Disaster Avoidance Control against Tsunami

Phuong Nga Tran
NTT Network Technology Laboratory
Tokyo, Japan
Email: p.tran@lab.ntt.co.jp

Hiroshi Saito
NTT Network Technology Laboratory
Tokyo, Japan
Email: saito.hiroshi@lab.ntt.co.jp

***Abstract*—We investigated challenges in network disaster management against tsunamis. A tsunami is a natural disaster of which the arrival time and devastating effect can be predicted. Based on this prediction, network operators can carry out different disaster management actions to avoid or reduce the damaging effect of tsunami on the network. We developed and evaluated heuristic algorithms to efficiently migrate service (virtual) networks away from a disaster affected area to minimize traffic loss when a tsunami arrives. The problem was first formulated as an integer linear programming problem, which can be solved using optimization solvers for small/medium-sized networks. For large networks, our heuristic algorithms ensure a good solution within a reasonable time. Extensive simulations were carried out to evaluate the performance of our proposed algorithms. On the basis of the tsunami predicted information and network status, network operators can select a suitable algorithm for their disaster management action.**

***Index Terms*—Disaster management, virtual network reconfiguration, tsunami**

I. Introduction

Due to global warming, the world has been impacted by an increasing number of furious natural disasters that do not only take thousands of human lives, but also cause widespread destruction of social infrastructures including communication networks [1]. In spite of the increasing reliability and resiliency of modern communication networks against physical damage, the risk associated with network failures remains serious due to society's growing dependence on communication facilities.

To reduce damage caused by natural disasters, network operators mainly follow two approaches: protection (provisioning) and recovery (restoration) [1]–[3]. The former is to provide spare resources when designing and planning communication networks for backup purpose. The latter includes actions during and after disasters, such as looking for backup resources, to restore networks from failures. These two approaches complement each other and play a central role in disaster management, although they are fundamental parts of fault and recovery management. After a disaster, network operators may also use temporary systems. A transportable terrestrial station of a satellite communication system is such an example.

Saito recently proposed the concept of Disaster-Free Network (DFN) [4] that includes robust physical network design [5] and disaster avoidance control against spatial disasters [6]. The concept of DFN is to avoid or reduce encountering disasters and is different from technologies based on fault and recovery management. In particular, disaster avoidance control is a rather new concept, which aims to avoid disasters. It includes mechanisms/algorithms for migrating network functions and service (virtual) networks away from forecasted disaster areas to avoid or reduce the damaging effect of physical network failures. This idea is well supported by software defined networking (SDN) - an emerging network paradigm [7] - which provides an adequate level of flexibility and programmability in the control plane; hence, it can facilitate the live migration of virtual networks (VNs) and devices.

The idea of disaster avoidance control is also supported by the fact that predicted information on natural disasters is increasingly accurate thanks to advancements in meteorology. Furthermore, historical disaster data and hazard maps describing high-risk areas for each type of disaster is also open to the public. On the basis of this information, network operators can implement different disaster avoidance control algorithms to enhance the robustness of communication networks against certain predictable natural disasters, such as tsunamis, hurricanes and heavy rains. As Mukherjee et al. suggested [2], progress in such technical areas is key to a new direction of disaster management.

In this paper, we propose and investigate a disaster avoidance control algorithms against tsunamis. A tsunami is caused by an earthquake occurring in the ocean that can push up powerful waves. The occurrence of tsunamis can be forecasted because earthquakes causing the tsunami can be observed. Based on earthquake information, such as seismic center, magnitude and observed seismic intensity, the arrival time and height of a tsunami can be predicted. It often approaches land within a few minutes for an earthquake occurring near the coast and nearly a day for an earthquake occurring at the opposite side of the ocean. When a tsunami is forecasted to hit an area and potentially break a section of the network, operators can execute a disaster avoidance control algorithm to relocate service network functions to a disaster-free area to avoid or reduce service network disruption. Since most large cities in Japan are near the sea, the physical network cables are therefore mainly installed along the coast line. The probability of a tsunami reaching and destroying a network cable is hence considerable. This motivated us to carry out this research.

In this study, we concretely investigated the relocation of virtual links in virtual private networks (VPNs) as a use case. Based on the evaluated disconnection probability between two given nodes, virtual links are rerouted to minimize the potential traffic loss when tsunamis arrive. In this research, we

978-0-9883045-1-2/16 $31.00 © 2016 ITC

use tsunami as a use-case. However, our proposed relocation method can be applied to other disasters of which the arrival time can be predicted in advance.

The rest of the paper is organized as follows. A review of related work is presented in Section II, a problem formulation is presented in Section III, our proposed heuristic algorithms are introduced in Section IV, a performance evaluation is discussed in Section V and a conclusion is given in Section VI.

II. Related work

Disaster management in communication networks has become increasing important because the number of natural disasters that may severely damage networks is growing, and at the same time human dependence on communication network is strengthening.

Two popular approaches to disaster management are protection (also called provisioning) and restoration (also called recovery). A thorough survey of these disaster management approaches was conducted [3]. However, these approaches imply that we cannot escape from natural disasters. Therefore, spare resources should be provided to maintain service networks and fast recovery mechanisms are required to minimize the discontinuity of services.

Network applications including grid- and cloud-computing services can be implemented by embedding VNs over physical infrastructure. The protection and restoration problems have evolved to a survivable VN mapping (SVNM) problem [8]–[18], which attempts to protect and recover service networks from failures. Yu et al. proposed four heuristic algorithms for VN mapping, which guarantee 100% recovery against an arbitrary regional failure [8]. Liu et al. proposed two mapping algorithms, minimum link risk prior selection and asymmetric parallel flow allocation, to embed VNs against multiple regional failures [9]. The advantage of their study is to investigate multiple regional failures, which is more adequate regarding huge natural disasters.

Meixner et al. developed disaster-resilient and post-disaster-survivable VN mapping algorithms using a probabilistic disaster model to reduce the expected VN disconnections and capacity loss [10]–[13]. With the post-disaster mapping algorithm, they take into account cascading failures, which may be consequences of a massive disaster. Rahman and Boutaba proposed a mapping algorithm using protection and restoration policies that take into account service level agreement (SLA) to increase long-term business profit of infrastructure provider [14]. Guo et al. [15] investigated a failure dependent protection scheme, in which each primary facility node would have a different backup facility node, as opposed to the failure independent protection. They introduced a so-called Enhanced-VN, which has one node more than requested to protect the original VN. The embedding algorithm is applied to the Enhanced-VN instead of the requested VN. A similar idea was proposed by Yu et al. [16]. However, they investigated a K-redundant scheme rather than the 1-redundant scheme. Guo et al. proposed two survivable mapping schemes, shared on-demand and shared pre-allocation backup, which make better use of substrate resources than the dedicated backup scheme without sharing, [17]. Similarly, Yeow et al. proposed a survivable mapping algorithm based on shared-backup resources to reduce the physical footprint of virtual backups while guaranteeing certain reliability [18]. In [19], Eriksson et al. proposed a framework that uses historical data of natural disasters to evaluate risk routes and to inform network operators with respect to backup paths, route changes and provisioning recommendations.

Disaster avoidance control [4], [6], which aims to migrate network functions and services away from forecasted disaster areas, is a new approach to disaster management. Unfortunately, research on this area is still under explored. In this study, we developed algorithms for disaster avoidance control, which attempts to relocate service networks to avoid disaster areas. In particular, this is the first step of disaster avoidance control against tsunamis and is also the first attempt that takes into account concrete disasters.

III. Disaster Avoidance Control - Model and Problem Formulation

A. Background Information

In this study, we considered regional/nation-wide networks, in which network cables are installed in an underground duct and network nodes are located in a large network building with its own power generator. It is unlikely that the network buildings, which are constructed to withstand the largest earthquake, would be destroyed by earthquakes and tsunamis. Therefore, we considered only link failures.

In Japan, a duct is usually sealed against water. When an earthquake occurs, the duct with cables can be in one of three states: disconnected, damaged (unsealed), or normal. The "disconnected" state means the disconnection of cables in the duct. Thus, service networks will be disrupted. In this case, a recovery mechanism must be carried out to restore the networks. If the duct is just damaged (unsealed), the cables in the duct and the network using them still work. However, if a tsunami hits the cracked point of a damaged duct, the cables in the duct will be disconnected due to water leakage. The corresponding network link will consequently fail. Fortunately, a tsunami arrives within several minutes or hours after an earthquake occurs. The approximated arrival time and height of a tsunami are predictable, although its actual height can be different depending on the local terrain. Therefore, network operators can take appropriate actions to migrate VPN routes away from the tsunami disaster area.

To migrate VPN routes away from a disaster area, the failure probability of each network cable when a tsunami arrives has to be estimated. This is explained in the next sections.

B. Estimation of damage

After a large earthquake occurs, by temporary monitoring and measuring, network operators can detect the disconnection of a cable and damage to ducts. However, the exact place where the duct is damaged and the number of damaged points

are unknown. If damaged parts are in a tsunami-free area, the damage will not result in a disconnection. Otherwise, the network link will be disconnected when tsunamis hit the duct. Therefore, we need to evaluate the damage along the duct/cable to estimate the failure probability of the link. (In the remainder of this paper, we focus on the damaged or normal parts of the network by removing the disconnected links in advance. Therefore, failure means the failure of the cable in the damaged duct.)

After an earthquake, the Japan Meteorological Agency makes an announcement of the earthquake intensity of each city/town. Since we maintain geographical route information of the cable network, we know which part of the network cable will be affected by the earthquake of a particular intensity. (In practice, we can use the geomorphological surface structure data for each square area of $250m \times 250m$ to adjust the earthquake intensity of the square in a city/town.)

The probability that a cable is damaged at location $\mathbf{x}$ depends on the earthquake intensity, terrain conditions at $\mathbf{x}$ such as a river (along a bridge crossing a river), construction methods, such as aerial or underground installation, and network component, such as the type of ducts/cables and years of use. NTT has collected data on the failure probability of network components under various earthquake intensities over several years. Hence, the failure/damage probability of each type of network component under certain conditions including earthquake intensity can be estimated. Consequently, the damage probability $\beta(\mathbf{x})$ of a link segment at $[\mathbf{x}, \mathbf{x}+d\mathbf{x})$ due to an earthquake can be obtained using such field data.

As a result, the link segment at $[\mathbf{x}, \mathbf{x}+d\mathbf{x})$ will be damaged and disconnected by a tsunami with probability $\beta(\mathbf{x})\mathbf{1}(\mathbf{x} \subset \Omega)$, where Ω is the area covered by the forecast tsunami and $\mathbf{1}(x)$ is the indicator function, which is 1 if x is true and 0 otherwise. We assume that the forecasted tsunami area has no forecast error, but a numerical example discussed later will demonstrate the effect of such error.

C. Estimation of physical link failure probability

Let D be the event in which the link is damaged during an earthquake and F be that in which the link will fail due to tsunamis. A link might be damaged by an earthquake but not necessary failed when a tsunami comes. Vice-versa, a failed link due to tsunami is surely damaged by the earthquake. According to Bayes's theorem, the conditional probability that a link fails given that it is damaged is:

$$\Pr(F|D) = \frac{\Pr(D|F) \cdot \Pr(F)}{\Pr(D)}$$

Because the cable will definitely be damaged if it fails due to tsunamis, $\Pr(D|F) = 1$. Therefore,

$$\Pr(F|D) = \frac{\Pr(F)}{\Pr(D)}$$

Assume that damage along a cable independently occurs with probability $\beta(\mathbf{x})$ ($\forall \mathbf{x} \in \mathcal{L}$). Thus, a failure independently occurs along a cable with probability $\beta(\mathbf{x})\mathbf{1}(\mathbf{x} \subset \Omega)$ ($\forall \mathbf{x} \in \mathcal{L}$) where $\mathcal{L}$ is the geographical route of the network cable. Geographical dependence of failures are modeled by the geographical dependence of $\beta(\mathbf{x})\mathbf{1}(\mathbf{x} \subset \Omega)$, although failures are assumed to independently occur.

We have:

$$\Pr(F) = 1 - \prod_{\forall \mathbf{x} \in \mathcal{L}} \big(1 - \beta(\mathbf{x})\mathbf{1}(\mathbf{x} \subset \Omega)\big)$$

$$\Pr(D) = 1 - \prod_{\forall \mathbf{x} \in \mathcal{L}} \big(1 - \beta(\mathbf{x})\big)$$

Hence,

$$\Pr(F|D) = \frac{1 - \prod_{\forall \mathbf{x} \in \mathcal{L}} \big(1 - \beta(\mathbf{x})\mathbf{1}(\mathbf{x} \subset \Omega)\big)}{1 - \prod_{\forall \mathbf{x} \in \mathcal{L}} \big(1 - \beta(\mathbf{x})\big)} \tag{1}$$

According to these link failure probabilities, we have an overview in which part of the network is under risk of failure and another part is risk-free. Thus, a disaster avoidance control algorithm will be executed correspondingly.

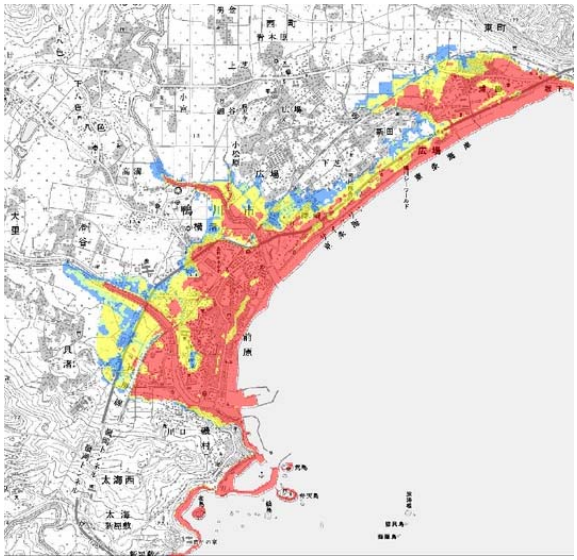

Fig. 1. Affected area of 10-m-high tsunami in Japan [22]

D. Problem Formulation

Given a physical network with a set of nodes $\mathcal{V}$ and links $\mathcal{E}$, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Each physical link $e \in \mathcal{E}$ has a capacity of C_e. The failure probability of each physical link, $\beta(e)$ ($e \in \mathcal{E}$), is given by Eq. (1) where $\mathcal{L}$ is a route of $e \in \mathcal{E}$.

Assume that a set of virtual links $\mathcal{S} = \{s\}$ is embedded on the given physical network. Each virtual link $s \in \mathcal{S}$ has a required bandwidth of b^s and a set of possible physical routes $\mathcal{R}^s = \{r\}$. If $s \in \mathcal{S}$ is embedded on a physical route $r \in \mathcal{R}^s$, its failure probability f_r^s under the assumption of independent failure is calculated as:

$$f_r^s = 1 - \prod_{e \in r} \beta(e) \tag{2}$$

where $e \in r$ indicates that $e \in \mathcal{E}$ belongs to $r \in \mathcal{R}^s$.

If two physical links (e.g. e_1, e_2) partly share the same duct, their failures will no longer be independent; hence, Eq. (2) cannot be directly applied. Assume that the shared part is e^*, we first need to evaluate the failure probability of three independently failed physical parts, which are $(e_1 - e^*)$, $(e_2 - e^*)$, and e^* (using Eq. (1)), where $(e_1 - e^*)$ and $(e_2 - e^*)$ are the non-shared parts of e_1 and e_2, respectively. Equation (2) is then applied to these three independently failed parts.

From the given physical network topology and set of virtual links, we compute a parameter $\delta^s_{r,e}$. $\delta^s_{r,e} = 1$ if $e \in \mathcal{E}$ belongs to $r \in \mathcal{R}^s$, and $\delta^s_{r,e} = 0$ otherwise.

Let x^s_r be a binary variable that takes the value of 1 if virtual link $s \in S$ is embedded on $r \in \mathcal{R}^s$ and 0 otherwise.

$$\min \sum_{s \in \mathcal{S}, r \in \mathcal{R}^s} x^s_r \cdot f^s_r \cdot b^s \tag{3}$$

$$\text{subj. to} \sum_{r \in \mathcal{R}^s} x^s_r = 1 \quad \forall s \in \mathcal{S} \tag{4}$$

$$\sum_{s \in \mathcal{S}, r \in \mathcal{R}^s} x^s_r \cdot \delta^s_{r,e} \cdot b^s \leq C_e \quad \forall e \in \mathcal{E} \tag{5}$$

The objective is to minimize the expected traffic loss when a tsunami arrives. The first constraint (Eq. 4) is to guarantee that all virtual links are successfully embedded on the physical network. The second constraint (Eq. 5) is a capacity constraint that ensures that the total required bandwidth of virtual links embedded on a physical link does not exceed the capacity of the physical link.

With the above formulation, it is assumed that all virtual links can be reconfigured. This will result in the best solution. However, the number of virtual links is often very large and network operators may want to reconfigure only virtual links that are under risk of disconnection when a tsunami arrives. In this case, the set of reconfigured virtual links becomes $\mathcal{S}^* (\subset \mathcal{S})$, which contains only virtual links in tsunami-affected areas, and the capacity constraint Eq. 5 is rewritten as:

$$\sum_{s \in \mathcal{S}^*, r \in \mathcal{R}^s} x^s_r \cdot \delta^s_{r,e} \cdot b^s \leq C^*_e \quad \forall e \in \mathcal{E} \tag{6}$$

where C^*_e is the residual link capacity (the bandwidth allocated to risk-free virtual links is subtracted from link capacity C_e).

IV. Heuristic Algorithms

While optimization solvers, such as Cplex [20] and Glpk [21], can obtain the optimal solution for virtual link reconfigurations, the integer linear programming (ILP) formulation presented in Section III-D exhibits a high computation time because the problem is NP-hard. Therefore, we propose the following heuristic algorithms to provide approximate solutions in a reasonable time, which is more applicable for large networks.

A. Iterative Reconfiguration

In this section, we present our two heuristic algorithms based on an iterative approach, namely, step-by-step searching for a new physical route of each virtual link in a certain order. The first algorithm is called Relocation Before Release (RBR) and the second is Reconfiguration After Release (RAR).

Algorithm 1: Reconfiguration Before Release (RBR)

Data: Physical network topology $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and link capacity C_e ($e \in \mathcal{E}$), failure probability of each physical link $\beta(e)$ ($e \in \mathcal{E}$), virtual links $\mathcal{S} = \{s\}$ with current physical route $r_s \in \mathcal{R}^s$ and required bandwidth b^s ($s \in \mathcal{S}$)
Result: New embedding solution for $\mathcal{S}$
Select virtual links in disaster-prone areas $\mathcal{S}^* = \{s^*\}$
Compute expected traffic loss of each virtual link $T^{s^*} = f^s_r b^s$ ($s^* \in \mathcal{S}^*$)
Sort virtual links by T^{s^*} in descending order
for $s^* \in \mathcal{S}^*$ **do**
  Find all physical routes $\mathcal{R}^s = \{r\}$ for s^*
  Sort physical routes by their number of hop-counts in ascending order
  for $r \in \mathcal{R}^s$ **do**
    if r *has enough bandwidth for* s^* **then**
      Compute expected traffic loss $T^{s^*}_r$ of s^* if embedded on r
  Select r ($r \in \mathcal{R}^s$) that has lowest expected traffic loss as the final embedding solution for s^*
  Update residual capacities of physical links

The main difference of the two algorithms is that with RBR, we keep the current mapping and step by step reconfigure each virtual link if a better solution can be found, while with RAR, we assume that no virtual links in disaster areas are embedded and then step-by-step search is conducted for the best embedding solution for each virtual link. With RBR, we can guarantee that all virtual links will be embedded because if we do not find a better route for a virtual link, it can stay with the old embedding solution. With RAR, this does not apply. Since we release all virtual links under risk, links that are embedded at the last step may not find an available route. However, we expect that if all virtual links can be embedded, RAR will result in a better solution because there are more free link capacities during embedding, which lead to more possibilities to find a good solution. This effect is investigated in more detail in Section V.

B. Grouping Reconfiguration

With RBR and RAR, we relocate each virtual link separately. In fact, the number of virtual links can be quite large. However, the number of source-destination pairs in a realistic physical network is not that large. Let us consider a realistic regional network in Japan, which has a ladder topology of

Algorithm 2: Reconfiguration After Release (RAR)

Data: Physical network topology $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and link capacity C_e ($e \in \mathcal{E}$), failure probability of each physical link $\beta(e)$ ($e \in \mathcal{E}$), virtual links $\mathcal{S} = \{s\}$ with current physical route $r_s \in \mathcal{R}^s$ and required bandwidth b^s ($s \in \mathcal{S}$)

Result: New embedding solution for $\mathcal{S}$

Select virtual links in disaster-prone areas $\mathcal{S}^* = \{s^*\}$

Sort virtual links by b^s in descending order

Release $s^* \in \mathcal{S}^*$ from the physical network and update residual capacities of physical links

for $s^* \in \mathcal{S}^*$ **do**

  Find all physical routes $\mathcal{R}^s = \{r\}$ for s^*

  Sort physical routes by their number of hop-counts in ascending order

  for $r \in \mathcal{R}^s$ **do**

    if r *has enough bandwidth for* s^* **then**

      Compute expected traffic loss $T_r^{s^*}$ of s^* if embedded on r

  Select r ($r \in \mathcal{R}^s$) that has lowest expected traffic loss as the final embedding solution for s^*

  Update residual capacities of physical links

12 nodes and 14 links. In this network, there are 66 source-destination pairs. Each pair has an average of 5.4 (a maximum of 8) different physical routes. Therefore, the number of distinct virtual links in terms of source, destination and physical route in this example is just approx. 355, which seems much smaller than a typical number of virtual links. Based on this observation, we propose another heuristic algorithm called "grouping reconfiguration". The main idea is to group virtual links that originate and terminate at the same nodes and follow the same physical route to form an equivalent virtual link. The reconfiguration process will be done on the equivalent virtual links, instead of each actual virtual link.

Algorithm 3: Grouping Reconfiguration Algorithm

Data: Physical network topology $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and link capacity C_e ($e \in \mathcal{E}$), failure probability of each physical link $\beta(e)$ ($e \in \mathcal{E}$), virtual links $\mathcal{S} = \{s\}$ with current physical route $r_s \in \mathcal{R}^s$ and required bandwidth b^s ($s \in \mathcal{S}$), *grouping factor* ν

Result: New embedding solution for $\mathcal{S}$

Select virtual links in disaster-prone areas $\mathcal{S}^* = \{s^*\}$

Group virtual links by the given grouping factor ν

Reconfigure equivalent virtual links using either RBR, RAR, or Optimization solvers

For this algorithm, we introduce a parameter called *grouping factor* ν. This indicates the number of virtual links that are grouped together to form an equivalent virtual link. Higher ν results in a smaller number of equivalent virtual links, which leads to a lower computation time. In certain cases, we can use an LP solver to solve the reconfiguration problem if the number of equivalent virtual links is not so large. However, we also expect that the performance will decrease with an increasing grouping factor.

The complexity of all three heuristic algorithms is $\mathcal{O}(|S||R^s|)$.

V. Performance Evaluation

A. Simulation scenarios

In this paper, we focus on the regional network of 12 nodes and 14 links in Fig. 2.

We assume all physical links have the same capacity of 100 units.

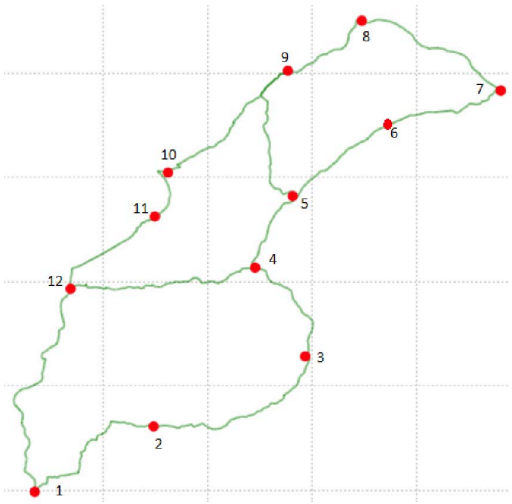

Fig. 2. Regional network for numerical studies

We consider three earthquake types that will occur with probability of at least 1% according to J-SHIS [23]. These earthquakes and their associated tsunamis will occur at different areas with different intensities and can damage various physical links. The devastating effect of tsunamis can be found at [22].

In reality, after an earthquake occurs, network operators will monitor and measure network status to determine which link is disconnected or damaged. They can the use their field data, together with the earthquake and tsunami hazard maps, and geographical route information to estimate the failure probabilities of physical network links. In this study, we just used hazard maps and route information to determine the links in disaster-affected areas. Due to the lack of concrete field data and for the sake of simplicity, we assume that $\beta(e)$ is 0.5 for all e.

Virtual links are randomly generated as follows: 1) source and destination nodes are evenly distributed in the network; 2) the required bandwidth of a virtual link is uniformly distributed in [0.01 - 0.1]; 3) virtual links are embedded on the shortest available physical routes. We generated from 1000 to 9000 virtual links, which leads to different traffic load in the

network. Traffic load is defined as the ratio of total allocated bandwidth to the total link capacity. Each result is obtained by taking the average results from 1000 same experiments.

B. Reduction in expected traffic loss

We first evaluated the performance of our proposed algorithms in terms of expected traffic loss under three earthquakes with their associated tsunamis. The effect of the three earthquakes on the physical network is as follows:

[Scenario 1] Local earthquake 1 occurs to the east of the network. No disconnected links but three links (1-12), (12-11) and (11-10) are damaged in the forecasted tsunami area.

[Scenario 2] Local earthquake 2 occurs to the east of the network. No disconnected links but two links (12-11) and (11-10) are damaged in the forecasted tsunami area.

[Scenario 3] Huge earthquake occurs to the south of the network. No disconnected links but six links (1-12), (12-11),(11-10), (1-2), (2-3) and (3-4) are damaged in the forecasted tsunami area.

In this experiment, we applied RBR. The result are illustrated in Fig. 3. The y-axis shows the ratio of the expected traffic loss after reconfiguration compared to that when no reconfiguration was carried out. Obviously, when the network load is low, it is easy to find alternative routes that go through disaster-free areas. Consequently, the expected traffic loss after reconfiguration is also low. In contrast, when the load is high, the reduction in expected traffic loss decreases because there are less free resources for relocating virtual links. From the network load of approx. 70% or more, it is almost impossible to migrate virtual links away from the disaster areas. This implies that to make disaster avoidance control effective, network operators should provision enough redundant resources for migration.

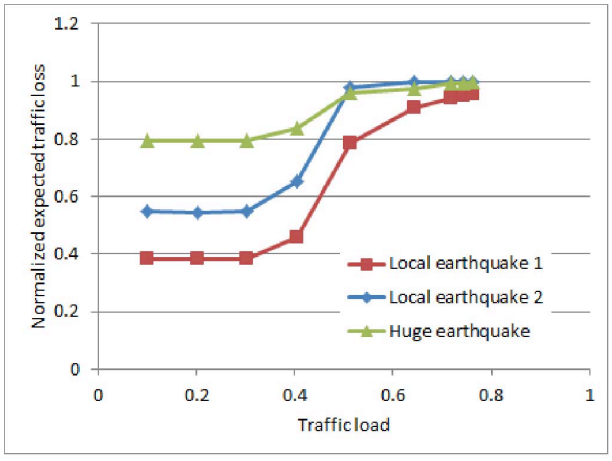

Fig. 3. Reduction in expected traffic loss vs. network load during different earthquakes

In this scenario, even when the network load is very low (only 10%), it is still impossible to migrate all virtual links away from the disaster areas. This is because several virtual links originate or terminate at node 11 (or node 2 & 3 in case of huge earthquake), which is connected by two links that are both under risk of failure. Therefore, it is impossible to make virtual links originate and terminate at node 11 risk-free.

The results in this simulation reveal that for the investigated regional network, it is impossible to migrate all virtual links to disaster-free areas. Nevertheless, this gives some hints to network operators on how to further improve network robustness. A solution can be installing an additional link connecting to node 11 that the tsunami does not reach.

C. Comparison of heuristic algorithms

The difference in the solution with our proposed heuristic algorithms and the optimal solution obtained using Glpk solver are presented in Fig. 4. In this experiment, we considered Local earthquake 1 as a use case. Since there are thousands of virtual links, it is impossible to directly use Glpk solver to solve the problem. We therefore applied grouping reconfiguration to reduce the number of virtual links so that the problem can be solved by using LP solver. We used a very high grouping factor, which groups all virtual links of the same source-destination and physical paths into an equivalent virtual link. This also applies for the results of grouping reconfiguration shown in Fig. 4. In this experiment, the grouping approach used RBR to do the reconfiguration.

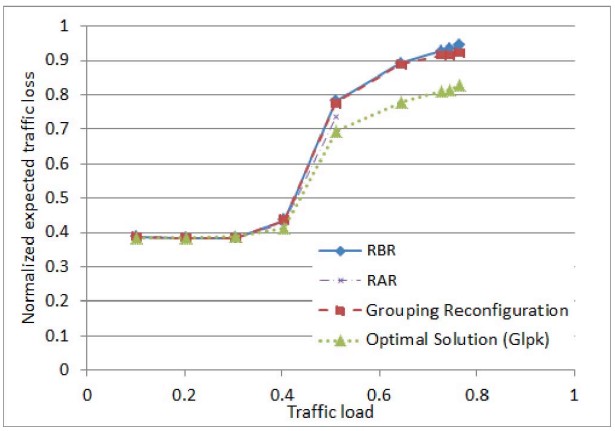

Fig. 4. Comparison of solutions with heuristic algorithms and optimal solution

Figure 4 shows that at low network load ($\leq$ 30%), all algorithms performed similarly. This is easy to understand, because there are plenty of free resources for the migration, so it is possible to find the best alternative routes for all virtual links. Hence, the obtained solution is also the optimal solution.

When the load increased, we observed a clear difference in the solution with the heuristic algorithms and the optimal solution. The difference was approx. 10% at a load of approx. 65% and above.

At middle load (40-50%), RAR performed better than RBR. The difference increased with network load. This is because RAR releases resources of all virtual links before re-embedding them, which leads to higher flexibility during

the reconfiguration. However, when the network load was very high, RAR resulted in infeasible solutions, namely some virtual links could not be embedded on physical networks. As a result, RAR should be chosen when the network is middle loaded and too large to be solved with an LP solver. However, the choice of algorithm also depends on the requirements of VNs. If "connect before break" is required to maintain service continuity, we have to use RBR. The optimal solution with Glpk also cannot be used in this case, since this approach implies that all virtual links should be released before being re-embedded.

It is surprising that grouping reconfiguration performed very similarly to RBR. This can be explained as follows. Let us consider a network at a load of approx. 70%. We generated 7000 virtual links, among which approx. 3000 virtual links needed to be reconfigured. In this regional network, there were roughly 300 distinct virtual links. Using the grouping approach, an equivalent link will contain approx. 10 virtual links on average and its bandwidth will be approx. 0.5 (since each virtual link has a bandwidth uniformly distributed in [0.01-0.1]). At a network load of 70%, there were approx. 30 units of free capacity on each link on average. This is much larger than the required bandwidth of an equivalent virtual link. Therefore, we observed almost no difference between the two algorithms. Furthermore, the greedy behavior of RBR may even favor the grouping approach. This explains why at a high load (higher than 70%), grouping reconfiguration performed slightly better than RBR without grouping.

D. Grouping reconfiguration with different grouping factors

In the previous sections, we mentioned that grouping reconfiguration performed very similarly to RBR. We therefore expect no notable difference in performance when applying different grouping factors. However, if we use an LP solver to solve the problem, it is expected that the higher the grouping factor, the worse the performance.

Figure 5 presents the performance of grouping reconfiguration with three different grouping factors (non-grouping (factor 0), group 2 virtual links into one (factor 2), and group all identical virtual links) at high load. In this experiment, we generated 700, 800, and 900 virtual links with bandwidth in the range [0.1-1]. We reduced the number of links so that we could obtain the solution with Glpk solver, but we increased the virtual link bandwidth to achieve a high network load. Links were grouped randomly; therefore, equivalent links had the bandwidth in [0.2-2].

As expected, there was a difference in performance of the algorithm with different grouping factors. The difference is relatively small (approx. 2.5% for grouping and non-grouping). The reason is due to the uniform distribution of source-destination pairs and requested bandwidth of virtual links, as discussed in Section V-C.

In fact, the difference in the performance among different grouping factors depends strongly on the distribution of source-destination pairs of virtual links. If some source-destination pairs have a significantly higher number of requested virtual links, the performance gap between high and low grouping factors is expected to be clearly larger. To demonstrate this, we carried out an experiment in which the number of virtual links between nodes (1) and (10) takes 50% of the total requested virtual links, and the remaining virtual links are uniformly distributed among other pairs. The results are shown in Fig. 6. As expected, the performance gap was larger than 10% in the range of middle traffic load. In this case, it makes sense to choose different grouping factors according to the available computation time.

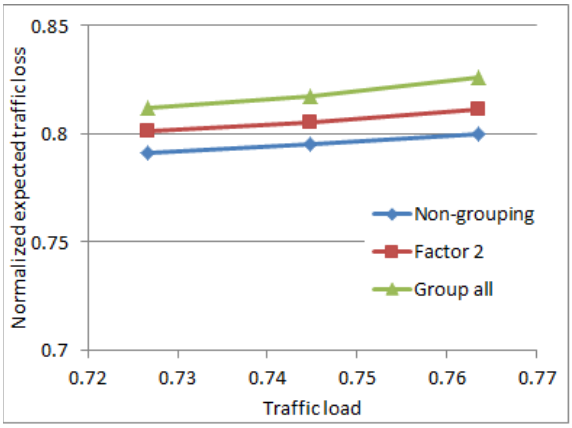

Fig. 5. Performance of grouping reconfiguration with different grouping factors - Uniform distribution of source-destination pairs. VLs = virtual links

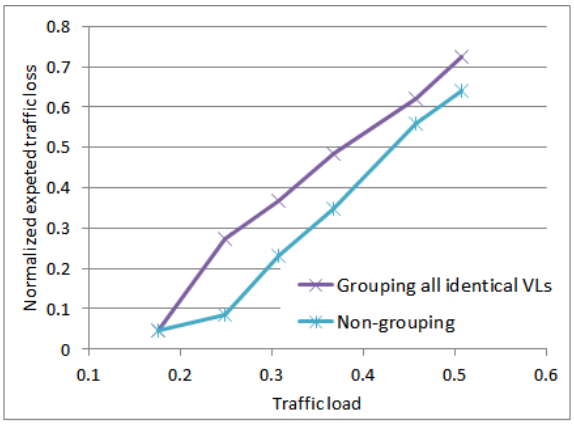

Fig. 6. Performance of grouping reconfiguration with different grouping factors - Non-uniform distribution of source-destination pairs

E. Effect of Estimation Error of Link Failure Probability on Performance

The failure probability for a damaged link is estimated on the basis of collected field data and predicted natural hazard information. These data are never 100% accurate; hence, the estimated link failure probability will have errors. In this section, we investigate the effect of estimation error on the

performance of RBR. The effect on the performance of RAR and grouping reconfiguration is expected to be similar.

We considered Scenario 1 (as described in Section V-B). For a given link failure probability, we added an estimation error, which is uniformly distributed in $[-\epsilon, \epsilon]$. We also generated two cases in which a link under failure risk is considered as risk-free and a risk-free link is considered to fail with high probability. The reconfiguration was done on the basis of failure probabilities with errors. We then generated failed links according to the given probabilities (without error) and calculated the traffic loss for each reconfiguration.

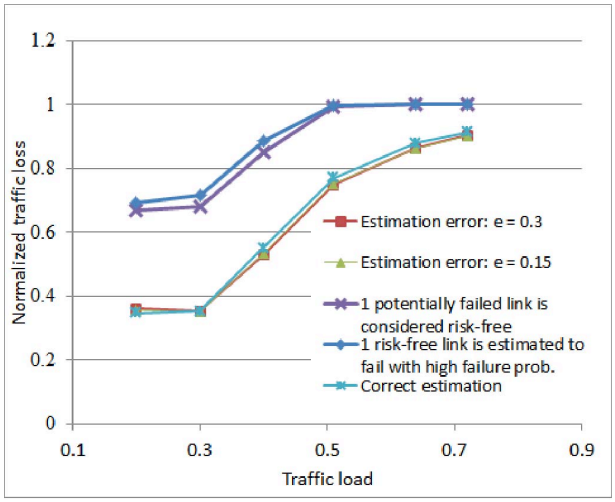

Fig. 7. Performance of RBR with different estimation errors of failure probability

The results are shown in Fig. 7. They revealed that the accuracy of failure probability had little effect on the performance of the proposed algorithms in terms of traffic loss. The important factor is to accurately detect links under failure risk and their relative failure probabilities. This is because the algorithms always attempt to relocate virtual links to disaster-free areas regardless of whether the failure probability of physical links is small or large, or if not possible, to areas where physical links fail with lower failure probabilities. Therefore, when a link may fail but is considered as risk-free link, virtual links embedded on it will not be migrated and even worse, other virtual links may be relocated to this link, which leads to further loss. Similarly, if a link does not fail but is estimated to fail with a high probability, virtual links on it will be relocated to lower risk links, if it is not possible to relocate them all to risk-free links. This explains why traffic loss in these cases is much higher .

We expect that detecting a potentially broken cable due to a tsunami and its degree of failure (failed with high probability or low probability) is much simpler than estimating its accurate failure probability. This proves that our proposal is practically efficient.

F. Computation time

In this section, we investigate the computation time of our proposed algorithms. We carried out an experiment on two networks, one regional network of 12 nodes and 14 links, and one nation-wide network of 18 nodes and 21 links under the assumption that three links are under risk with a failure probability of 0.5. Obviously, the computation time depends on the number of virtual links that need to be relocated and the number of alternative physical routes of the virtual links, which strongly depends on network topology.

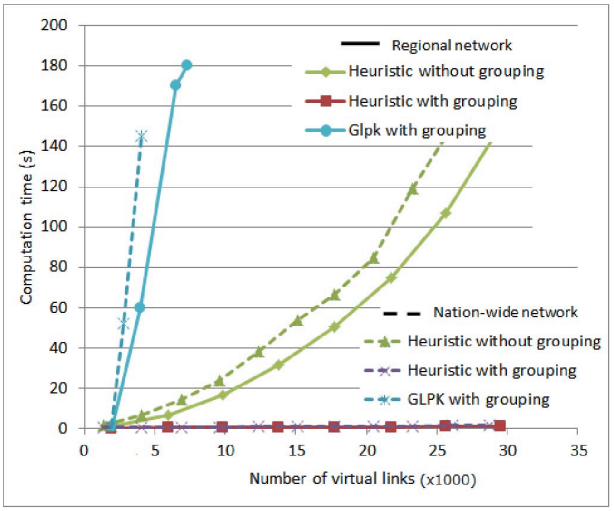

Fig. 8. Computation time for different networks and number of virtual links

The results are shown in Fig.8. Only the computation time of RBR is shown in the graph but that of RAR is expected to be the same. The computation time of our heuristic algorithms is predictable and increases with the number of virtual links and physical paths. With grouping reconfiguration (in which all identical virtual links are grouped into one equivalent link), the computation time of the heuristic algorithms reduced significantly and remained almost stable because the number of equivalent virtual links stayed almost constant and did not depend on the number of actual virtual links.

Without grouping reconfiguration, Glpk solver running on a quad-core 2.2GHz processor computer hardly obtained a solution within an hour in case of thousands of virtual links. With grouping reconfiguration, the number of variables decreased significantly and a solution was obtained within minutes. However, the computation time increased drastically with the number of virtual links, even though we grouped all identical virtual links, which resulted in an almost constant number of equivalent virtual links. When the number of actual virtual links increased, the traffic load also increased. At high network load, there might be many similar embedding solutions than that at low network load, which would result in a long searching time for the branch-and-bound algorithm (algorithm implemented in Glpk).

G. Which algorithm should be used?

There are several factors that effect the decision of choosing an appropriate algorithm. We now introduce a framework, which takes the following factors as inputs for the decision making process:

- Requirements of the reconfiguration process
- Network load
- Available computation time
- Network topology and number of reconfigured virtual links

If it is required to always set up a new virtual link before tearing down the old one, RBR must be used even though it yields the lowest reduction in expected traffic loss at a high network load.

If the network load is low ($\leq 30\%$), we can apply RBR (or RAR) with grouping reconfiguration, since the performance is the same as the optimal one but the computation time is small. If the network load is above 40%, the computation time should be taken into account to achieve the best possible performance.

The available computation time is the predicted arrival time of tsunamis subtracted by the time required to finish the reconfigurations.The computation time strongly depends on the network topologies and the number of relocated virtual links as illustrated in Fig. 8. If the computation time is critical, i.e., a tsunami is forecasted to arrive within minutes, grouping reconfiguration combined with either RAR or RBR should be used. However, if a tsunami is predicted to arrive in a day, the grouping approach combined with an optimization solver can be used in order to obtain a better solution than with the heuristic algorithms. Thanks to the development of cloud computing, which can provide very high computational power on demand, the problem is expected be solved within a reasonable time. In addition, a better LP solver, such as Cplex, can also solve the problem faster than Glpk.

VI. Conclusion

We proposed disaster avoidance control algorithms against tsunamis, which contribute to the realization of the DFN concept. Our focus was to develop efficient algorithms for migrating virtual links away from tsunami-affected areas. The objective of the algorithm was to minimize the expected traffic loss, given the information on the devastating effect of a tsunami. We formulated the problem as an ILP problem and proposed three heuristic algorithms (RAR, RBR, and grouping reconfiguration) to solve it in case of large networks. The simulation results revealed that when the network is lightly loaded, all algorithms performs similarly, and when the network is heavily loaded, the algorithm resulting in a lower expected traffic loss often takes longer time. On the basis of predicted arrival time of tsunamis and network status, network operators can choose an appropriate algorithm for reconfiguration, which performs well but also fast enough to avoid tsunamis.

This study showed that RAR (and optimization solver) performs better than RBR but does not guarantee no disrupted service during the reconfiguration. However, we expect that if the reconfiguration of virtual links is done in an appropriate order, we can obtain a good solution while still ensuring the "connect before break" requirement. Therefore, a future research direction can be to guarantee no service disruption during the reconfiguration process while minimizing expected traffic loss at the same time.

References

[1] E. Asimakopoulou, et al., A Collective Intelligence Resource Management Dynamic Approach for Disaster Management: A Density Survey of Disasters Occurrence, IEEE INCoS, pp. 735–740, 2011.

[2] B. Mukherjee, et al, Network Adaptability from Disaster Disruptions and Cascading Failures, IEEE Communications Magazine, pp. 23 - 238, May 2014

[3] M. F. Habib, et al., Disaster survivability in optical communication networks, Computer Communications, 36, pp. 630 - 644, 2013.

[4] H. Saito, Concept and Implementation of Disaster-free Network, Keynote, 11th DRCN, Kansas City, USA, 2015.

[5] H. Saito, Spatial Design of Physical Network Robust against Earthquakes, IEEE Journal of Lightwave Technology, 33, 2, pp.443-458, 2015.

[6] H. Saito, et al., Proposal of Disaster Avoidance Control, Networks 2014, Funchal, Madeira Island, Portugal, Sept, 2014.

[7] K. Nguyen, et al., A Software-Defined Networking Approach for Disaster-Resilient WANs, in Proceeding of 22th International Conference on Computer Communications and Networks (ICCCN), Bahamas, Aug. 2013

[8] H. Yu, et al., Regional Failure-Resilient Virtual Infrastructure Mapping in a Federated Computing and Networking System, IEEE/OSA Journal of Optical Communications and Networking, Vol. 6, Issue 11, pp. 997-999, 2014.

[9] X. Liu, et. al., Disaster-Prediction Based Virtual Network Mapping against Multiple Regional Failures, IEEE/IFIP IM'15, Ottawa, Canada, May 2015.

[10] F. Dikbiyik, et al., Minimize the Disaster Risk in Optical Telecom Networks,IEEE OFC/NFOEC, Los Angeles, USA, 2012

[11] C. C. Meixner, et al., Disaster-resilient virtual-network mapping and adaptation in optical networks, in Proceeding of 17th Optical Network Design and Modeling (ONDM), Brest, France, April 2013

[12] C. C. Meixner, et al., Disaster-survivable cloud-network mapping, Journal of Photonic Network Communications, Vol. 27, No. 3, pp.141-153, June 2014

[13] F. Dikbiyik, et al., Minimizing the Risk From Disaster Failures in Optical Backbone Networks, in Lightwave Technology, Journal of , Vol.32, No.18, pp.3175-3183, 2014

[14] M. R. Rahman, et al., SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization, IEEE Trans. on Networks and Service Management, Vol. 10, No. 2, 2013

[15] B.Guo, et al., Survivable Virtual Network Design and Embedding to Survive a Facility Node Failure, Journal of Lightwave Technology, Vol. 32, No. 3, 2014

[16] H. Yu, et al., Cost efficient design of survivable virtual infrastructure to recover from facility node failures, in IEEE International Conference on Communications (ICC), Kyoto, Japan, June 2011

[17] T. Guo, et al., Shared backup network provision for virtual network embedding, in IEEE International Conference on Communications (ICC), Kyoto, Japan, June 2011

[18] W.L. Yeow, et al., Designing and Embedding Reliable Virtual Infrastructures, ACM Computer Communication Review, Vol. 41, No.2, 2011

[19] B. Eriksson, et al., RiskRoute: A Framework for Mitigating Network Outage Threats, in ACM CoNEXT'13, California, USA, Dec 2013

[20] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[21] https://www.gnu.org/software/glpk/

[22] http://www.bousai.pref.chiba.lg.jp/portal/05_sonae/58_hazard/tnm/tnm1.html?/7/4/8

[23] http://www.j-shis.bosai.go.jp/en/

[24] https://www.openstreetmap.org/