Evaluating Local Disaster Recovery Strategies

Jorik Oostenbrink
Delft University of Technology
J.Oostenbrink@tudelft.nl

Fernando A. Kuipers
Delft University of Technology
F.A.Kuipers@tudelft.nl

Poul E. Heegaard
Norwegian University of Science and Technology
poul.heegaard@ntnu.no

Bjarne E. Helvik
Norwegian University of Science and Technology
bjarne@ntnu.no

ABSTRACT

It is of vital importance to maintain at least some network functionality after a disaster, for example by temporarily replacing damaged nodes by emergency nodes. We propose a framework to evaluate different node replacement strategies, based on a large set of representative disasters.

We prove that computing the optimal choice of nodes to replace is an NP-hard problem and propose several simple strategies. We evaluate these strategies on two U.S. topologies and show that a simple greedy strategy can perform close to optimal.

1. INTRODUCTION

In the last decades communities worldwide have become more and more dependent on communication networks to communicate, coordinate and stay informed, even more so during and after disasters [9]. Yet, the disaster itself can cause significant damage to network infrastructure, disconnecting whole portions of the network.

Repairing a network can take days to months, during which functionality is only slowly restored. Thus, there is a need for a simultaneous quick response to recover a bare amount of network functionality in the affected areas as quickly as possible.

In this paper, we consider the possibility of temporarily replacing some of the failed network components by emergency equipment, such as MDRUs [8]. We propose a framework to evaluate different recovery strategies, based on a set of representative disasters. The evaluation only considers the effect of the recovery on the network area enclosing the disaster region, as the focus of these recovery efforts is to restore vital network functionality to the affected area.

Using our framework, network operators can decide beforehand which strategy they want to employ, such that after a disaster the strategy can be implemented immediately.

Our main contributions are as follows:

- We propose a model (section 2) and algorithm (section 4) for evaluating the effectiveness of a quick recovery strategy.

- We describe an optimal strategy as an optimization problem (section 3.1), and prove that it is NP-hard (appendix A).

- As computation resources in and communication within and from a disaster region are limited, we propose alternative simple strategies (section 3.2).

- We apply our framework to two U.S. topologies, and evaluate our strategies.

While there has been other work on network recovery strategies after a large-scale disaster (e.g., [4, 5, 10]), to the best of our knowledge we are the first to propose an evaluation framework for different strategies, as well as the first to focus on a local area enclosing the disaster region.

2. EVALUATION MODEL

2.1 Model

We model a telecommunications network as an undirected graph $G = \{V, E\}$ of nodes V connected by links E. The nodes of the network are the routing and computing nodes of the network, as well as its base stations, while the edges are the cables (or radio links) connecting them.

To evaluate different strategies to a wide range of possible situations, we work with a representative set of disaster scenarios D, as was done in [7]. These can for example be historical disasters, randomly sampled disasters, or specific scenarios created by experts.

Each disaster $d \in D$ affects a region of the network, called the disaster region. We assume all nodes in the affected region fail. Links remain unaffected in our model, as (underground) cables have been proven to be relatively resistant to earthquakes and tsunamis [8]. However, our methodology is easily extended to other damage patterns as well.

We assume exactly one disasters occurs and that we are given the occurrence probability $P(d)$ of each disaster $d \in D$. We use these probabilities to weigh the relative importance of each disaster to the overall evaluation of a recovery strategy.

To quickly recover the functionality of the network, damaged nodes can be replaced by temporary emergency nodes, such as MDRUs [8]. The exact functionality (e.g. base station, router, edge computing) of these nodes would depend on the node it replaces. To connect the emergency node to the rest of the network, the cable to the old node will be dug up, spliced, and connected to the emergency node. The emergency nodes can have a smaller capacity as the node they replace, as long as it at least takes over some bare minimum of its functionality.

In the case of a disaster, large amounts of manpower will be made available to recover the network. However, the

number of other resources available might be more limited. As such, we assume that only K temporary nodes can be placed, but the process of placing these K nodes can be worked on simultaneously.

The time it takes to place and connect an emergency node depends on both the reachability of its intended location, as well as the properties of the area and soil around it. For example, it could take much more time to place a device on top of a mountain than on an area of farmland. We assume we are given a cost(v) for each $v \in V$, where cost(v) is the time it takes to replace node v.

Let $A(d)$ be all nodes affected by disaster $d \in D$. The choice to be made after a disaster, using a recovery strategy, is the set of at most K nodes out of $|A(d)|$ to replace.

Given such a choice of actions, the state of the network after a disaster d can be described by a vector

$$s(d) = [(G_1, 0), (G_2, t_2), \ldots, (G_{K+1}, t_{K+1})] \quad (1)$$

of length $K + 1$. Where G_1 is the topology of the network directly after the disaster, i.e. the graph G minus the affected nodes. G_2 is the topology of the network at time t_2, directly after the first recovery action has been completed, G_3 is the topology of the network at time t_3, directly after the second recovery action has been completed, etc.

2.2 Local Area

The focus of recovery efforts is to restore vital network functionality to the local affected area. However, it is also important to consider those nodes that are disconnected by the disaster, but are not in the disaster region, and are thus still functioning. The most effective method to reconnect these nodes will be through the disaster region.

As such, we only consider the placement of emergency equipment and the effect of this equipment in a local area around the disaster region. By limiting ourselves to a smaller area, we also limit the size of the graph we need to consider when determining where to place the emergency nodes and when evaluating the effectiveness of the approach, thus reducing the amount of processing time required, and increasing the level of network details that can be considered.

Specifically, we define the local nodes $V_L \subseteq V$ after a disaster as the nodes of the network that are directly affected by the disaster ($A(d)$), or are distanced only 1 hop from an affected node. Thus the local network of interest is $\{V_L, E_L\}$, where $E_L = \{(v, x) \in E | v, x \in V_L\}$.

2.3 Evaluation Metrics

Nodes that are cut off by the disaster, but are not part of the local area, still need to be reconnected to the rest of the network. This is taken into account by increasing the weights of nodes on the border in proportion to the portion of the network they connect to the local area.

Let $p(v)$ be the weight of node v in G. Define

$$\mathcal{C}(v) := \{x \in V | h(v, x) \leq h(y, x), \forall y \in V_L\} \quad (2)$$

as the nodes closest to node v, where $h(v, x)$ is the smallest number of hops from v to x in G.

Now, the weight of node $v \in V_L$ is set to

$$w(v) = \sum_{x \in \mathcal{C}(v)} n(x)p(x) \quad (3)$$

where

$$n(x) = \frac{1}{|\{v \in V_L | x \in \mathcal{C}(v)|}} \quad (4)$$

Note that $w(v) = p(v)$ for all nodes in the disaster region itself. These weights can be seen as representative for the amount of traffic demand we expect to/from the nodes.

Functioning nodes in the large connected component will have a much higher weight than other functioning nodes, which in turn generally have a higher weight than the nodes in the disaster area. Thus, by setting these weights, we prioritize connecting areas to the core network and connecting the smaller components to the giant connected component.

Our framework can be used with any network metric. In this paper we consider a weighted version of the Average Two-Terminal Reliability (ATTR).

DEFINITION 1. *Weighted Average 2-Terminal Reliability (WATTR)*
Let

$$I(v, x) = \begin{cases} 1 & \text{if node } v \text{ is connected to node } x \\ 0 & \text{otherwise} \end{cases}$$

The weighted average 2-terminal reliability (WATTR) is defined as

$$WATTR := \frac{1}{W} \sum_{v \in V_L} \sum_{x \in V_L - \{v\}} w(v)w(x)I(v, x) \quad (5)$$

where $W := \sum_{v \in V_L} \sum_{x \in V_L - \{v_i\}} w(v)w(x).$

WATTR can be seen as a measure of the proportion of (potential) connections in a network that are still functioning.

If we let $C \subseteq V_L$ be the set of all connected components of the network in V_L and define $\text{sum}(c) := \sum_{v \in c} w(v)$ for all $c \subseteq V_L$. Then

$$W = \sum_{v \in V_L} w(v) * (\text{sum}(V_L) - w(v)) \quad (6)$$

and

$$\text{WATTR} = \frac{1}{W} \sum_{c \in C} \sum_{v \in c} w(v) * (\text{sum}(c) - w(v)) \quad (7)$$

The metric evaluates the network at a specific state. To evaluate the complete emergency recovery process, we use a weight function $\mathcal{W} : \mathbb{R}^+ \to \mathbb{R}^+$ such that $\int_0^\infty \mathcal{W}(t)dt = 1$

We then evaluate the vector $s(d)$ after the disaster as

$$M(d) = \sum_{k=1}^{K+1} \frac{M(G_k) - M(G_1)}{1 - M(G_1)} \int_{t_k}^{t_{k+1}} \mathcal{W}(t)dt, \quad (8)$$

where $M(G_k)$ is the value of the metric (in our case WATTR) on the graph G_k, $t_1 := 0$ and $t_{K+2} := \infty$. The value $\frac{M(G_k) - M(G_1)}{1 - M(G_1)}$ measures the effect of the recovery operations in the local network and ranges from 0 (no effect) to 1 (full recovery). In case $A(d) = \emptyset$, i.e. the disaster does not affect the network, we define $M(d) = 1$.

3. RECOVERY STRATEGIES

3.1 Optimal Strategy

If we let $V = \{v_1, v_2, \ldots, v_{|V|}\}$, and describe the choice of nodes as a vector of binary values $\mathbf{x}$ such that $x_i = 1$ if and only if v_i is replaced, then an optimal strategy is the solution to the problem

$$\max M(d|\mathbf{x}) \tag{9}$$

$$\text{s.t.} \sum_{i=1}^{|V|} x_i \leq K \tag{10}$$

$$x_i = 0 \qquad \forall\, v_i \notin A(d) \tag{11}$$

$$x_i \in \{0,1\} \qquad \forall\, i \tag{12}$$

where $M(d|\mathbf{x})$ is the value of $M(d)$ given the choice $\mathbf{x}$ of nodes to replace.

THEOREM 1. *When using the WATTR as the evaluation metric, computing the optimal strategy is strongly NP-hard even for the 0 cost case. i.e., when repair time is not considered.*

PROOF. See appendix A $\quad\square$

As computing the optimal strategy is an NP-hard problem and there might only be a limited amount of resources available after a disaster due to the destruction and chaos, computing the optimal choice of nodes might take too much time. In addition, the choice of which nodes to replace has to be made as quickly as possible after a disaster, at which point the complete state of the network might not be known. As such, it might be preferable to make some quick decisions based on a simple rule of thumb instead.

These rule of thumbs, or simple strategies, might be suboptimal for the specific situation, but give good results in general, whatever state the network might be in. In the following section, we propose several simple strategies.

3.2 Simple Strategies

We use $R \subseteq A(d)$ to indicate the nodes that will be replaced.

The basic idea of these strategies is as follows. Choose some node-metric $\mathcal{M}$, then iteratively select nodes to replace with the highest value of $\mathcal{M}$:

1. $R \leftarrow \emptyset$

2. Let $\mathcal{B} \subseteq A(d)$ be all nodes $v \in A(d)$ such that v is at most 1 hop away from (i.e .directly connected to) at least one node in $V_L - A(d)$. That is, $\mathcal{B}$ is the intersection of the neighborhood of $V_L - A(d)$ and $A(d)$. We want to limit ourselves to only replacing nodes in $\mathcal{B}$, as otherwise we would replace nodes without connecting them to a connected component.

3. Pick a $v \in \mathcal{B} - R$ such that $\mathcal{M}(v) \geq \mathcal{M}(y) \forall y \in \mathcal{B} - R$.

4. $R \leftarrow R \cup \{v\}$

5. $\mathcal{B} \leftarrow \mathcal{B} \cup \{y \in A(d) | \{v, y\} \in E_L\}$

6. If $|R| < K$ and $|R| < |A(d)|$, repeat steps 3-6

We consider 4 node-selection strategies:

- Greedy, that is, pick the node that has the largest effect on M: $\mathcal{M}(v) := M(d|R \cup \{v\}) - M(d|R)$.

- Pick the node with the highest weight-to-cost ratio: $\mathcal{M}(v) := \frac{w(v)}{\text{cost}(v)}$

- Pick the node with the highest neighbors-to-cost ratio: $\mathcal{M}(v) := \frac{|y \in V_L | \{v,y\} \in E_L|}{\text{cost}(v)}$

- Pick a node randomly. This strategy might not perform very well, but is very easy to execute after a disaster.

If $M(d)$ can be computed in polynomial time, the node-metrics can also be computed in polynomial time. As such, the simple node-selection strategies are all of polynomial complexity.

4. ALGORITHM

Let M be the random value of the evaluation metric after one of the disasters in D randomly occurs. Given a (general) recovery strategy, we want to compute the distribution over all possible values of M. Then, by comparing these distributions and the comparative effort to implement each strategy, a general recovery strategy can be chosen by the network operator and other involved parties. As soon as a disaster actually occurs, this strategy can then be implemented immediately, thus wasting no time on deciding on how to best recover the network.

For the purpose of our evaluation algorithm, we consider each possible recovery strategy as a function $R : V \rightarrow V$ from the damaged nodes $A(d)$ to a choice of nodes to replace with emergency nodes. Our algorithm is given in figure 1. We start by computing the set of affected nodes (the outcome) $A(d)$ for each disaster. As the state vector s will be the same for each disaster affecting the same nodes, i.e.

$$A(d_1) = A(d_2) \Rightarrow s(d_1) = s(d_2) \,\forall d_1, d_2 \in D \tag{13}$$

we can compute these states, and M, for each possible outcome instead of for each possible disaster to reduce the computation time.

Next, we go over each possible set of affected nodes and compute the corresponding local network, choose the nodes to recover, create the final state vector s and compute the value of M.

Using these properties, we can easily compute $P(M = m)$ for each $m \in \mathbb{R}$ by taking the sum of the probabilities of all disasters/outcomes resulting in this value of M. Computing all possible outcomes requires us to iterate over each disaster and each node, which takes $\mathcal{O}(|D||V|)$ time (assuming we can determine if a node is in the disaster region in constant time). The process can be sped up by using an R-tree.

Creating the local network takes $\mathcal{O}(|V| + |E|)$ time. However, computing the weights of the local nodes takes more time, as we need to find the closest nodes in V_L of each node in V. This can be accomplished by doing $|V_L|$ breadth-first searches, and thus takes $\mathcal{O}(|V_L||V| + |V_L||E|)$ time.

The time it takes to compute the choice of nodes to recover depends on the strategy that is used. For example, the weight-to-cost ratio strategy takes $\mathcal{O}(|K||V_L| + |K||E_L|)$ time to compute $R(G_1)$.

Finally, assuming integrating the weight function can be accomplished in constant-time, and the metric used is the WATTR, computing M takes $\mathcal{O}(|K||V_L| + |K||E_L|)$ time.

Input: undirected graph $G = \{V, E\}$, disaster set D, Recovery strategy function $R : V \to V$
Output: $P(M = m) \, \forall m \in \mathbb{R}$
  $O \leftarrow \emptyset$
  for all $d \in D$ **do**
    Determine $A(d) \subseteq V$
    if $A(d) \in O$ **then**
      $P(A(d)) \leftarrow P(A(d)) + P(d)$
    else
      $P(A(d)) \leftarrow P(d)$
      $O \leftarrow O \cup \{A(d)\}$
    end if
  end for
  for all $o \in O$ **do**
    $G_1 \leftarrow G - o \; \{o \subseteq V\}$
    $V_L \leftarrow \{v \in V | \exists x \in o \; h(x, v) \leq 1\}$
    Compute $R(o)$
    Order $[v_1, v_2, \dots] = R(G_1)$ such that
    $\mathrm{cost}(v_1) \leq \mathrm{cost}(v_2) \leq \mathrm{cost}(v_3) \leq \cdots$
    $t_1 \leftarrow 0$
    for $i = 1$ **to** $i = |R(G_1)|$ **do**
      $G_{i+1} \leftarrow G_i + v_i \; \{\text{Where } \{V_i, E_i\} + v_i = \{V_i \cup \{v_i\}, E_i \cup$
      $\{(x, y) \in E | x, y \in V_i \cup \{v_i\}\}\}$
      $t_{i+1} \leftarrow \mathrm{cost}(v_i)$
    end for
    $s \leftarrow [(G_1, t_1), (G_2, t_2), \dots]$
    Compute $M(s)$
    $M(o) \leftarrow M(s)$
  end for
  $\forall m \in \mathbb{R} \; P(M = m) = \sum\limits_{o \in O | M(o) = m} P(o)$

Figure 1: Recovery strategy evaluation algorithm

Thus, the time complexity of the algorithm is

$$\mathcal{O}(|D||V|^2 + |D||V||E| + |D|F(|V|, |E|, |K|)) \qquad (14)$$

where $F(G)$ is the time-complexity of the strategy.

5. EXPERIMENTS

We apply the framework to two U.S. topologies from the Topology Zoo [6]: Kentucky Datalink and ITC Deltacom. We ignore all nodes without any geographical coordinates.

ITC Deltacom consists of 101 nodes connected by 151 links, while Kentucky Datalink consists of 726 nodes connected by 822 links. Both networks are concentrated in the eastern half of the United States.

For each node v of these networks, we set $p(v)$ to the population of the county containing this node, based on the 2010 US Census [2].

The replacement costs $\mathrm{cost}(v)$ of each node are set randomly to a value between 6 hours and 120 hours (5 days). We use a weight function that decreases linearly to 0 at $t = 120$ hours, and is constant from then on. After 5 days the emergency recovery operations should be over, and repair operations should be in full swing.

As a use case, we consider a scenario where the network operator knows a hurricane will make landfall in a few days, but not the exact path it will take. Thus, his goal will be to decide on both a strategy and the number of emergency nodes to prepare. We generate a disaster set based on the *5 AM EDT THU AUG 25 2005* hurricane Katrina track

	K=1	K=2	K=3	K=6	K=10
Optimal	0.067	0.098	0.127	-	-
Greedy	0.067	0.095	0.122	0.180	0.215
Weight/Cost	0.022	0.043	0.063	0.134	0.191
Neighbors/Cost	0.035	0.064	0.083	0.128	0.189
Random	0.006	0.013	0.020	0.041	0.082

(a) ITC Deltacom

	K=1	K=2	K=3	K=6	K=10
Optimal	0.076	0.101	0.123	-	-
Greedy	0.076	0.099	0.118	0.149	0.161
Weight/Cost	0.064	0.088	0.105	0.138	0.153
Neighbors/Cost	0.063	0.082	0.096	0.130	0.151
Random	0.030	0.038	0.046	0.074	0.101

(b) Kentucky Datalink

Table 1: Expected value of M for different strategies, and different number K of temporary number of nodes, after hurricane Katrina, based on the *5 AM EDT THU AUG 25 2005* hurricane Katrina track prediction.

prediction of the National Hurricane Center (NHC) [3].

To predict potential storm surge flooding, and to assess the probability of wind surface probabilities, the NHC performs Monte Carlo simulations based on the predicted hurricane track and historical errors in their predictions. We propose using these Monte Carlo simulations as representative disaster set. As we do not have access to these simulations, and to demonstrate our approach, we use a simpler hurricane model, based on the NHC Track Forecast Cone. The "Tropical Cyclone Track Forecast Cone" shows the probable path of the center of a tropical cyclone. The cone is formed by simply placing a circle around each predicted track position and connecting them. The size of each circle is set so that two-thirds of historical official forecast errors over a 5-year sample fall within the circle.

We assume the actual track positions (in 2D projected coordinates) are distributed around the predicted positions according to a bivariate Normal distribution. This distribution is composed of normal distributions for the horizontal and vertical positions, each with a standard deviation of $\sqrt{\left(\frac{r^2}{\ln(10000/1225)}\right)}$, where r is the radius of the corresponding circle, to ensure 65% of samples lie inside the cone.

We can randomly sample hurricane tracks for our own Monte Carlo approach by sampling the track positions and then connecting them with a straight line segment. This only leaves us with the problem of computing the disaster region based on a hurricane track. The strike circle of a hurricane, based on the typical extent of hurricane force winds, is a circle with diameter 231.5 km, centered 23.15 km to the right of the hurricane center (based on its motion) [1]. In our approach we take this circle as the disaster region. Because the hurricane moves through the network area, the complete disaster region of each sampled track takes the form of a union of hippodromes.

Thus the complete approach to generating D is as follows:

1. Sample N sets of track positions.

2. For each track: compute the resulting disaster region.

3. Set all occurrence probabilities to $\frac{1}{N}$.

The potential hurricane realizations affect between 5 and 38 nodes of the ITC Deltacom network, and between 0 and 89 nodes of the Kentucky Datalink network, depending on their track through the network. On average, around 16 ITC Deltacom nodes and around 18 Kentucky Datalink nodes fail.

Table 1 shows the expected values of M utilizing each strategy for a number of different values of K. Due to its high computation cost, we did not compute the expected values of the optimal strategy for $K > 3$. The randomized node selection was evaluated by taking the average of 20 random recovery choices for each possible disaster outcome.

Selecting nodes at random performs very badly compared to the other strategies, especially on the ITC Deltacom topology. This shows how much of a difference it can make to recover nodes according to a suitable strategy.

In this use case, and for these topologies, the greedy strategy performs very close to optimal (at least for $K \leq 3$). As this strategy has polynomial complexity, it seems like a suitable choice.

Acknowledgments

6. REFERENCES

[1] Glossary of nhc terms. www.nhc.noaa.gov/aboutgloss.shtml. Accessed: 07-06-2017.

[2] Tiger/line with selected demographic and economic data. https://www.census.gov/geo/maps-data/data/tiger-data.html.

[3] Tropical storm katrina discussion number 7. https://www.nhc.noaa.gov/archive/2005/dis/al122005.discus.007.shtml?, 2005.

[4] N. Bartolini, S. Ciavarella, T. F. L. Porta, and S. Silvestri. Network Recovery After Massive Failures. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 97–108. IEEE, jun 2016.

[5] S. Ciavarella, N. Bartolini, H. Khamfroush, and T. L. Porta. Progressive damage assessment and network recovery after massive failures. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, Atlanta, GA, USA, may 2017. IEEE.

[6] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo. *Selected Areas in Communications, IEEE Journal on*, 29(9):1765 –1775, october 2011.

[7] J. Oostenbrink and F. Kuipers. Computing the Impact of Disasters on Networks. *ACM SIGMETRICS Performance Evaluation Review*, 45(2):107–110, oct 2017.

[8] T. Sakano, Z. M. Fadlullah, T. Ngo, H. Nishiyama, M. Nakazawa, F. Adachi, N. Kato, A. Takahara, T. Kumagai, H. Kasahara, and S. Kurihara. Disaster-resilient networking: a new vision based on movable and deployable resource units. *IEEE Network*, 27(4):40–46, July 2013.

[9] Y. Shibata, N. Uchida, N. Shiratori, and Others. A Resilient Network for Large Scale Disasters from Experience Based on the Great East Japan Earthquake. *Economic Alternatives*, (1):149–159, 2017.

[10] J. Wang, C. Qiao, and H. Yu. On progressive network recovery after a major disruption. In *2011 Proceedings IEEE INFOCOM*, volume 1, pages 1925–1933. IEEE, apr 2011.

APPENDIX

A. NP-HARDNESS OF THE OPTIMAL STRATEGY

Our proof is inspired by the proof of theorem 1 in [10].

We prove theorem 1 by giving a reduction from the well-known NP-complete SET COVER problem to the decision version of the optimization problem (with costs 0).

Note that the weight function $\mathcal{W}$ is irrelevant if all replacement costs are 0, thus, we will not include further mentions of the weight function in the proof.

The SET COVER problem can be described as follows: given a set $\mathcal{U} = \{u_1, u_{2,3}, \ldots, u_n\}$, a family $\mathcal{S} = \{S_1, S_2, S_3, \ldots, S_m\}$ of subsets of $\mathcal{U}$ s.t. $\cup_{i=1}^m S_i = \mathcal{U}$ and an integer $k \leq m$, is there a cover $C \subseteq \mathcal{S}$ such that $\cup_{c \in C} c = \mathcal{U}$ and $|C| \leq k$?

Given an instance of the SET COVER problem, we construct a (local) graph with nodes $V_L = \{b\} \cup \mathcal{U} \cup \mathcal{S}$. That is, V_L consists of a (base) node b, a node for each element in $\mathcal{U}$ and a node for each set in $\mathcal{S}$.

We directly connect b to all nodes in $\mathcal{S}$. In addition, for all nodes $S_i \in \mathcal{S}$ we add the links $\{\{S_i, u_j\}|u_j \in S_i\}$ to E_L. More formally, $E_L = (\{b\} \times \mathcal{S}) \cup \{\{S_i, u_j\} \in \mathcal{S} \times \mathcal{U}|u_j \in S_i\}$.

The weight of all nodes in $\mathcal{S}$ is set to 0, and the weight of all other nodes to 1. We let $A(d) = \mathcal{S}$, i.e. a node S_i is in the disaster region of the disaster iff $S_i \in \mathcal{S}$.

Note that this is a valid local selection of nodes and links, as all nodes in V_L are within 1 hop of the failed nodes.

Now, let $K = k$, the decision problem will be to determine if there exists a choice of at most K nodes of $A(d)$ to be replaced such that $M(d) = \text{WATTR}(G_K)$ will be greater or equal than 1.

Suppose there is a solution to the problem instance of SET COVER. That is, there exists a $C \subseteq \mathcal{S}$ such that $\cup_{c \in C} c = \mathcal{U}$ and $|C| \leq k$. By replacing all corresponding nodes $S_i \in C$, all nodes with a weight greater than zero will be connected to each other (through b). Thus, C is also a solution to the optimal strategy instance.

Conversely, suppose there is a solution to the optimal-strategy instance. That is, we have a set C of at most K nodes in $A(d)$, such that when these nodes are replaced, the WATTR of the local network will be 1. So every node $U_i \in \mathcal{U}$ must be connected to b through at least 1 node $S_j \in C$. That is, $\forall u_i \in \mathcal{U} \; \exists S_j \in C$ s.t. $u_i \in S_j$. Or alternatively, $\cup_{c \in C} c = \mathcal{U}$. So C is also a solution to the SET COVER instance.

We have provided a (polynomial) reduction from the strongly NP-complete SET COVER problem to the decision variant of the optimal-strategy problem with costs 0. As a result, we can conclude that the optimal strategy problem for the 0 cost case is strongly NP-hard.