Emulation-based ICT System Resiliency Verification for Disaster Situations

Shingo Yasuda*, Kunio Akashi[‡], Toshiyuki Miyachi*[‡], Razvan Beuran[‡], Yoshiki Makino[‡], Tomoya Inoue[§],
Shinsuke Miwa*[‡] and Yoichi Shinoda[¶]
*Cyber Range Laboratory
National Institute of Information and Communications Technology, 2-12 Asahidai, Nomi, Ishikawa, Japan
Email: s-yasuda@nict.go.jp
[†]School of Information Science
Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, Japan
Email: k_akashi@jaist.ac.jp
[‡]Network Testbed Research and Development Laboratory
National Institute of Information and Communications Technology, 2-12 Asahidai, Nomi, Ishikawa, Japan
Email: {miyachi,razvan,ymakino,danna}@nict.go.jp
[§]Dependable Network Innovation Center
Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, Japan
Email: t-inoue@jaist.ac.jp
[¶]Research Center for Advanced Computing Infrastructure
Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, Japan
Email: shinoda@jaist.ac.jp

***Abstract*—Disaster situations require resilient ICT systems in order to provide as good as possible communication conditions in such catastrophic circumstances. The resiliency verification of ICT systems is however difficult, because reproducing large-scale disaster conditions in production networks is impossible without affecting their users.**

In this paper we present an emulation-based approach for evaluating the resiliency of ICT systems in disaster situations. This is achieved by reproducing disaster-like conditions in a emulated environment running on a large-scale testbed on which actual network protocols and applications are executed in real time.

The experimental results presented in the paper demonstrate how the effects of both the emulated disasters and those of the recovery technologies we subsequently deploy can be quantified objectively. This methodology can be used to improve the resilience of the systems under test.

***Keywords*-ICT systems, resiliency verification, network emulation, disaster situations.**

I. Introduction

In recent years, ICT systems have become an essential part of the social infrastructure. Hence, it is important that they function as well as possible at all times, including during and after the occurrence of natural disasters.

Improving the resilience of ICT systems is therefore a significant task of researchers and engineers in this area. This can be done first of all by creating technological specifications for such resilient systems. However, of utmost importance is the *verification* of the resilience of the ICT systems.

Nevertheless, natural disasters cannot be considered as "test environments" for ICT technologies, since this can have grave consequences in terms of human and material losses. Moreover, disaster conditions cannot be reproduced in production networks, as this would affect their current users. Therefore, a new methodology for resilience verification is required.

In this paper we present an emulation-based approach for ICT system resilience verification that targets specifically disaster situations. Our approach uses as infrastructure StarBED, a large-scale network testbed managed by the National Institute of Information and Communications Technology in Japan [1], and wireless network emulation capabilities provided by the QOMET set of tools that we developed [2].

On top of this infrastructure we have created a disaster emulation framework that reproduces the effects of natural disasters, such as earthquakes and tsunamis, in a virtual environment. These natural effects correspond to network effects that are emulated as well on our testbed. This approach makes it possible to assess the resilience of ICT systems in a repeatable and controllable manner.

The evaluation and case study presented in this paper demonstrate how typical disaster situations can be reproduced using our approach. The results also emphasize how the performance of ICT systems can be quantified in an objective manner, before, during and after disasters occur. We also show several examples of network-recovery technologies, and determine their effects on the overall network performance.

The paper is organized as follows. In Section II we briefly introduce the network emulation testbed that constitutes the infrastructure for our framework. Then, in Section III we provide more information about the disaster emulation methodology that we employ, illustrated through a preliminary evaluation. A detailed case study is then presented in Section IV, including the disaster emulation scenario and the applications used to evaluate network performance. Several experimental results drawn from this second case study are discussed in Section V.

978-1-4799-3211-5/13 $31.00 © 2013 IEEE
DOI 10.1109/SITIS.2013.143

IEEE computer society

The paper ends with a section of future work, conclusions, acknowledgments and references.

II. Network Emulation Testbed

In this section we briefly describe the network emulation testbed on which our framework is executed. Its main components are StarBED and QOMET.

A. StarBED

StarBED is a large-scale wired-network testbed at the Hokuriku StarBED Technology Center of the National Institute of Information and Communications Technology, located in Ishikawa, Japan. With over 1300 interconnected PCs, users can perform a wide range of network experiments on this testbed. A key characteristic of StarBED is that experiment traffic and control traffic are handled by different networks, so that there is no impact of the control traffic on the experiments.

A software suite called SpringOS is used for conducting experiments on StarBED. The main functions of SpringOS are: (i) resource management; (ii) power management of the nodes; (iii) software installation; (iv) network topology construction; (v) scenario execution.

B. QOMET

QOMET is a set of tools for wireless network emulation that we have been developing since 2006. QOMET is used to reproduce the characteristics of wireless networks in the wired network of StarBED. QOMET has behavior models of wireless nodes, and it configures the wired-link characteristics of the nodes so as to reproduce a user-defined scenario.

QOMET allows the definition of various complex scenarios, including experiments with various wireless technologies, with node mobility and in urban settings. The most important components of QOMET are the libraries called `deltaQ` and `wireconf`. The `deltaQ` library is in charge of computing the communication conditions between wireless nodes given an XML-based scenario. These communication conditions are recreated during the real-time experiment by the `wireconf` library. This library applies the corresponding network degradation, i.e., packet loss, delay and bandwidth limitation, to the experiment traffic. At operating system level this is effectively done by using dummynet/ipfw3 [3].

III. Disaster Emulation

A. Overview

Our target is to replicate disaster influences in an experimental replication of IT infrastructure, and to run resilient technology to fix the problems on an actual-node-based testbed. The environment should be have a large number of elements, and we want to run the same experimental scenario many times on it for comparing experimental results with different parameters. Thus, the environment should be built and the events should be executed automatically according to a user-defined configuration.

SpringOS can control physical configuration of experimental nodes and run commands on them. QOMET can emulate wireless link conditions on wired links. These tools emulate parts of the elements on our experimental environment for emulating critical ICT environments, and we call them *emulators*. Moreover, they have their own configurations and run individually. Therefore, in order to orchestrate these emulators, a mechanism that controls and triggers them with a single overall scenario — including all these sub-technologies' configurations — is required for building a realistic, flexible controllable and reproductive environment.

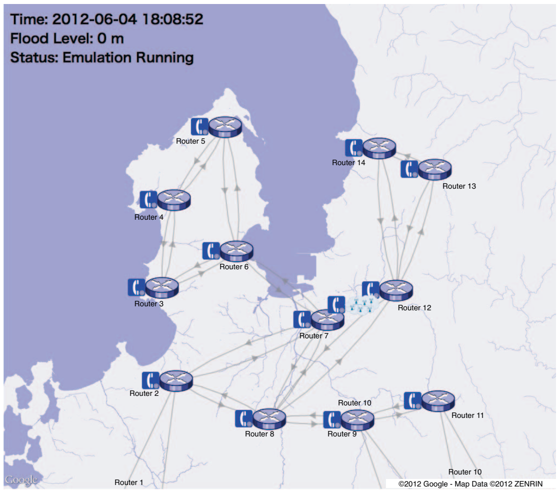


Figure 1. Stable network topology for the preliminary evaluation of our approach

B. Preliminary Evaluation

For building the whole environment, we first evaluated the concept of our framework by building the whole target with our existing tools. We run our tools such as SpringOS and QOMET manually to build an environment and to run the scenario of a critical situation.

In particular, our disaster scenario was that an earthquake and tsunami hit an area, and they destroy ICT infrastructure in the area. Then, resilient network-recovery technologies are run to fix the problems that occur because of the disaster.

There are OSPF routers emulating the ISP inner network, SIP servers and clients, and homes which have network connection to an OSPF router. The emulated homes correspond to only a part of the entire emulation area. 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. The stabilized topology is shown in Figure 1. 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 (blue icon indicates 76-100% success of SIP connections).

When the earthquake hits the region, OSPF routers go down because of power failure. In a while, they recover, but then two OSPF routers (Router 6 and 7) suffer because of tsunami and they go down again. This leads to two network problems. One is network separation, as three OSPF routers (Router 3, 4 and 5) cannot connect to the backbone area. The other is that the emulated homes around Router 7 and 12 lost their Internet connection because the OSPF router that they connect to is down. Next we tested how recovery technologies can be used to mitigate these problems.

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 a network topology that reaches another OSPF router to have its external link. For the connection of the separated networks, we use a strong radio signal to connect the Routers 5 and 14 in each network. Both recovery technologies are wireless-related, so we emulate the communication environment by utilizing QOMET.

Figure 2 shows the recovered status of this preliminary evaluation scenario. The figure indicates not only the wired topology but also the wireless one that is built by the actual OLSR protocol running on the emulated wireless network (see the bottom-right area of the figure).

IV. Case Study

The preliminary evaluation discussed so far was a very simple demonstration. It included only one application (SIP), only one ISP network, and no emulated mobility. Moreover, the WiFi mesh network was constructed using home router devices and it got enough continuous connectivity as the result. On the other hand, it needs a high density of devices to get enough connectivity on such as WiFi mesh network utilized by static devices.

In this section, we describe the experimental environment, scenario and recovery techniques for a more complex case study. More specifically, we propose a new experimental environment and scenario including three network recovery techniques, and one new application whose characteristics are assessed using disaster emulation.

A. Overview

We present the topography and network environment and the disaster scenario for this case study. Figure 3 shows the topography and stable network topology. There are 2 regional ISPs and there are 12 routers in total in the area. In addition, there are 5 routers for each ISP's backbone network router located outside the visible area. These ISPs have independent backbone networks in normal times. The ISP1 has backbone to bottom-left side. The ISP2 has backbone to top-right side. In our emulation these routers are software routers that utilize Quagga [4] running on virtual machines (KVM) [5].

There is a WiFi mesh network as a first network recovery technique for inter-router network recovery. It recovers the inter-router connectivity between router 5 and 6 if they are disconnected. We emulate the WiFi mesh network using OpenWRT with QOMET running as virtual machines too. The visualization for WiFi mesh network and other two recovery techniques that follow will be shown in the next section with experimental results.

The next recovery technique is a vehicular network at center of the topography. It is a DTN that utilizes mobile devices on vehicles which are moving disaster area (e.g., from home to evacuation center). We emulate the vehicles as WiFi network devices; here too we utilize OpenWRT [6] with QOMET running as virtual machines.

The third recovery technique is inter ISP recovery. We design the two local ISPs in disaster area. In this scenario we configure the peer connection between ISP1 and ISP2 for recovery the backbone network.

B. Applications

1) EMA: We developed an application that is specifically designed for post-disaster conditions. The application is called EMA (Emergency Messaging Application). A special application is particularly needed if connectivity is provided over DTNs, since such intermittent connectivity differs from the typical end-to-end network connections. EMA has a relatively simple design, and it has been written in the Ruby programming language, which was selected as a convenient solution for fast prototyping.

Screenshots of EMA's interface are shown in Figure 4. EMA was designed to send messages to various destinations, and to receive acknowledgments that the messages have been successfully delivered. Users receive visual feedback regarding each message status from the application, with a three-color code to differentiate between: (i) messages sent but not acknowledged yet; (ii) messages sent and acknowledged; (iii) messages sent but not acknowledged within a certain time interval (set to 240 s in our experiments). The right-hand side balloons in the EMA's interface show the sent messages. The left-hand side balloons show the received messages from other EMAs. The received message balloons are always yellow because there is no corresponding acknowledgment on the receiver side.

To connect to and utilize the vehicular network, EMA utilizes IBR-DTN [7] as bundle layer of DTN [8]. Specifically EMA uses two built-in IBR-DTN commands: "dtnsend" to send messages, and "dtnrecv" to receive messages as well as message acknowledgments. "dtnsend" has option to set the message's life time. In this case study, we specified this option as 120 seconds. EMA also logs the time of sending and receiving messages. Note that during our emulation experiments the message content was extracted from a database instead of being written and sent by real users.

2) SIP: We developed a SIP system that utilizes open source software for this study. This SIP System reports to the data logger the network condition which applications perceive by displaying the call success rate from SIP clients. We developed these clients by using the eXosip Library [9]. The SIP clients run on several virtual machines. Table I show the specifications of the SIP system.

We configured two physical nodes who are assigned one physical node per ISP. A total of 17 virtual machines run

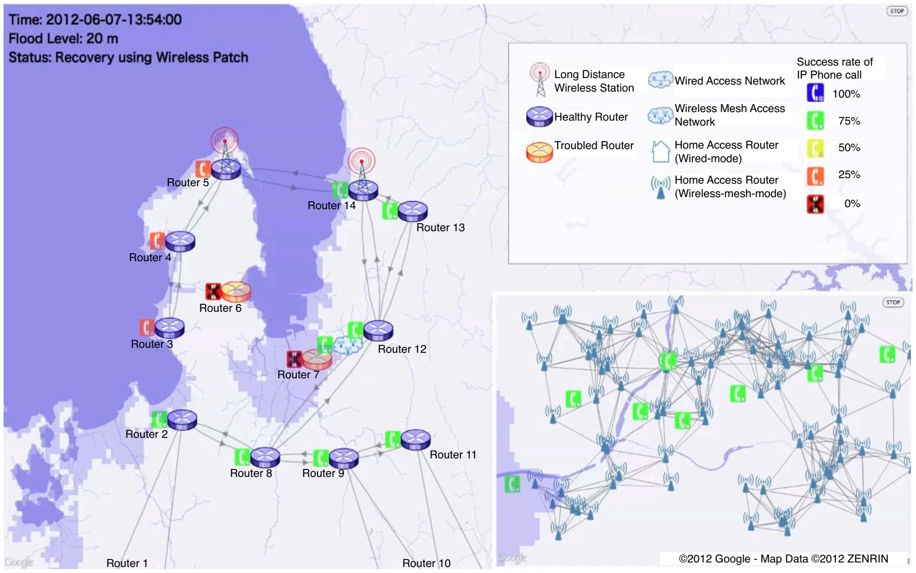

Time: 2012-06-07-13:54:00
Flood Level: 20 m
Status: Recovery using Wireless Patch
Long Distance Wireless Station
Healthy Router
Troubled Router
Wired Access Network
Wireless Mesh Access Network
Home Access Router (Wired-mode)
Home Access Router (Wireless-mesh-mode)
Success rate of IP Phone call
100%
75%
50%
25%
0%
Router 1
Router 2
Router 3
Router 4
Router 5
Router 6
Router 7
Router 8
Router 9
Router 10
Router 11
Router 12
Router 13
Router 14
STOP
©2012 Google - Map Data ©2012 ZENRIN

Figure 2. Recovered network topology in the preliminary evaluation of our approach

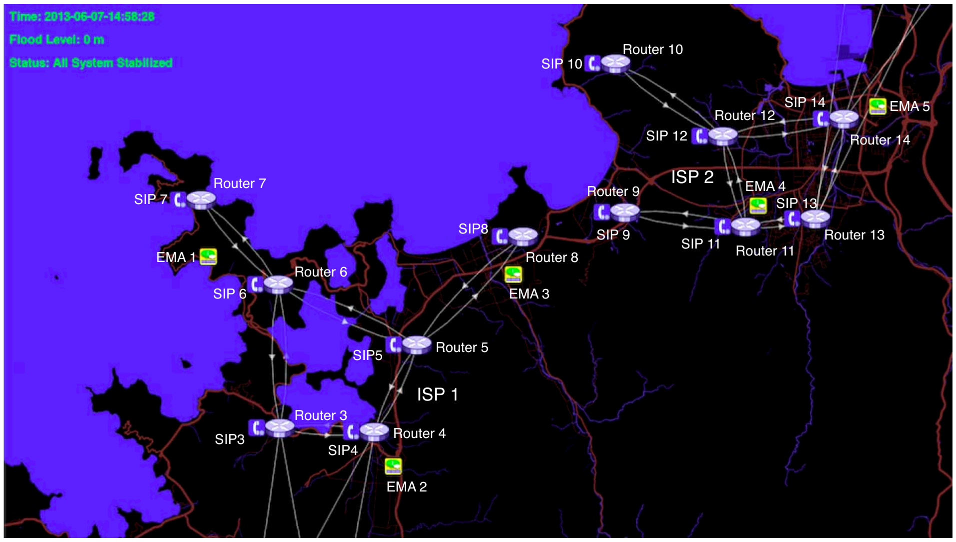


Figure 3. Topography and stable network environment

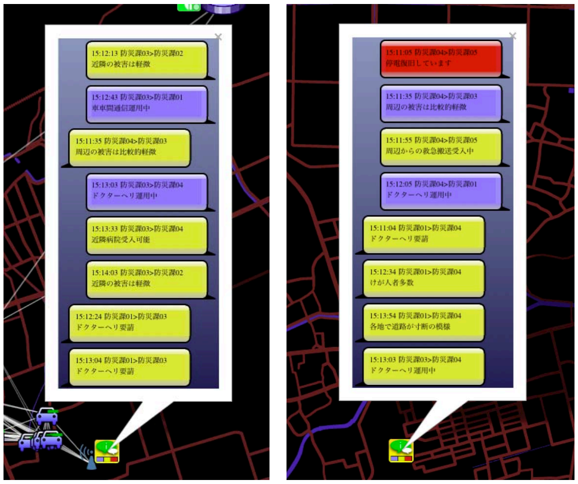
Table I
SIP SYSTEM SPECIFICATIONS

SIP server	OpenSIPS 1.8.0
SIP client	A custom application utilize eXosip2-3.6.0

on the nodes, equal to the number of routers. The virtual machines connect to each of the routers, and SIP clients are run on the virtual machines. Every client makes call requests to all clients except itself once every 30 seconds, and they make call requests around 32 times per minute. The request timeout is set to 5 seconds. Two SIP servers utilize openSIPS [10] and are deployed for both ISP as SIP registrar and SIP proxy.

C. Scenario

Table II shows the scenario for this case study. For this case study, we decided to use seven scenario events. We did preliminary experiments for each event to verify the convergence time for the effects of each event. Then we created the scenario timeline by putting those events in sequential order. Experimental scenario lasts for 1080 s in total. At first, the earthquake hits the area. Hence, all node instances lose power supply. At 150 s, the power supply for all nodes is recovered, but at 220 s, Router#3, #4 and #14 lose the power supply again because Tsunami



15:12:13 防災課03>防災課02
近隣の被害は軽微

15:12:43 防災課03>防災課01
車車間通信運用中

15:11:35 防災課04>防災課03
周辺の被害は比較的軽微

15:13:03 防災課03>防災課04
ドクターヘリ運用中

15:13:33 防災課03>防災課04
近隣病院受入可能

15:14:03 防災課03>防災課02
近隣の被害は軽微

15:12:24 防災課01>防災課03
ドクターヘリ要請

15:13:04 防災課01>防災課03
ドクターヘリ要請

15:11:05 防災課04>防災課05
停電復旧しています

15:11:35 防災課04>防災課03
周辺の被害は比較的軽微

15:11:55 防災課04>防災課05
周辺からの救急搬送受入中

15:12:05 防災課04>防災課01
ドクターヘリ運用中

15:11:04 防災課01>防災課04
ドクターヘリ要請

15:12:34 防災課01>防災課04
けが人者多数

15:13:54 防災課01>防災課04
各地で道路が寸断の模様

15:13:03 防災課03>防災課04
ドクターヘリ運用中

Figure 4. Screenshots of the user interface of the EMA

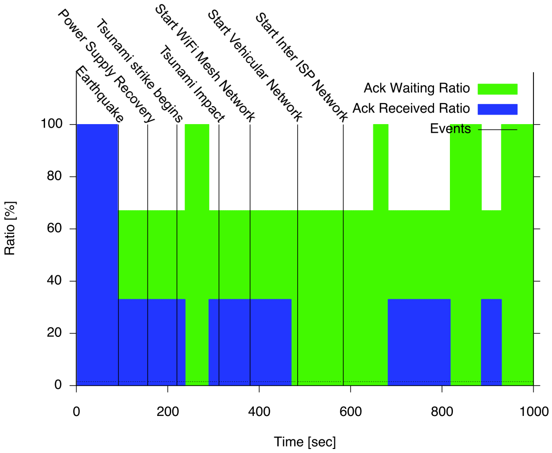


Figure 8. Ack waiting/receved ratio(a stacked bar graph) of EMA#3

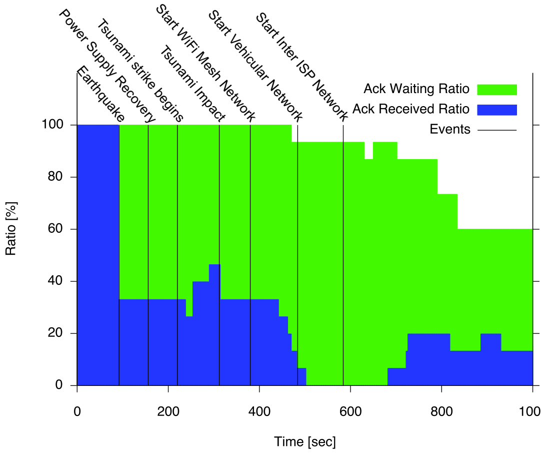


Figure 7. Ack waiting/receved ratio(a stacked bar graph) of all EMA

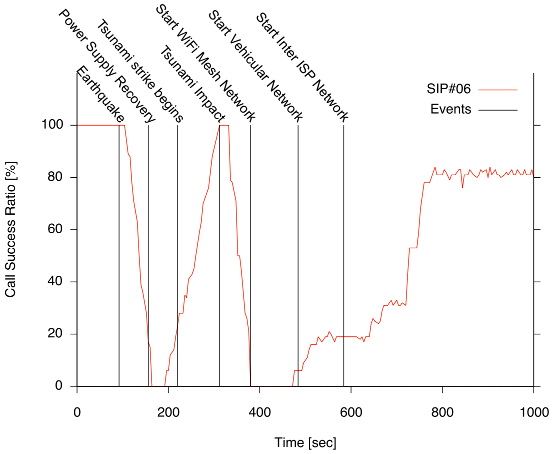


Figure 9. Call success ratio of SIP#06

strike the sea side area. Furthermore, the physical link between router#5 and router#6 is down too. For these reasons, the network of each ISP is broken. After that, the three recovery techniques start one by one.

V. Experimental Results

Figure 5 shows the screenshot after tsunami impact. Routers #3, #4 and #14 are down by tsunami impact. Furthermore, the physical link between router#5 and #6 is down by tsunami impact too.

Figure 6 shows the screenshot at the end of emulation. All network recovery techniques are started. The right-bottom side box is a zoom view of the vehicular network which is shown as a rectangle in center of the area.

A. EMA Results

Figure 7 shows the average of ack waiting/received ratio for all EMAs. All routers abruptly shutdown and the ack received ratio decrease after the blackout event caused by earthquake occurs at 90 s. The actual time when lost messages occurred is delayed. This is due to EMA's configuration of interval to decide the lost of message (240 s). After start the vehicular network, we can show the recovery of the message transfer. Its result enables us to get connectivity by three recovery techniques.

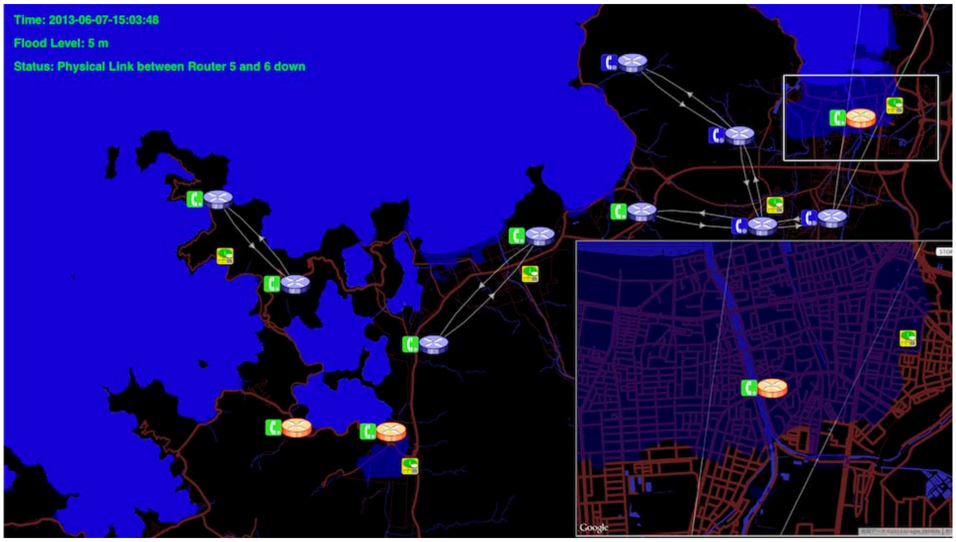
Figure 8 shows the average of ack waiting/received ratio for all EMA#3. EMA#3 connectivity recovers due to the vehicular network after tsunami impact which restores communication between EMA#3 and router#3. Hence there are the ack received message after start the vehicular network in the graph.

B. SIP Results

Figure 9 shows that all routers abruptly shutdown and the call success rate decreases after the earthquake occurred at 90 s. There is some delay between the blackout event and shutdown of routers because some time passes until the requests timeout. This explains the delay between the blackout event and the decrease in the call rates. After power supply recovery, there are also some delays till

Table II
SCENARIO TIME TABLE

Time[sec]	Event Name	Event Detail
0	Experiment Start	All System Stabilized
90	Earthquake	All Router Lost the Power Supply
150	Power Supply Recovery	All Router Power Supply Recovery
220	Tsunami Strike Begins	Tsunami Strike Begins from Sea Side
310	Tunami Impact	Physical Link Between Router#5 and#6 Down Router 3, 4 and 14 Down
380	Start WiFi Mesh Network	Start WiFi Mesh Network to Recover the Link Between Router#5 and #6
480	Start Vehicular Network	Start Vehicular Network to Recover the connectivity Between EMA#3 to Router#8
580	Start Inter-ISP Connection	Start Inter-ISP Connection Between Router#8 and Router#9
1080	Experiment Finish	Finish the Scenario and Stop All Nodes


Time: 2013-06-07-15:03:48

Flood Level: 5 m

Status: Physical Link between Router 5 and 6 down

Figure 5. Screenshots of the topology after Tsunami impact

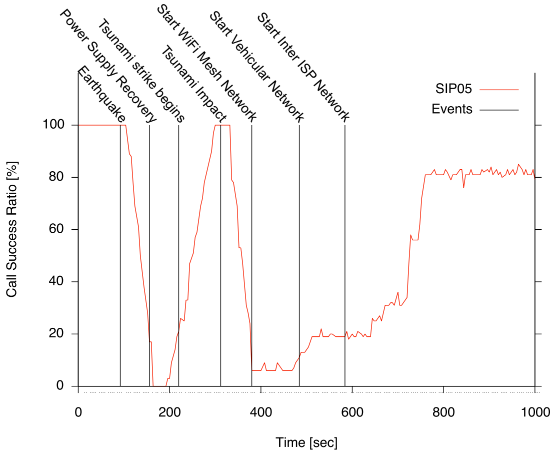
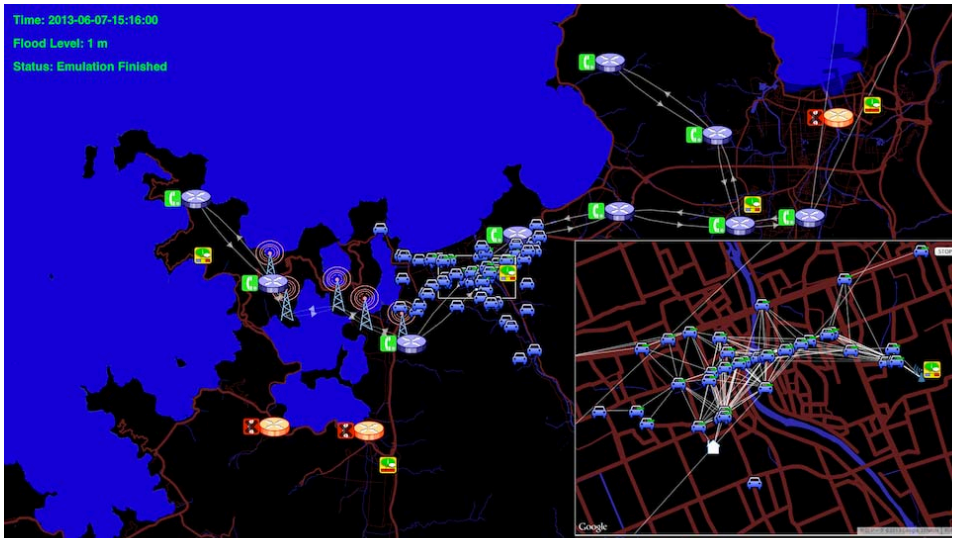


Figure 10. Call success ratio of SIP#05

the rate of requests recovers. This is because of routing information recovery time and potentially also due to issues of SIP server or SIP client.

Figure 9 also shows that it takes time until the call rates start decreasing once the link goes down between router#5 and router#6 and router shutdown (which occurs at 310 s). Moreover, the figure indicates that SIP clients at router 6 and router 7 lost the route to SIP server and call success rate becomes 0% after link down between router 6 and router 5.

On the other hand, SIP clients at router 5 and router 8 can send messages to SIP server and call requests between clients at router 5 or router 8 succeed. Therefore the call success rate become about 6% in Figure 10. The connectivity recovery between router 5 and router 6 by a WiFi multi-hop network begins at 380 seconds, and it takes almost one minutes until all links are established and provide complete service. Therefore the recovery of call success rate takes about 1 minute and 30 seconds from the starting time of mesh network construction. After an Inter-AS link between router 8 and router 9 is established, all SIP clients with connection with working routers recover and the success rate increase to around 80%. The delay of the effect of events related to the call success rate depends on the routing table reconstruction time, on the mesh network construction time, and on the behavior of SIP client and SIP server. In order to make the detailed cause of the result of this simulation clear, investigating the detailed trace of other simulators is required.



Time: 2013-06-07-15:16:00

Flood Level: 1 m

Status: Emulation Finished

Figure 6. Screenshots of the topology at emulation finished

VI. Future Work

Through a preliminary evaluation and a case study we validated the feasibility of our approach which applies existing tools to replicate critical situations and evaluate resilient technologies.

The most important issue we currently face is that our tools run individually. There is no mechanism to synchronize the emulators that control experimental resources. In these experiments, we only used shell scripts and simple communication based on JSON[11] to run overall scenarios. This is not user friendly and it makes conducting experiments difficult.

Experimenters typically would execute the same scenario many times to know which kind of network parameters make the infrastructure stronger and the resilient technologies more effective. Instead of shell scripts and simple communication, we need a more detailed framework to control the emulators and to use their functionality more effectively.

In current experiments, we just assumed the effect of natural disaster to experiment actors in the virtual replica of the ICT environment. Ideally, given a target area for the emulation, a *scenario sequencer* or *status calculator* tool should generate experiment actors' statuses at each timing according to their location, their physical and logical nature, human behavior and so forth. Then an *emulator trigger* should receive the results of this computation, and it should drive the emulators through a unified protocol. The emulators would then manage the experimental resources under their control to execute the scenario. Of course the result of each elements' action have impact on the next actions in the scenario, so the current status of experimental resources should be provided as a seed to scenario calculator.

VII. Conclusion

In this paper we have presented a disaster emulation framework that makes it possible to objectively determine the effects of disaster such as earthquakes and tsunamis on ICT systems. The framework also makes it possible to study how network performance is improved after deploying recovery technologies. Both these aspects can be used to improve the resiliency of ICT systems.

The experimental results presented demonstrate the feasibility of the approach, and make it possible to study the cause-effect relationship between various network events and the corresponding network performance.

Based on our experience we have now started to design a global architecture that will drive our various experiment tools and emulators in an unified manner that follows in a realistic manner a given disaster scenario. As discussed in Section VI, various modules are needed to coordinate the activities of the experiment actors, and to compute the conditions for each step of the scenario.

Acknowledgment

The authors would like to thank to Ph.D. Yasuo Tan, Yuki Kaji, Kouji Iwahashi for their contributions. The authors would like to thank to him for his significant contribution.

References

[1] T. Miyachi, K. Chinen, and Y. Shinoda, "StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software," in *Proceedings of the Intl. Conf. on Performance Evaluation Methodologies and Tools (Valuetools 2006)*, 2006.

[2] R. Beuran, J. Nakata, T. Okada, L. T. Nguyen, Y. Tan, and Y. Shinoda, "A Multi-purpose Wireless Network Emulator: QOMET," in *Proceedings of the 22nd IEEE International*

Conference on Advanced Information Networking and Applications (AINA 2008) Workshops, FINA 2008 symposium, 2008, pp. 223–228.

[3] M. Carbone and L. Rizzo, “Dummynet Revisited,” *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 12–20, 2010.

[4] Quagga, “Quagga routing suite website,” http://www.nongnu.org/quagga/.

[5] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, “kvm: the linux virtual machine monitor,” in *Proceedings of the Linux Symposium*, vol. 1, Ottawa, Ontario, Canada, Jun. 2007, pp. 225–230. [Online]. Available: http://linux-security.cn/ebooks/ols2007/OLS2007-Proceedings-V1.pdf

[6] OpenWRT, “Openwrt wireless freedom website,” https://www.openwrt.org/.

[7] S. Schildt, J. Morgenroth, W.-B. Pottner, and L. Wolf, “Ibr-dtn: A lightweight, modular and highly portable bundle protocol implementation,” *Electronic Communications of the EASST*, vol. 37, pp. 1–11, January 2011.

[8] K. Scott and S. Burleigh, “Bundle Protocol Specification, RFC5050,” November 2007.

[9] eXosip, “The extended osip library website,” http://savannah.nongnu.org/projects/exosip/.

[10] openSIPS, “Opensips website,” http://http://www.opensips.org/.

[11] D. Crockford, “The application/json Media Type for JavaScript Object Notation (JSON), RFC4627,” July 2006.