Using Emulation to Validate Post-disaster Network Recovery Solutions

Razvan Beuran
National Institute of
Information and
Communications Technology
Japan
razvan@nict.go.jp

Shingo Yasuda
National Institute of
Information and
Communications Technology
Japan
s-yasuda@nict.go.jp

Tomoya Inoue
Japan Advanced Institute of
Science and Technology
Japan
t-inoue@jaist.ac.jp

Shinsuke Miwa
National Institute of
Information and
Communications Technology
Japan
danna@nict.go.jp

Yoichi Shinoda
Japan Advanced Institute of
Science and Technology
Japan
shinoda@jaist.ac.jp

ABSTRACT

Given the high importance of communication for disaster management and recovery operations, it is essential to develop solutions for restoring network connectivity after natural catastrophes. We have used a disaster emulation framework built on top of the large-scale network testbed StarBED in order to virtually recreate a series of disaster conditions. Thus we could evaluate the potential of using Delay / Disruption Tolerant Network (DTN) protocols and inter-vehicle communication for post-disaster network recovery.

The results of our network emulation experiments prove that DTN in itself is a feasible alternative for network recovery. The objective assessment done using our emulation testbed on how communication performance depends on the mobility pattern of the vehicles and on their density can be used to make various practical decisions, for instance on how to manage the mobility of some of the vehicles in order to optimize network performance.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques

General Terms

Experimentation, Performance, Verification

Keywords

Network emulation, disaster emulation, disaster response, network recovery, vehicular networks, DTN

1. INTRODUCTION

Many of the countries in the Asia-Pacific Region are prone to natural disasters, such as earthquakes and tsunamis, due to the existence of the circum-Pacific seismic belt (also known as the "Ring of Fire"). Experience with previous disasters, such as the 2011 Tohoku earthquake and tsunami in Japan, has shown that communication networks greatly influence the response to the disaster and the recovery operations, including regarding the nuclear power station accident that was caused by this catastrophe [6].

Given that the existing communication infrastructure is often damaged in the affected areas, both Internet connectivity and cellular phone usage become difficult or impossible in post-disaster conditions. This has lead to an increased activity for developing network recovery solutions that often rely on the deployment of wireless networks (e.g., ad hoc/mesh networks, satellite communication), both in static and in mobile scenarios, since it is generally faster to deploy such technologies than to repair the damage to the existing network infrastructure.

One of the first efforts in this area was the DUMBO (Digital Ubiquitous Mobile Broadband OLSR) project [2] that was created as a response to the 2004 Indian Ocean earthquake and tsunami. This project has built an OLSR-based MANET to connect a number of sites that participated in a "simulated" rescue operation (i.e., not following a real disaster) and used it for various field experiments.

While we agree that field trials are very important in the context of determining viable solutions for network recovery, emulation techniques are useful for validating these solutions in a wide range of conditions, while running real applications and protocols. A network emulation testbed such as ours, which includes disaster emulation capabilities, makes it easier to assess the potential of each solution in a much more cost-effective and controllable manner. The field experiments can then focus on aspects which are difficult or time-consuming to emulate accurately, such as terrain or weather conditions [1].

Several static solutions for post-disaster communication recovery exist. Satellite dishes can be used to provide In-

ternet connectivity locally, but there can be various issues when establishing communication in this manner [6]. Mesh networks can be deployed in order to provide connectivity at several fixed locations. Mesh network deployments face challenges in positioning the wireless routers, since good connectivity can only be guaranteed if terrain knowledge is available. Other issues may include the operating life of the wireless routers in case they are battery operated.

One possible mobile solution for post-disaster network recovery is the use of inter-vehicle communication. This has several advantages compared to static deployments: (i) the car battery can function as power supply, thus not requiring additional power sources; (ii) car mobility leads to coverage of a potential larger area compared to static deployments. The latter is simultaneously one of the main challenges for inter-vehicle communication, since mobility leads to intermittent network connectivity and potentially low network performance. Other practical constraints may indirectly affect the performance of such networks, such as the lack of gasoline, which may restrict the number of moving vehicles and their mobility.

In this paper we investigate the potential of using inter-vehicle communication for post-disaster network recovery. To deal with the intermittent connectivity of this communication paradigm, we propose to deploy Delay/Disruption Tolerant Network (DTN) protocols, designed to cope with varying network conditions. Our main contributions are:

- We present a possible solution for network recovery that uses a DTN protocol running on top of a vehicular network, and that includes a messaging application for emergency communication;

- We evaluate the proposed solution through emulation on a testbed that includes both network and disaster emulation features in order to determine objectively the constraints involved, and what are the expected performance characteristics of such a solution.

The remainder of the paper is organized as follows. In Section 2 we summarize the most important aspects regarding our network emulation testbed. Then, in Section 3 we describe the main features of the disaster emulation framework built on top of the testbed. Section 4 introduces our post-disaster recovery solution and its components: vehicle communication, DTN, and emergency messaging. This is followed by an evaluation of the proposed solution through emulation (Section 5) that assesses its performance under various scenarios. The paper ends with conclusions, acknowledgments, and references.

2. NETWORK EMULATION TESTBED

StarBED is a network testbed located in Ishikawa prefecture, Japan, at the Hokuriku StarBED Technology Center of the National Institute of Information and Communications Technology [5]. StarBED makes available for experiments more than 1300 interconnected PCs — the physical infrastructure of our emulation platform.

QOMET (acronym for *Quality Observation and Mobility Experiment Tools*) is a set of tools for the wireless network emulation of various complex scenarios, including with node mobility. QOMET provides the necessary mechanisms for performing experiments in a distributed manner by reproducing on StarBED the communication conditions between the emulated wireless nodes.
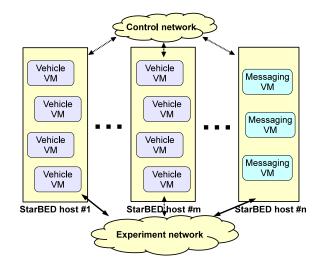

Figure 1: Overall network emulation architecture.

2.1 Emulation architecture

The architecture used in the emulation experiments presented in this paper is as follows. The vehicles are emulated by using virtual machines (VMs) running the OpenWrt Linux distribution for embedded devices [8]. The software installed on each of these VMs includes the DTN protocol implementation and the QOMET modules necessary for wireless network emulation.

QOMET is used to reproduce the wireless communication conditions between vehicles as they move in the virtual world in which the scenario takes place. For these experiments, the mobility itself is generated by using the ONE simulator, more precisely the Shortest Path Map-Based Movement (SPMBM) model included in ONE [3]. The input for the mobility model is created by using real road data for the area of interest obtained from the OpenStreetMap website [7] and converted to a ONE-compatible format by using the `osm2wkt` program [4]. The mobility trace generated by ONE is used by QOMET when computing the communication conditions between the emulated mobile nodes.

The emergency messaging application software is executed on different VMs running the Ubuntu 12.04 OS. The DTN protocol is also running on these VMs in order to interface with the vehicular network. These aspects will be covered in more detail in Section 4.

The overall network emulation architecture is shown in Figure 1. The VMs that emulate vehicles are denoted by "Vehicle VM". They are run on StarBED hosts; in particular, for 100 vehicle experiments we have used 5 physical StarBED hosts (with dual Intel 6-core Xeon 3 GHz CPUs and 48 GB RAM). The VMs running the messaging application are denoted by "Messaging VM". A total of 6 such VMs were run on a single StarBED host. The StarBED control network is used to manage the experiment, and the experiment network carries the experiment traffic.

3. DISASTER EMULATION FRAMEWORK

In order to be able to do experiments in circumstances that emulate disaster scenarios, we created a disaster emulation framework that interacts with the network emulation testbed to recreate conditions that reflect the occurrence and
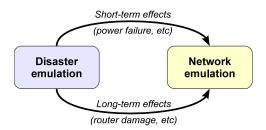
Figure 2: Interaction between the disaster emulation framework and the network emulation testbed.
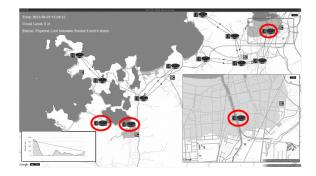

Figure 3: Earthquake and tsunami scenario recreated using our disaster emulation framework.

consequences of natural disasters. Conceptually, the disaster emulation framework can be thought of as a module that has information about various disaster types and their effects; this module uses the network emulation mechanisms available on the testbed in order to recreate these effects through emulation according to a predefined sequence of events (see Figure 2).

For earthquakes, the most significant effect is considered to be temporary power failure of the network equipment that uses mains electricity in the affected areas (e.g., backbone routers). Another possible effect is long-term damage to the inter-connecting infrastructure (optical fibers, etc.).

Regarding tsunamis, the main effect is considered to be the flooding of a certain area, which causes long-term damage to the network equipment in that area. The region affected by flooding is determined by using information about how the height of the tsunami changes over time, and terrain data for the area of interest. As tsunami height varies in time, this leads to a gradual increase of the water level, followed by its decrease after a certain interval.

A sample scenario as displayed by our visualization interface is shown in Figure 3. The scenario replicates the effects of an earthquake and the ensuing tsunami. The screenshot is taken soon after the tsunami has reached the maximum level. As a consequence, three of the backbone routers are flooded and become unusable for the remainder of the experiment (the affected routers, located in the lower-left and the upper-right parts of the figure, are circled for clarity). A detail view in the lower-right corner of the figure includes notable information at various moments in the scenario, in this case the flooded router in the upper-right corner. This disaster scenario was presented more thoroughly in [10].

We envisage that various further improvements can be made in order to increase the realism of the disaster emulation framework. For instance, the affected area and the strength of the effects of an earthquake could be emulated better if epicenter and magnitude information are used together with topography information. Both for earthquakes and tsunamis, effects other than those directly related to the network infrastructure could be considered, such as damage to the transport infrastructure (e.g., fallen bridges, flooded roads). This kind of damage has an indirect effect on the network, since it influences the areas in which network recovery equipment can be placed, the mobility of people and vehicles, etc. Such improvements represent the main focus of our future work.

4. NETWORK RECOVERY USING DTN

The proposed solution for post-disaster network recovery using DTN and inter-vehicle communication follows.

4.1 Vehicular network

The most important component of our solution is the vehicular network. We have already discussed some of the advantages of this solution in the introduction; here we provide several additional details. We believe that an important requirement is to make available such connectivity to the vehicle network so that it can be used by typical users carrying regular smartphones or laptops, for instance by using the IEEE 802.11 standard. For the backhaul of the vehicle network, special technologies designed for inter-vehicle communication could be used, such as 802.11p or even white-space networking, but in the current experiment we assume the use of IEEE 802.11b. In particular, we used a transmit power of 18 dBm, and for propagation the log-distance path-loss model with attenuation coefficient 3.2 (outdoor environments).

As mentioned in Section 2.1, the mobility in our experiments is generated using the SPMBM model included in the ONE simulator. In particular, we specify certain points of interest which are selected by the moving vehicles as their destination with user-configured probabilities. This allows us to influence the mobility of the vehicles (instead of using less-realistic completely random motion), and to evaluate its effect on network performance. More details about these settings will be given in Section 5.1.

4.2 DTN protocol: IBR-DTN

In order to cope with the intermittent connectivity of the vehicular network, we propose to use the DTN protocol. In our experiments we have used the IBR-DTN implementation from the Technical University Braunschweig, Germany, which is a light-weight implementation targeting embedded systems [9]. IBR-DTN was installed both on the OpenWrt vehicular VMs used in our experiments (v0.6.5), and on the Ubuntu messaging VMs (v0.8.0).

We configured IBR-DTN to use the "flooding" protocol, so as to maximize the probability to deliver messages. Thus, each node determines dynamically the potential peers during the experiment (as the vehicles move), and forwards the messages in its queue to these nodes. For experimental purposes we configured the lifetime of the DTN data bundles to 2 minutes, but in practice they could be much longer. As information that is received too late might not be relevant anymore, this parameter is application dependent.

Although in real life DTN nodes are typically not synchronized with each other, in order to make possible precise
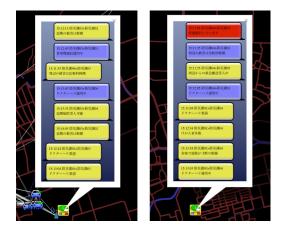
Figure 4: Screenshots of the user interface of the emergency messaging application (EMA).

Figure 5: Screenshot of the DTN and inter-vehicular communication experiment scenario.

end-to-end delay measurements we used the NTP protocol to synchronize the DTN node clocks. The protocol was running on the control interface of the host, hence no interference with the experimental results occurred. In practical vehicular network scenarios, using the GPS signal received by on-board navigation systems is a possible solution for time synchronization if necessary. In general, the required synchronization accuracy depends on the magnitude of the time parameters involved (lifetime, etc.).

4.3 Emergency messaging application: EMA

We believe that restoring some form of network connectivity is not enough in post-disaster circumstances, since in many cases the applications used in normal conditions may not function properly. This is potentially due to various degrees of network damage: for instance, mail servers may not be reachable anymore, etc. Therefore we also developed an application that is specifically designed for post-disaster conditions. The application is called EMA (Emergency Messaging Application). Such a special application is particularly needed if network access 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. Although currently EMA requires a regular computer for execution, in the future we envisage that smartphone versions of EMA could be made available for use in practical circumstances.

EMA instances send/receive both messages and acknowledgments that the messages have been successfully delivered. Users get visual feedback regarding each message status with a three-color code: (i) messages sent but not acknowledged yet; (ii) messages sent and acknowledged; (iii) messages sent but not acknowledged within a certain time interval. Screenshots of EMA's interface are shown in Figure 4.

To connect to and utilize the vehicular network, EMA uses two built-in IBR-DTN commands: "dtnsend" to send messages/acknowledgments, and "dtnrecv" to receive them. EMA also logs the time of sending and receiving messages. Note that during our emulation experiments the message content was extracted from a pre-configured message database instead of being written and sent by real users.
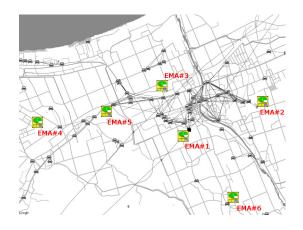
5. EXPERIMENTAL RESULTS

The experiments aimed at validating the feasibility of our post-disaster network recovery solution were conducted by running actual implementations of the corresponding software systems over the emulated vehicular network.

5.1 Overview

The overall scenario considers a certain area on the coast of Japan that is affected by an earthquake, followed by a tsunami. These disasters cause damage to the network infrastructure, and various solutions can be used for partial network recovery, including collaboration between ISPs to restore Internet connectivity to a given disconnected area, or a statically-deployed mesh network to restore connectivity between backbone routers [10].

The experimental results shown in this paper focus on a third solution, which is the use of DTN and inter-vehicle communication to allow information exchanges between several sites of interest in an area of about 6×4 km^2 (see Figure 5). We assume that the road infrastructure in this area was not significantly affected by the disaster, hence it allows traveling by car around the sites of interest.

In our scenario, the main role of inter-vehicle communication is to allow sending and receiving information between the secondary school used as gathering point of disaster victims (EMA#2) and several other sites, such as the town hall (EMA#1), a railway station (EMA#3), and another school (EMA#4). For comparison purposes, we also considered two mobile messaging nodes, EMA#5 and EMA#6. The assumption is that users will exchange messages about their well-being and other useful information with other people that can connect to the vehicular network.

We focus on three types of mobility patterns, with different mobility model parameters regarding vehicle destination selection. The probabilities for each car to select as destination the town hall, the secondary school or another random destination are given in Table 1 for the three mobility patterns that we denoted by A, B and C, respectively. Vehicle speed is randomly selected between 10 and 50 km/h.

5.2 Performance evaluation

Our assessment focused on evaluating the message communication conditions between the secondary school site (EMA#2) and the other sites with messaging capabilities.

Table 1: Probability for car motion destinations for the three mobility patterns

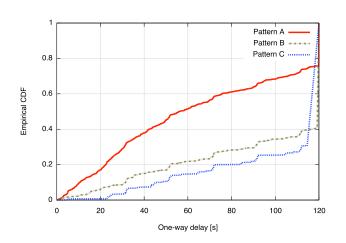| Mobility pattern	Town hall (EMA#1)	Secondary school (EMA#2)	Other destinations
A	0.5	0.4	0.1
B	0.4	0.3	0.3
C	0.3	0.2	0.5

Figure 6: CDF of one-way delay for experiments with 100 vehicles and the three mobility patterns.

For this purpose we configured the EMA#2 application to send packets at regular intervals to the other EMA instances and calculated the one-way delay for each message; we also computed the average loss rate per node. The results that will be shown below are obtained with a message interval of 20 s and a message size of about 300 bytes. As the lifetime of the DTN bundles was configured to 2 minutes, the DTN nodes will drop messages once their "age" exceeds this interval. Experiments had a duration of 10 minutes, and we did 10 runs for each of them.

A first class of experiments investigated the influence of *mobility patterns* on performance. Therefore we used 100 vehicles with the mobility traces being generated according to each of the three mobility patterns A, B, and C mentioned before. The empirical CDF of one-way delay measurements for each mobility pattern is shown in Figure 6. It is obvious that results are best in the case in which the probability of vehicles to reach the sender, EMA#2, are highest (mobility pattern A), and become worse as the vehicles chose more random destinations (mobility patterns B and C).

For a better insight we show in Figure 7 the boxplot of the one-way delay per node for each mobility pattern and message destination. Average loss rates are also displayed below the node labels on the x axis. The performance for messages towards EMA#1 and EMA#3 is somewhat similar even though no vehicle heads directly for EMA#3; this is because EMA#3 is located not far from EMA#1 and both are serviced by partially common roads. On the other hand, EMA#4, which is a remote location, has high loss rates and large delays. EMA#5 and EMA#6 are mobile nodes, hence their performance is directly influenced by their individual trajectories, with better communication conditions for the first of them.
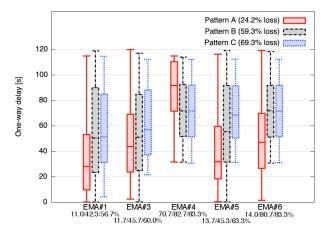

Figure 7: Per node delay for experiments with 100 vehicles and the three mobility patterns.
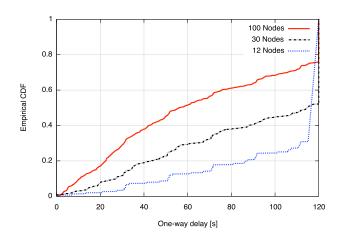

Figure 8: CDF of one-way delay for experiments with mobility pattern A and different vehicle counts.

A second class of experiments looked at the influence of *vehicle count* on performance. In this case we used one type of mobility pattern, namely pattern A, and changed the total number of vehicles that form the network from 100 to 30 and then to 12 vehicles, respectively. Again we plot first the empirical CDF for each experiment (Figure 8). The decrease of the communication performance with vehicle count is obvious and appears to be linear.

The boxplot of the one-way delay per node for each experiment is shown in Figure 9. For EMA#1, which is directly "serviced" by the vehicles, the performance change is very clear, with both delay and loss increasing as vehicle count decreases. Similar changes are observed for the mobile nodes EMA#5 and EMA#6. EMA#3 is not directly serviced, and its delay performance doesn't vary so much for the 30 and 12 vehicle cases; however the loss increases from 50% to 83%, hence in this case too the influence is strong. The remote node EMA#4 doesn't experience significant changes in performance because it is low to start with (70% loss even with 100 vehicles).

Finally, in Table 2 we present both the average delay and the average loss rate for all the possible combinations of
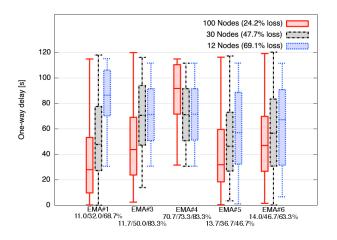
Figure 9: Per node delay for experiments with mobility pattern A and different vehicle counts.

Table 2: Average delay and loss rate per experiment for all mobility patterns and vehicle counts

Pattern	100 cars	30 cars	12 cars
A	47.2 s/24.2%	58.0 s/47.8%	67.5 s/69.1%
B	58.8 s/59.3%	74.0 s/65.3%	70.9 s/73.7%
C	64.9 s/69.3%	74.8 s/76.2%	71.2 s/77.1%

mobility patterns and vehicle counts that we discussed so far. As expected, the maximum performance (i.e., lowest average delay and loss rate) is achieved for the case with the tightest mobility control and highest vehicle count: mobility pattern A with 100 vehicles (loss rate of 24%, and an average delay of 47 s).

In practical circumstances a trade-off between cost and performance often has to be made. Our experiments indicate that if the movement of the vehicles can be controlled/influenced, then vehicle count can be decreased; the measurement results can help making informed and objective decisions in such circumstances. For instance, mobility pattern A with 30 vehicles still has loss rate under 50% and an average delay of 58 s, compared to the less constrained mobility pattern B, for which even when using 100 vehicles the loss rate is around 60% and the average delay is 59 s.

6. CONCLUSIONS

We have presented a network emulation testbed that has disaster emulation capabilities. We have used this testbed to objectively investigate the performance characteristics of a post-disaster network recovery solution that involves the DTN protocol running on a vehicular network. Details about StarBED and the tools available for it can be found on the following website: `http://www.starbed.org`.

Our emulation experiments using a custom-designed messaging application for emergency communications (EMA) have shown that the influence of the mobility pattern is strong, and that a trade-off can be made with vehicle count if necessary, at the expense of an increased delay and packet loss. These experiments demonstrate how our testbed can be employed for a thorough evaluation of practical solutions related to network recovery in disaster situations (but not limited to that).

We are planning to extend the usability and realism of the disaster emulation framework with support for better emulation of the damage caused by disasters based on physical factors such as earthquake intensity, tsunami speed (including the damage to road infrastructure, etc.). We are also currently working on a generic model for the interface between the disaster emulation component and the network emulation testbed that will allow more flexibility when taking into account the effects of disasters during emulation.

Acknowledgments

7. REFERENCES

[1] R. Beuran. *Introduction to Network Emulation*. Pan Stanford Publishing, 2012.

[2] K. Kanchanasut, A. Tunpan, M. A. Awal, D. K. Das, T. Wongsaardsakul, and Y. Tsuchimoto. DUMBONET: A Multimedia Communication System for Collaborative Emergency Response Operations in Disaster-affected Areas. *International Journal of Emergency Management*, 4(4):670–681, 2007.

[3] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools '09)*, 2009.

[4] C. P. Mayer. osm2wkt - OpenStreetMap to WKT Conversion, 2010. `http://www.tm.kit.edu/~mayer/osm2wkt`.

[5] 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.

[6] Nuclear Emergency Response Headquarters, Government of Japan. Report of Japanese Government to the IAEA Ministerial Conference on Nuclear Safety – The Accident at TEPCO's Fukushima Nuclear Power Stations, 2011. `http://www.kantei.go.jp/foreign/kan/topics/201106/iaea_houkokusho_e.htm`.

[7] OpenStreetMap. The Free Wiki World Map Website. `http://www.openstreetmap.org/`.

[8] OpenWrt. OpenWrt Wireless Freedom Website. `https://openwrt.org/`.

[9] S. Schildt, J. Morgenroth, W.-B. Pottner, and L. Wolf. IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation. *Electronic Communications of EASST*, 37:1–10, 2011.

[10] S. Yasuda, K. Akashi, T. Miyachi, R. Beuran, Y. Makino, T. Inoue, S. Miwa, and Y. Shinoda. Emulation-based ICT System Resiliency Verification for Disaster Situations. In *Proceedings of the 9th International Conference on Signal Image Technology & Internet based Systems (SITIS 2013), Workshop on Resilient Internet based Systems (REIS 2013)*, 2013.