Press, ISBN 1-59593-504-5, Oct. 2006.

[10] E. Muramoto, T. Yoneda, A. Nakamura, M. Misumi, T. Miyachi, Y. Shinoda, "Report on a Method of Simulating Multicast Group Communication on the Internet", *In proceedings of International Sysmposium on Towards Peta-Bit Ultra Networks (PBit) 2003*, ISBN 4-9900330-3-5, pp. 182–188, Sep. 2003.

[11] "Quagga Software Routing Suite", `http://www.quagga.net/`.

[12] Cooperative Association for Internet Data Analysis (CAIDA), "AS ranking", `http://as-rank.caida.org/`.

[13] University of Oregon Route Views Project, "Route Views Project Page", `http://www.routeviews.org/`.

[14] S. Hemminger, "Network Emulation with NetEm", linux.conf.au 2005, Apr. 2005.

[15] M. Suzuki, H. Hazeyama, Y. Kadobayashi, "Expediting experiments across testbeds with AnyBed: a testbed-independent topology configuration tool", *In proceedings of Second International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities* (TridentCom2006), Mar. 2006.

[16] H. Hazeyama, M. Suzuki, S. Miwa, D. Miyamoto, Y. Kadobayashi, "Outfitting an Inter-AS Topology to a Network Emulation TestBed for Realistic Performance Tests of DDoS Countermeasures", *In proceedings of Workshop on Cyber Security Experimentation and Test 2008* (CSET '08), Aug. 2008.

[17] Advanced Industorial Science and Technology, "VMKNOPPIX: Collection of Virtual Machine", `http://unit.aist.go.jp/itri/knoppix/vmknoppix/`.

[18] P. Schmidt, "ttylinux Homepage", `http://www.minimalinux.org/ttylinux/showpage.php?pid=1`.

[19] S. Crosby, D. E. Williams, J. Garcia, "Virtualization With Xen: Including XenEnterprise, XenServer, and XenExpress", Syngress Media Inc.,ISBN 1-597-49167-5, 2007.

[20] Japan Network Information Center (JPNIC), "AS Numbers", `http://www.nic.ad.jp/ja/ip/as-numbers.txt`.

[21] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, K. Gibbs, "NLANR/DAST : Iperf - The TCP/UDP Bandwidth Measurement Tool", `http://dast.nlanr.net/Projects/Iperf/`.

[22] WIDE Project, "WIDE Project", `http://www.wide.ad.jp/`.

[23] G. Karypis and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs", *SIAM Journal on Scientific Computing*, Vol. 20, no. 1, pp. 359–392, Aug. 1998.

[24] Merit Network Inc., "RADb: Routing Assets Database", `http://www.radb.net/`.

[25] N. Spring, R. Mahajan, D. Wetherall, "Measuring ISP Topologies with Rocketfuel", *In Proceedings of SIGCOMM2002*, Aug. 2002.

[26] C. Fillot, "Cisco 7200 Simulator", `http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator`.

[27] P. Barford and L. Landweber, "Bench-style Network Research in an Internet Instance Laboratory", *In Proceedings of SPIE ITCom*, Aug. 2002.