

Realistic Simulation of Internet

Toshiyuki Miyachi^{†‡§} Junya Nakata^{†‡§} Razvan Beuran^{†§} Ken-ichi Chinen^{†‡§}
Kenji Masui^{§‡§} Satoshi Uda^{§‡§} Yasuo Tan^{†‡§} Yoichi Shinoda^{§‡§}

[†]School of Information Science / [§]Center for Information Science

[‡]Internet Research Center, Japan Advanced Institute of Science and Technology

[§]Hokuriku Research Center

National Institute of Information and Communications Technology

ABSTRACT

The Internet is a complex collection of various nodes which run independently. The hardware and software of nodes are quite diverse and these elements make the complexity of the Internet. It is difficult to model these behaviors, so we proposed and implemented StarBED, a large-scale testbed using general PCs and switches. Thus we created an environment for experiment using realistic nodes and software. We also designed and implemented SpringOS, which is a supporting software for making experiments on StarBED. Using SpringOS, users can make experiments on StarBED easily. In this paper, we describe the design of StarBED, and how to make an experiment on the StarBED using SpringOS. Moreover, we describe the design of StarBED2, StarBED2 is a testbed for ubiquitous and sensor networks based on StarBED.

KEY WORDS: Experiment, Verification, Network Testbed

INTRODUCTION

Current Internet has grown as one of the infrastructures for human life. Therefore, running and evaluating newly developed distributed software directly on the current Internet is no longer an option, because they may severely impact the existing crucial services. So, today, the changes and additions to services on the Internet must be tested for certain level of correctness before they are actually deployed, just like with any other social infrastructures, such as electricity, telephone, gas and water.

In order to develop and evaluate the technologies for the Internet, we need a realistic simulated environment of Internet. The best way of realizing this test process is to make an exact copy of the current Internet including people's activities and make changes to this copy, which is next to impossible.

We built StarBED, a facility which has many PCs only for making experiments. We have tried to simulate the Internet using StarBED. In this pa-

per, we describe our approach to simulate realistic Internet.

Our approach achieves a realistic simulation of Internet through the following: (i) use of real nodes in a large scale setup; (ii) flexibility of the test environment; (iii) a powerful management system; (iv) extension capabilities for ubiquitous networks.

REALISM OF INTERNET

In this section, we describe the nature of the Internet and our approach.

The difficulty of simulating the Internet

Fundamental nature of the Internet is a complex collection of autonomously operated networks with so many services running simultaneously. The nodes which belong to these networks are also various, and they use the same protocols to communicate with each other, however, assessing their behavior in detail is difficult including bugs. The complexity of the Internet comes from the

Table 1: PC nodes in StarBED

Client group	Number of nodes	CPU		Memory	Disk Type	Network Interface		
		Type	Clock			GbE	FE	ATM
A	208	Pentium3	1GHz	512MB	ATA	1	0	0
B	64	Pentium3	1GHz	512MB	ATA	0	1	1
C	32	Pentium3	1GHz	512MB	SCSI	0	4	1
D	144	Pentium3	1GHz	512MB	ATA	0	1	0
E	64	Pentium3	1GHz	512MB	ATA	0	1	0
F	168	Pentium4	3GHz	2GB	SATA	4	0	0

variety of elements that form the Internet. Understanding the behavior of these elements independently is not so difficult, however, expecting and modelling behaviors of the network which consists of many of these elements are difficult.

The scale of the Internet is now quite large, which is a reason of the difficulty of the Internet simulation. The result of experiments sometimes varies according to the scale of the environment, however, we often don't know where is the point of changing the result.

Existing Approach

Software simulation is the most popular approach to evaluate networks, ns-2[1] and SSFnet[2] is the well known examples of such simulators. users can make assumed environment for their target using software simulation. The environment is suitable for evaluating the ideas and steps of their target. However, we often cannot use implementations of target technologies directly on the Internet. Since they often require simulator-specific implementations which are different from those which will actually be running on the Internet. The implementations for simulators may differ from those for the Internet. The behaviors of the implementation for software simulators may also be different from those for the Internet, and it is difficult to model the new technologies for software simulation.

Our approach

Our motivation is to make a realistic environment for experiment, and evaluate real software/hardware implementations for the Internet. Therefore, we consider to build the realistic simulation environment using only the realistic elements or implementations on the Internet. Our

approach is to build an environment using many actual nodes like generic PC which is used on the real Internet. We can run real software implementations on these nodes. We assume that many nodes are located on the same site, and all these resources are physically accessible. So we can get all of information about the experiment traffic since all switches that connect experiment nodes are located on the same site. In order to build an Internet-like environment, we need a variety of equipments and installations, which make such kinds of projects quite difficult to achieve. To solve the difficulty of building experiment topologies, the network will be built beforehand based on fixed hardware connections and using virtual networks, customized topology will also be built. These connectivity and switch configuration will be used to produce a target network topology. The experiment can be driven on the virtual environment, thereby we don't have to change the physical topology. We also decided to make our environment as a multi-user system. Nodes are assigned by the administrator to the users.

A simulation environment behaves very similar to real network because of directly using real nodes of real networks. However, it is very difficult to build a large simulation environment by this approach, since it requires huge money and costs such as human resource, time and physical space. To minimize these costs, we are developing a supporting software which helps user to make experiment. A user should write a configuration file, and the experiment will execute following the file. Moreover it will improve the accuracy of experimental results. It can execute user's scenarios according to the description correctly.

We design and implement StarBED and Sprin-

gOS. StarBED is a general purpose network testbed based on many actual nodes and SpringOS is the supporting software for experiments on the StarBED. StarBED was built in 2002 as National Institute of Information and Communications Technology(NICT) Hokuriku IT Open Laboratory, and it is renewed as NICT Hokuriku Research Center with updating the facilities in April 2006.

STARBED AND SPRINGOS

In this section, we describe StarBED and SpringOS. More details on the design of StarBED are described on the StarBED Project website [3] and the design and the behavior of SpringOS are described in [4][5].

Construction of StarBED

At the core of the StarBED, there is an array consisting of about 700 nodes, which are actually standard PCs, with redundant connectivity into a large switch cluster. Table 1 shows the specifications of PC nodes of StarBED, the PC nodes are separated to 6 groups according to the specifications.

We can use a popular end-user operating system such as Windows on these nodes, or software router system based on Unix operating system, or even a special piece of software that turns nodes into delay lines. There are additional but empty positions in the core, where users of the simulator can plug in their own devices to form an appropriate target network. These devices may be the products under test, commercial routers, measurement equipments, and so on. Surrounding the core, there is a dedicated management network that controls and monitors the nodes and switch activities. Nodes can be loaded with appropriate software, controlled and monitored using the management network, without affecting a running simulation. The switch cluster has backdoor connection that can be used to monitor network activities or to create connections to link the core to external world. Figure 1 shows the concept topology of StarBED. By separating the network for management from that for experiment, traffic of management and experiment don't affect each other. Moreover, users can use their desired IP addresses for their experiments. If there

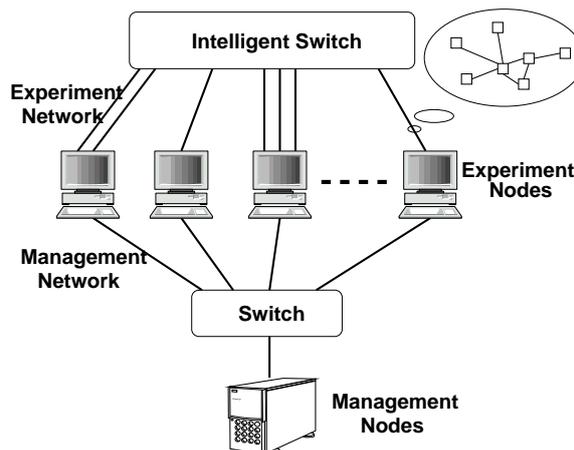


Figure 1: Concept Topology of StarBED

is no fixed network for management, they should set up the IP address via console. This is not a feasible approach when using many nodes.

Making Experiments on StarBED

We implement SpringOS which helps to make experiments on StarBED, and enhance StarBED functionalities. SpringOS is not single module program, it consists of many modules which has special functions. In this subsection, we describe the behavior of SpringOS on StarBED, in order to make experiments.

First, we describe the core technologies for controlling facilities on StarBED.

Power management of nodes

In order to power on PC nodes, we use Wake on LAN based on the magic packet[6] technology. To power off and reboot the nodes, we adopt SNMP. Therefore a SNMP client should run on every node. Nodes of group F support IPMI[7]. Using IPMI, we can control nodes' power state in detail, and we will support IPMI to power management.

Switching boot methods of nodes

Nodes for experiments load OS using the bootloader provided by PXE[8]. The bootloader specifies whether the node starts as diskless system or boots from a partition of local disk. The bootloader configurations are specified by the DHCP server. Before booting up leased nodes, the user

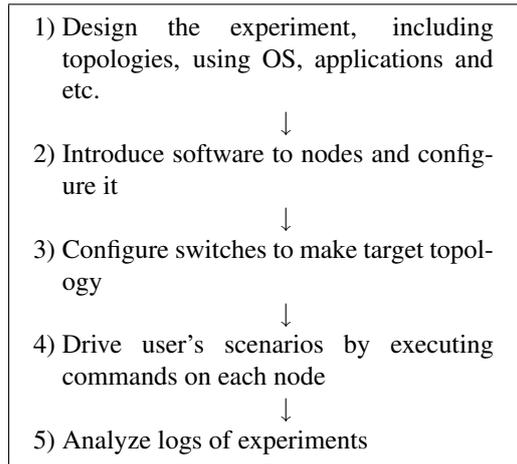


Figure 2: General step of making experiments on StarBED

may setup the PXE so that the node boots from a specified partition and using Wake on LAN to boot up nodes.

Building target topologies

We design StarBED to reduce the costs in building target experiment. Therefore, the target topology of the experiment is configured virtually without changing physical topologies. We can set the desired topology for the experiments using VLAN and ATM VP/VC.

We describe next the main procedures of StarBED and SpringOS. The general step of making experiments is showed with Figure 2. SpringOS helps steps 2)-4), we describe the behavior for that. SpringOS loads the user's configuration file including node definition, target topologies and scenarios for experiment. It executes the steps automatically according to the user's description. Many experiments have been executed on StarBED.

Introduce software to nodes

Almost all of the time, we have the same OS and applications installed on nodes having the same role in the experiment. Therefore, disk images could be created for every role and installed in the node collection having the same role at the same time. In many cases we only need a few disk images to setup all the experiment nodes. Moreover, the setup of IP addresses of each network interface is done according to the configuration file.

When a user needs to boot up nodes it is as a diskless system. The user should prepare the disk images, kernel images and bootloader for their system, and switch the bootloader.

Build the target topology

SpringOS configure the VLAN setting on switches on network for experiment automatically.

Drive user's scenarios

We adopt server-client model to drive user's scenarios. In this paper, we call the server as scenario master, and client as scenario slave. The usual method is to distribute the scenario before the experiment starts by scenario master, and generally each scenario slave on nodes execute the scenario independently. When an experiment node has to synchronize, it connects to the scenario master to mediate with other experiment nodes. When a node should synchronize with other nodes, the scenario master controls the nodes with message passing. The details of the scenario driving of SpringOS, especially concerning synchronization of nodes, are described in [5].

STARBED2

Today various kind of ubiquitous networks including sensor networks and home networks are researched more and more actively, or are already in use. As the ubiquitous networks have been connected to the Internet and introduced to our life, their influence on our life grows.

In order to simulate ubiquitous networks, a testbed has to have some specific properties which are sometimes different from those of the ordinary network simulators. First of all, it is required to support simulation of a great number of heterogeneous nodes connected to each other with various kind of network media. Moreover, it is also needed to have functionalities to simulate surrounding environments since the behavior of applications executed in ubiquitous network system, especially in sensor networks, depends on the information obtained from the surrounding environments. In addition, to have ability to run simulation in real-time is really useful when the simulation is done in a real/virtual mixed environment, that is when simulated nodes cooperate

with real nodes during the simulation.

To implement a testbed which has the functionalities mentioned above is not easy because simulating an ubiquitous network system in real-time is a highly computational-power-consuming task, not only because of the huge number of nodes each network has, but also because of the complexity of the behavior of surrounding environments as well as themselves.

Our approach to implement StarBED2, a ubiquitous network simulator, is to utilize StarBED as its basis. By doing so, StarBED2 can have ability to emulate ubiquitous network system in which thousands of heterogeneous network nodes exist, and accomplishes such a heavy task efficiently.

The major aim of StarBED2 is to create an emulation environment in which various kind of nodes, networks, and environments can be emulated under real-time constraint cooperatively with real nodes so that it can be widely used for prototyping, evaluation and testing of ubiquitous networks at any point of the development phase.

CONCLUSION

Currently the evaluation of new software and hardware for the Internet is very important. However, it is difficult to make realistic environments dedicated to experiments.

Simulating the Internet is difficult because it is very complex and large. In order to realize the complexity, we created a testbed built with network equipment and computers used in real environments. With this approach, users can use realistic hardware and software on it. On the testbed, they can introduce many kinds of applications for the Internet on their target environment, and these many kinds of applications will build the desired complexity. Our approach to build a large-scale environment for experiments is building a cluster composed of many general computers, and preparing supporting software which control the testbed.

We designed and implemented StarBED which is a cluster composed of 680 PCs dedicated to experiments, and SpringOS which is a supporting software for making experiments on StarBED. Using StarBED and SpringOS, we can make a larger topology than other methods, and the re-

sult of the experiments is more realistic. Using SpringOS on StarBED, the steps of experiment are executed automatically. Users should prepare only the description of their experiment.

In order to enable ubiquitous experiments, now we design StarBED2, by enhancing StarBED for realizing a large-scale ubiquitous network emulator. It enables emulation of ubiquitous networks with thousands of heterogeneous nodes.

REFERENCES

- [1] The VINT Project. *The ns Manual*. April 2002. <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [2] *SSFnet network simulator*. <http://www.ssfnet.org>.
- [3] The StarBED Project. <http://www.starbed.org/>.
- [4] Toshiyuki Miyachi, Ken-ichi Chinen and Yoichi Shinoda. Automatic configuration and execution of internet experiments on an actual node-based testbed. In *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities(Tridentcom)*, February 2005.
- [5] Ken-ichi Chinen, Toshiyuki Miyachi and Yoichi Shinoda. A rendezvous in network experiment — case study of kuroyuri. In *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities(Tridentcom)*, March 2006.
- [6] Advanced Micro Devices, Inc. *Magic Packet Technology*, November 1995.
- [7] Intel Corporation. *IPMI v2.0 specifications Document Revision 1.0*, February 2004.
- [8] Intel Corporation. *Preboot Execution Environment (PXE) Specification Version 2.1*, sep 1990.