On Network System Evaluation under Various Failures

Toshiyuki Miyachi*, Razvan Beuran*, Yoshiki Makino*, Kunio Akashi†, Shingo Yasuda†,
Tomoya Inoue†, Shinsuke Miwa*, Satoshi Uda†, Yasuo Tan† and Yoichi Shinoda†
*Hokuriku StarBED Techonology Center
National Institute of Information and Communications Technology, Ishikawa, Japan
Email: {miyachi,razvan,ymakino,danna}@nict.go.jp
†Japan Advanced Institute of Science and Technology, Ishikawa, Japan
Email: {k_akashi,s-yasuda,t-inoue,zin,ytan,shinoda}@jaist.ac.jp

***Abstract*—We always have possibilities of being involved in various network troubles because of system misconfigurations and trouble by natural disasters, etc. So evaluations of network technologies under these situations are important. We are developing a framework for emulating environments with failures and evaluating technologies for recovering from them by utilizing our existing technologies for building large-scale network experimental environments. We successfully conducted several case studies using our technologies with various failure types.**

I. Introduction

Evaluating the scalability or performance of target technologies themselves is, of course, important, but studying their behavior in abnormal situations is also important for understanding problems with surrounding services and the target technology itself.

Our system uses simulation and emulation in a large-scale testbed to reproduce abnormal situations including router misconfiguration, wireless network faults, and severe congestion. Various experiments have been conducted using fault injection; here we present two case studies and our approach.

II. Case Studies

We executed several experiments using fault injection. In this section, we introduce two examples; one involved emulating home networks, power management by a home energy management system (HEMS), and wide area networks, and the other is emulation of disaster situation and recovery technologies.

A. HEMS emulation

The topology of this experiment is shown in Figure 1. This experimental network has about 1100 nodes as experimental elements utilizing about 60 physical machines in StarBED[1]. The main elements of this experiment are emulated smart houses that have HEMS electronics and servers for power management. The HEMS electronics managers send power consumption information to the servers. The servers send a message to reduce the consumption when they detect a house is using too much electricity.

We introduced three types of faults in this environment. The first fault is negative effect of thunderstorm to the wireless access network. We changed the link characteristics on the network using QOMET[2] which emulates wireless link characteristics on wired networks. The concrete symptom is that the wireless links go down according to size and location of the thunderstorms.

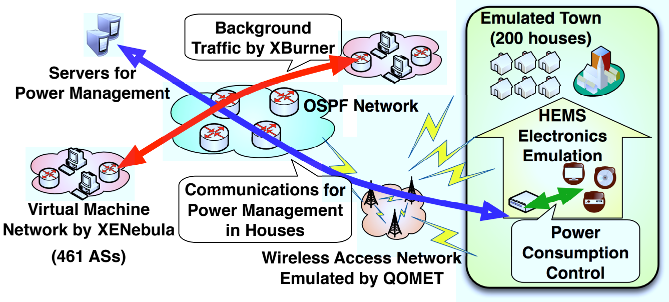

Fig. 1. Topology for HEMS Emulation

The second fault is misconfigurations of OSPF router and it involved the OSPF daemon down state.

The last fault involves burdening the OSPF network with a massive amount of background traffic using XBurner[3]. We run *apache* and *wget* on virtual machines using XENebula and these applications generate massive traffic in the OSPF network.

We actually built two environments: one with these faults and one without faults. The three types of faults caused communication failures between the HEMS servers and clients, and we observed the effects of these failures. The power consumption in the environment with faults is larger than for that without faults, because the management servers could not collect data from the houses or because the control messages from them to the HEMS clients were lost.

The details of this experiment are shown in [4].

B. Disaster emulation

Natural disasters also cause network trouble and there are many proposals for recovering from these situations. To provide environments for observing behaviors of network elements in these critical situations and evaluating recovery technologies, we executed an experiment in a disaster situation. The visualized environment is shown in figure 2.

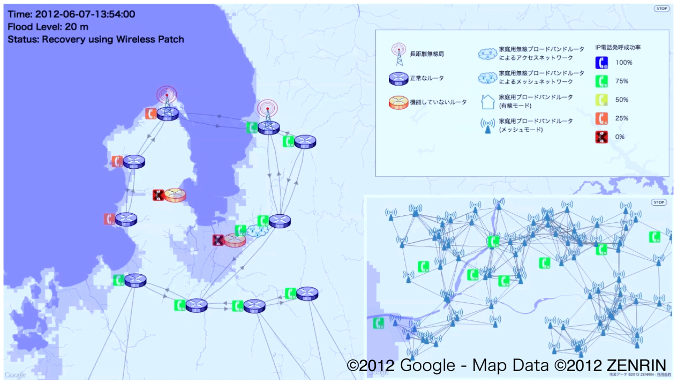
Time: 2012-06-07-13:54:00
Flood Level: 20 m
Status: Recovery using Wireless Patch
STOP
長距離無線局
正常なルータ
機能していないルータ
家庭用無線ブロードバンドルータによるアクセスネットワーク
家庭用無線ブロードバンドルータによるメッシュネットワーク
家庭用ブロードバンドルータ(有線モード)
家庭用ブロードバンドルータ(メッシュモード)
IP電話発呼成功率
100%
75%
50%
25%
0%
STOP
©2012 Google - Map Data ©2012 ZENRIN

Fig. 2. Disaster Visualization

There are OSPF routers emulating ISP inner network, SIP servers and clients, and homes which have network connection to a OSPF router. The emulated homes aren't emulated in all emulation area, only in a part of it. The OSPF router daemons, SIP servers and its clients are real implementations for the Internet and they work on StarBED physical nodes or virtual nodes.

We made a disaster scenario including an earthquake and a tsunami. When the earthquake hits the region, OSPF routers down because of power failure. In a while, they recovered but then two OSPF routers suffer tsunami and they are down again. In this moment, there are two problems. One is network separation, as three OSPF routers cannot connect the backbone area. The other is that the emulated homes lost their Internet connection because the OSPF router that they connect to is down. This is a disaster scenario in which we test how recovery technologies work.

We assume that future access routers in general homes have functions for building wireless mesh topology. So we run OLSR routing daemons in emulated homes to build the topology that reaches another OSPF router to have its external link. For the connection of the separated networks, we use a strong radio wave to connect between OSPF routers in each network. In this case, both recovery technologies are wireless-related, so we emulate the communication environment by utilizing QOMET.

The network status is shown as statistical values of SIP connection establishment every 3 minutes. The value goes down when OSPF routers are shutdown and goes up as the recovery techniques work.

We also implemented a visualization framework to collect emulation results and provide it to the visualizer.

III. Approach

In order to meet our goals, there were some technological hurdles; building a large-scale experimental setup, controlling it including fault processes and their consequences and so on. We use our tools for realizing a large-scale and realistic experimental environment on StarBED. SpringOS is supporting software for conducting experiments. The major functions of SpringOS are software installation, network topology construction, scenario driving, and related elements for facilitating the creation and execution of experiments. It facilitates us the creation of experiments. XENebula[5] enables users to run many virtual machines. The roles of XENebula are creating disk images and configurations for all virtual machines based on templates, allocating virtual machines to available physical nodes based on the computing resources of the physical machines, and running configured virtual machines on physical machines. QOMET can emulate characteristics of a wireless network on a wired network. It has behavior models of wireless nodes, and it configures the wired link characteristics using network emulators. It can be used to emulate fault situations by utilizing fault reproduction models instead of regular wireless models. XBurner introduces background traffic utilizing actual software and hardware for the Internet onto experimental environments. Figure 3 indicates the use of our technologies for fault reproduction.

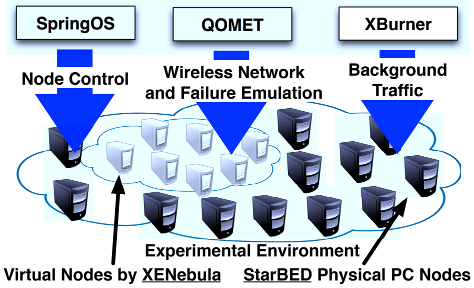

Fig. 3. Our Technologies for Building Experimental Environments for Failure Situations

IV. Conclusion and Future work

We built several experimental environments with faults utilizing our facilities and technologies: StarBED, SpringOS, XENebula, QOMET and XBurner. Example experiments indicate that we can emulate trouble situations with our existing technologies. But the reality of scenarios, models of trouble situations, topology and so on are not enough for precise emulations. Moreover, we execute many steps to make those experiments because some technologies we employed don't have collaboration mechanisms. Now we are designing and implementing a framework that will solve these problems. It will enable us to easily build and control disaster situations by utilizing simulation and emulation technologies.

References

[1] T. Miyachi, K. Chinen, and Y. Shinoda, "StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software," in *Valuetools 2006*, Oct. 2006.

[2] R. Beuran, J. Nakata, T. Okada, L. T. Nguyen, Y. Tan, and Y. Shinoda, "A Multi-purpose Wireless Network Emulator: QOMET," in *AINA 2008*, Mar. 2008.

[3] T. Miyachi, S. Miwa, and Y. Shinoda, "XBurner: A XENebula-based Native Traffic-generation Platform," in *Tridentcom2010, Poster*, May 2010.

[4] T. Miyachi, R. Beuran, S. Miwa, Y. Makino, S. Uda, Y. Tan, and Y. Shinoda, "Fault injection on a large-scale network testbed," in *AINTEC 2011*, Nov. 2011.

[5] S. Miwa, M. Suzuki, H. Hazeyama, S. Uda, T. Miyachi, Y. Kadobayashi, and Y. Shinoda, "Experiences in Emulating 10K AS Topology with Massive VM Multiplexing," in *VISA 2009*, Aug. 2009.