Improving Sensor Data Delivery During Disaster Scenarios with Resilient Overlay Networks

Kyle E. Benson and Nalini Venkatasubramanian
Donald Bren School of Information and Computer Sciences
University of California, Irvine
Email: kebenson@ics.uci.edu, nalini@ics.uci.edu

***Abstract*—In this paper, we consider many-to-one communication, in particular Internet-connected sensors and their relation to disaster response. We explore the application of resilient overlay networks to aid these devices, or individuals if we consider participatory sensing, in quickly and effectively routing around geographically correlated failures in the underlying network infrastructure, as would occur during a large-scale natural disaster. We develop a formal model of this system, a heuristic for choosing overlay paths without relying on any knowledge of the underlying network infrastructure, and show its merit through simulations using real Internet topologies.**

I. Introduction

With the increasing availability and decreasing cost of microelectromechanical systems (MEMS) sensors, several projects have begun exploring the use of these devices in Internet-connected distributed sensing efforts. The Quake-Catcher Network [1] and Community Seismic Network [2] utilize small inexpensive accelerometers attached to volunteers' computers to monitor seismic activity. When they detect abnormal ground motion, indicative of a possible seismic event, these hosts report to a central server that processes the information and determines if an earthquake has occurred. Other such devices, whose information could help understand regional impact of phenomena, include weather stations, pollution detectors, and geiger counters [3].

We believe that the domain of disaster response could benefit from the inputs of these community scale networked sensors. Such sensor networks could potentially detect these events' onset and warn possibly affected individuals to find shelter, as well as aid first responders through increased situational awareness. However, network failures can severely hamper these networks' ability to gather useful information in a timely manner, especially important for those aimed at monitoring fast-moving destructive physical phenomena such as earthquakes and floods. Such events often result in large-scale geographically correlated failures in addition to serious network congestion as individuals contact each other or request help, exacerbating failures or tying up channels entirely.

Many previous projects have explored resilience to failures in the Internet, although few have addressed large-scale geographically correlated failures. Most of these works aim to formally model failures and identify strategies for designing more reliable network infrastructure. For example, [4] studied regional failures by defining line segments that cut any intersecting links in the graph representative of the network topology under consideration. Some of the most devastatingly impacting link cuts possible in a particular network provider were identified and categorized in [5] to aid in planning more resilient networks. Both [6] and [7] discuss general challenges to networked systems and discuss the proposed *ResiliNets* framework, which aims to formalize the steps and strategies involved in designing and maintaining more robust networks.

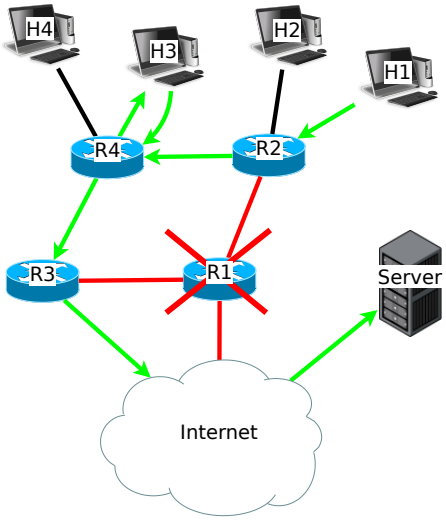

Fig. 1. An overlay routing example. Consider R1, R2, R3, and R4 are routers. R1 (crossed out) has failed and host H1's normal (shortest) path to the server is unavailable. Instead, it can route through nearby nodes in the order shown (R2, R4, H3, R4, R3, through the Internet and finally to the server). H3 serves as an intermediary hop so that H1 can target a different path along the underlying network, without the network knowing about it.

A. Resilient Overlay Networks

In this paper, we explore the creation and use of resilient overlay networks (RONs) to address large-scale disasters and the resultant failures that hamper sensor devices', as well as responders', ability to deliver data over the Internet backbone. Previous research [8], [9] has shown that these overlay networks can help route packets along alternative communication paths when the primary one is damaged, unavailable, or simply congested. This particularly helps while routing protocols still

978-1-6128-4937-9/13/$31.00 ©2013 IEEE

have not converged and established new end-to-end paths.

When a particular link becomes unavailable, whether due to a physical failure or congestion, the network's underlying routing protocols may take several minutes to find an alternative route. Several case studies have identified serious problems with routing over the Internet, such as [10] that discovered several paths hopping to other continents unnecessarily after a major earthquake in Taiwan. This paper also determined that BGP policies significantly reduced Internet resilience due to disallowing certain paths. Most visible failures were found to not exceed 5-15 minutes in [9] and BGP route update convergence was found to take up to 15 minutes after a fault in [11]. During this time, some end-to-end connections may be unavailable because certain paths are non-functional but others may exist that the routing infrastructure is not yet aware of.

In RONs, routers try to find an alternative path when the main one fails to deliver a packet, as shown in Figure 1. They attempt to make contact with another node in the overlay to see if that node is reachable and has a working path to the desired destination. If it does, then the traffic is routed through this intermediate node to the destination until a more direct path becomes available or less congested. Adding this level of intelligence to the routing infrastructure may incur large amounts of additional complexity and cost, but it can also be accomplished with simple end hosts in a peer-to-peer-like fashion. Deploying end hosts for the specific purpose of establishing a RON, or using those that are already part of a distributed sensing effort for this purpose as well, could possibly increase the reliability of a system without having to modify any of the routers in the underlying physical network.

II. Approach

To lend focus to our work, we explored this problem in the context of CSN [2]. In order to effectively identify and categorize earthquakes in a timely manner, the small messages sent by the seismic sensors, referred to as *picks*, must arrive at the server for analysis within a few seconds at most, especially if CSN is to be used as any sort of early warning system. One expects possible disruptions of the telecommunications infrastructure during a powerful seismic event and so this scenario seemed a perfect application for our technique.

In this section, we describe our system's formal model, our design goals, and approaches to achieving them. Later, we will describe how we extrapolated simulations from this design.

A. Model and Notation

Let $G = (V, E)$ be the graph defining the network under consideration, where V is the set of nodes representing routers and end hosts and E is the set of undirected edges representing physical links between two nodes. Let R be the set of regions under consideration, $f : V \to R$ map each node to the region it is located in, and $C_D \in C$ be the location of the disaster. In this paper, we consider each region as a city and so f assigns each $v \in V$ to the city whose center is closest to the location of v, which could be gleaned from GPS, IP address, or user-specified data. Note that although these approaches may not give perfectly accurate location information, the coarse granularity of f means that they should reasonably suffice for our purposes. Let $S \in V$ be the server (sink) to which each sensor node within R_D attempts contact with during the disaster. Therefore, if we let $O \subset V$ be the nodes chosen for the overlay network, then $O_D \subset O = \{o \in O \mid f(o) = R_D\}$ are the sensor nodes (RON clients) that report picks.

B. Failure Recovery

When some $o_1 \in O_D$ has sensor data to report to S, it first attempts a direct connection to S, but may detect a possible failure in the network's chosen path as evidenced by a timeout. In response, o_1 should try to connect via a working alternative path through the overlay. Let $P_1 = (e_1, e_2, ...e_n)$ be the sequence of edges along the path that the undelivered message would normally take. The message should then travel some path $P_2 = (e'_1, e'_2, ...e'_m), P_1 \neq P_2$ instead to reach its destination. In the case of an overlay, we have at least one o_i, which we generally refer to as o_2 when discussing one-overlay-hop connections, incident with some $e'_i, e'_{i+1} \in P_2$ to aid in routing the message around the failed links in P_1.

Note that $\exists C \subset P_1 \cup P_2$ where C is a cycle. Indeed, many previous works, such as [12], [13], identify cycles in a network a priori to quickly establish alternate routes. However, these approaches generally rely on complete knowledge of the underlying network structure, as well as the failed link(s) or node(s), because they targeted smaller internal networks. Considering Internet-scale networks invalidates this assumption as traceroute, the typical method of learning an external network's topology, provides somewhat unreliable data (due to i.e. dynamically adapting paths, administrators obfuscating internal nodes on a network, etc.) and so may not accurately reflect the routes or failure location(s). Therefore, we aim to identify heuristics for choosing overlay paths in this work.

While a timeout may not indicate a truly damaged path, the packet may have been lost due to congestion or a poor connection somewhere in the infrastructure. Therefore, it may benefit o_1, and others on the network as well, to try routing around this path as an alternative may exhibit less latency and a higher delivery ratio. This technique was proven quite effective in [9], which found that "overlay networks can typically route around 50% of failures."

In the case of CSN, o_1 could immediately send the pick to some $o_2 \in O, o_2 \neq o_1$ and request that it be forwarded because adding sensor data (less than 1KB) does not dramatically increase the connection information request packet's size. As per the finding in [14] that the majority of end-host pairs can establish paths with the same diversity in a single overlay hop as in multiple hops, we only consider a single hop at this time.

C. Overlay Construction

To facilitate routing sensor readings (picks in our CSN scenario) around network failures, we use end hosts with stable (wired) Internet connections and ample power supplies as O. We opt for this approach to avoid relying on Internet Service Providers (ISPs) adopting and deploying new technologies. It

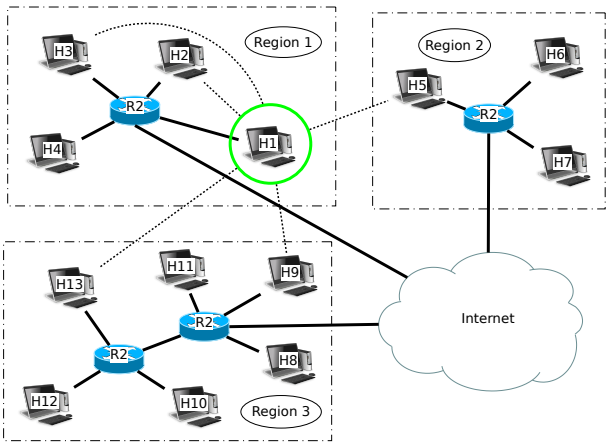

Fig. 2. A network overlay. The dashed edges incident with the circled node represent overlay connections. There are no direct physical connections between these nodes, but they can reach each other through their respective routers and the Internet with knowledge of each others' IP addresses.

also opens the possibility to include more sophisticated functionality, such as in-network processing of sensor readings, in future renditions without overburdening systems that are designed to route traffic at high rates.

Each $o_i \in O$ should scalably maintain information locally about a subset of the other $o_j \in O \mid f(o_i) = f(o_j)$. Due to the proximity of these nodes, they are more likely to maintain, or quickly recover, connectivity during failures and so can share the load of maintaining knowledge about the rest of the network, sending this information to others when necessary.

In addition to this local knowledge, each o_i must know about some $o_k \in O \mid f(o_i) \neq f(o_j)$ to establish overlay connections outside of the local area, like in Figure 2. When o_i contacts o_k, the latter may also return information about some other $o'_k \mid f(o'_k) = f(o_k)$ so as to provide an alternative in case o_k fails. This strategy resembles that used in the peer-to-peer system Pastry [15], except that it uses the physical locations of the nodes, f, to assign locations within the overlay.

In this manner, even if o_i does not know of, or cannot establish a connection with, any o_k, it could contact one of its neighbors $o'_i \mid f(o'_i) = f(o_i)$ that it does know in order to hopefully learn about an o_k, like how the shaded node in Figure 2 could query its neighbors for the locations of other nodes outside its area. Therefore, each $o \in O$ shares the load of storing each others' addresses while still providing a quick method for looking up an o_k outside of the local region.

In our current simulations, we adopt a simplistic approach of keeping full knowledge of O on each $o \in O$, which clearly would not scale well in a real deployment, as opposed to only maintaining a subset of O. As explained in [16], a scalable method that ensures connectivity of the overlay with high probability would be for each o to store information about $O(\log(|O|))$ other nodes. This paper discusses a technique for establishing peer-to-peer connections based on locality, but did not explicitly address geographically correlated failures. We did not address the specifics of bootstrapping and maintaining the overlay in this work and will do so in the future.

D. Route Selection

When requesting an overlay connection, o_1 should consider the location of o_2 and the path between the two. If o_1 knows $f(o_2)$ and $f(S)$ and has some idea of the spatial properties of underlying network, it could choose a more stable overlay path. One should note the possibility that $P_1 \cap P_2 \neq \phi$, that is, they share at least some edge. Choosing P_2 to minimize $|P_1 \cap P_2|$ would likely decrease the probability of some failed link or node being present in P_2. Kim takes this approach in [17], in which a node in the overlay considers the physical distance between the routers along a path between two nodes. Neighbors are chosen to have a lower path correlation than the other candidates, thereby improving the likelihood of viable alternative paths. This technique improves data dissemination reliability during disaster scenarios, but it assumes complete knowledge of the underlying network.

In this paper, we test a heuristic that only assumes knowledge of the nodes' physical locations. The *Orthogonal Distant Path Heuristic (ODP)*, depicted in Figure 3, maps each peer $o \in O$ to a point within a two-dimensional vector space, as determined by the latitude and longitude of $f(o)$. Let v_r, v_s, v_o be the vector locations of the reporting node o_1, the server S, and the chosen overlay peer o_2, respectively. Let A be the angle between the vectors $v_o - v_r$ and $v_s - v_o$. Let D be the minimum distance from v_o to the line spanning between v_r and v_s. When o_1 detects a failure along the path to S, it chooses the peer, o_2, closest to v_o such that A is as close to orthogonal as possible. Furthermore, o_2 is chosen such that D is as close to ideal as possible, where the ideal distance is that of an isosceles triangle with the vertices v_r, v_s, v_o, where v_o satisfies the orthogonality described above. We define the objective function of ODP as:

$$err(A)^2 + err(D)^2 \tag{1}$$

where $err(x)$ is the percent error of x from its ideal value. When choosing an overlay path, ODP picks o_2 to minimize this function. Figure 3 explains our rationale for this heuristic.

III. Evaluation Methodology

Due to security and privacy concerns, as well as the realistic issue of powerful earthquakes fortunately occurring relatively infrequently in the region covered by CSN, we opted to test our design in a simulation environment first. We used the ns-3 [18] network simulator because its open source nature allowed us to make extensive changes (described in the next section) to the underlying system to fit our purposes.

A. ns-3 Changes and Additions

To generate realistic Internet topologies, we used ns-3's RocketfuelTopologyReader model that builds network topologies from the trace files compiled in the Rocketfuel [19] project. This project mapped the nodes and links in several

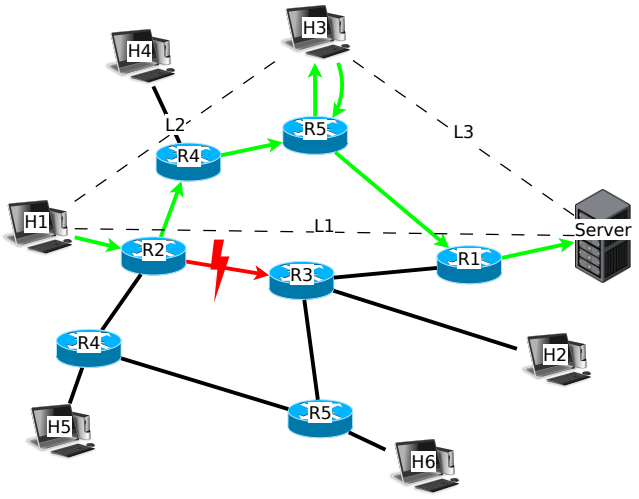

Fig. 3. When H1 tries to report data and finds its normal (shortest) path to the server disrupted, it chooses H3 as an overlay peer. Because the angle between lines L2 and L3 (A) is close to orthogonal and the minimum distance from H3 to L1 (D) is large, the packet is more likely to find an alternate route and not traverse portions of the normal path, which may be damaged or congested. While D could be maximized to increase the probability of finding a different path, this would actually decrease A and unnecessarily increase the latency.

Autonomous Systems (ASes), including some interconnections between them. The biggest change made to ns-3 was expanding upon this model, adding support for node locations for the purpose of defining f. To map city names to latitude and longitude coordinates, we used data from the GeoNames database [20]. ns-3 currently cannot combine multiple ASes in one simulation, but we intend to extend it in the future to include this feature.

We encountered scalability issues when simulating very large topologies (1000+ nodes) in ns-3 and so utilized the known workaround of instead establishing routes on-demand via NixVectorRouting and caching them [21]. We had to modify this module to follow down links, which is how we modeled failures in the network, as otherwise this method would automatically route around them.

To further improve the speed of our simulations, we also extended ns-3 to support running multiple simulations on the same collections of objects. This allowed the simulator to utilize more cached routes and not have to parse the topology files and build the network in between each different run.

In addition to the above changes, we added an Application model for RON clients and servers, including a packet Header for storing information about the chosen overlay path through which a packet should be routed. This enabled easy installation of simulated RON overlay software on O and easy manipulation of the parameters affecting their behavior for testing different system designs. Whenever some o contacts S, it replies with an acknowledgement (ACK) packet along the same overlay path used to reach it.

B. Simulation Design

To evalute the effectiveness of using RONs for improving Internet-connected sensors' delivery ratios during large-scale disasters, we chose a larger AS (Level 3, AS #3356) and a large city (New York City). Here we describe the formal structure of our simulation, including our failure model, and the parameters that we defined.

From the network topologies, we choose each $o \in O$ such that $\deg_G(o) = 1$. This lessens the possibility of choosing backbone routers as end hosts are going to be connected to stub routers within an AS, or at least to routers that link together several local area networks (LANs). We install a RON client application on each $o \in O$.

S is chosen at random from the nodes $\{s \in V \mid s \notin O \wedge f(s) \neq R_D\}$ and a RON server application is installed on it. We chose S as outside of the disaster region because we assume that multiple servers might be available and that O_D are programmed to report to a server outside of their region to prevent the data from being lost if S were to fail during a disaster. In the context of CSN, each $o' \in O_D$ reports to a cloud-hosted server that is outside of California in an attempt to ensure the data's availability if a severe enough earthquake were to partition the network.

To represent failures during a disaster, we randomly chose (with some probability) nodes and links within R_D to fail before starting the simulation. Formally, we chose $V_F = \{v \in V \mid f(v) = R_D\}$ and $E_F = \{e = (v_1, v_2) \in E \mid f(v_1) = R_D \vee f(v_2) = R_D\}$, where v_1 and v_2 are the nodes connected by e and with each node or link being chosen with probability p(fail) , where p(fail) is a parameter defined in the simulator and passed in at run-time. We tested nine different p(fail) values: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. After constructing routing tables, each $v \in V_F$ and $e \in E_F$ are turned off in the simulation by setting each network device on v, or connected to e, to down and not forward traffic.

In addition to the failure probability, we defined several other parameters to change our systems behavior. First, we specify the input Rocketfuel trace files for the particular AS that we are studying. We also choose a R_D from the R listed in these files for the disaster to occur in. Furthermore, the clients' timeout value and number of retries are given to control the RON. When a connection attempt times out the first time, $o' \in O_D$ will attempt to contact some $o_k \in O$ if its retry parameter is at least 1, where o_k is chosen based on the current heuristic under study. If this connection times out as well, it will try a different node and so on until it either successfully reaches S (receives an ACK over the overlay) or fails to do so a number of times equal to this retry parameter.

For the timeout value we used 500ms because the round-trip-time across the continental United States is typically 200-300ms and utilizing an overlay node would increase this further. Additionally, our model did not incorporate realistic network delays from queueing, congestion, and channel errors so the round-trip time to the server was typically much less than 100ms, making a 0.5 second timeout quite conservative for our simulation. Even if this value resulted in wasted retries, one must remember that our use case demands fast delivery times but does not cause much congestion due to small message sizes. Therefore, making additional connection attempts

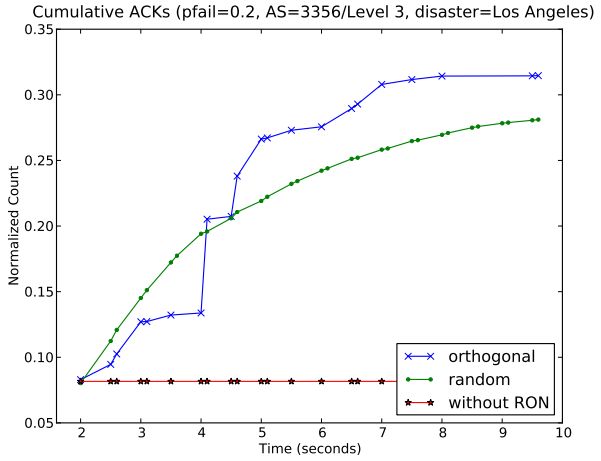

Fig. 4. Cumulative ACKs over time for failure probability of 0.2 in Los Angeles, CA and AS 3356 (Level 3). Note that the ODP heuristic establishes a few more connections later on than the random heuristic, but a few less initially. The y-axis is normalized (by dividing by the number of actively reporting nodes) so as not to bias against higher p(fail) values in which many sensor nodes fail and cannot contact the server regardless of network availability.

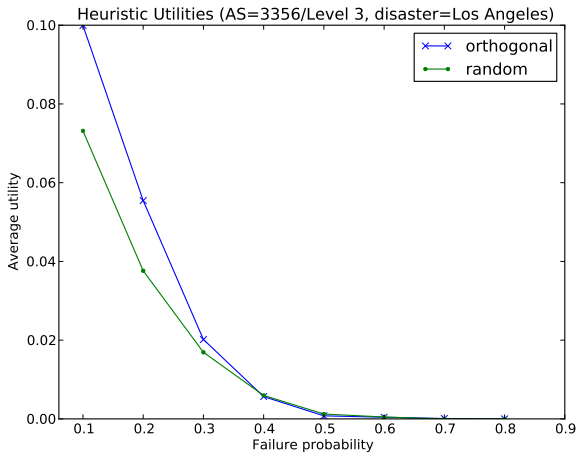

Fig. 5. The computed average utility metric of the two heuristics for various p(fail) values. Note the convergence of the curves on each other as p(fail) increases and fewer redundant paths are left undisrupted.

proactively may decrease response time without negatively impacting the network significantly.

We set the number of retries to 20, resulting in a simulation length of 10 seconds. We chose this simulation length because we assume that the routers and switches within an AS will likely be able to update their routing tables within 10-15 seconds. Furthermore, the sensors should upload their data quickly to S for time-sensitive processing and response. We will use longer simulation times when studying larger overlays spread across several ASes due to route updates taking longer to propagate across higher diameter graphs.

IV. Experimental Results

For this paper, we compared two different heuristics: a baseline random heuristic and the aforementioned ODP (Section II-D). In the baseline random heuristic, overlay peers are chosen randomly from all possible choices of $o_2 \in O$. A more effective heuristic should perform significantly better than this one, which uses absolutely no knowledge of the underlying network or other peers' physical locations. To study different scenarios, we varied the following parameters: failure probability (0.1,0.2,...0.9) and disaster location (New York, NY and Los Angeles, CA). Each set of parameters was simulated 200 times and the results of these averaged to lessen the impact of edge cases and better represent the expected values. The results show considerable promise for our application.

Figure 4 shows the two studied heuristics' convergence towards the percentage of failure recoveries that they are capable of making. It also shows the number of ACKs in the non-RON case to visualize the improvement over traditional routing. The slope of the curves decreases over time because sensors will stop attempting contact with S once they receive an ACK. The ODP heuristic appears to make more recoveries over time than the baseline random heuristic, but less in the initial stages, indicating that some fine-tuning of ODP may greatly improve its performance. If the simulations continued indefinitely, we would likely see both curves converge on the same value, but messages received that long after the event may not be as useful for applications such as early warning.

To evaluate each heuristic, we defined a *utility metric* for a node that uploads its data (receives an ACK) at time t (seconds) as:

$$u(t) = \begin{cases} 0, & \text{if ACK never received} \\ \frac{1}{t}, & \text{if ACK received at time } t(seconds) \end{cases} \quad (2)$$

This metric assigns a higher utility to earlier connections because of the time-sensitive nature of the data. If the server receives this information earlier, it can process it and act sooner, whereas the message may not be of much use later on, or it could have been delivered without an overlay if the routing infrastructure has updated its default routes. For each simulation run, we average this metric over all nodes in the simulation to get its expected value.

Figure 5 shows the effect of varying p(fail) on the utility of the two heuristics. Once again, ODP appears slightly more effective than the baseline random heuristic, but only up to p(fail) of approximately 0.2. Not surprisingly, the utilities of both heuristics decrease as p(fail) increases and leads to less opportunities for establishing alternate routes.

Interestingly, this figure shows that a higher failure probability is not necessarily proportional to the computed utility metric. Higher p(fail) values increase the chance of long-haul links necessary for connecting with other cities being disrupted, which means that fewer sensors can contact the server via RON peers. During our tests, we found that most topologies and locations would establish very few, if any, connections

for much higher p(fail) (0.7-0.9). Obviously, RONs can only help in so far as physical paths through the network still exist, but they can certainly help discover them quickly, especially if suitable alternatives are decided ahead of time.

To empirically compare our heuristics, we used a two-sample t-test to compare the mean utilities for the 200 samples drawn from the distributions created by the two heuristics. We consider each combination of p(fail) values and disaster locations as separate populations because these choices greatly affect the mean and variance of the delivery ratio and time at which ACKs are first received. We hypothesized that the ODP heuristic would improve the utility metric because it would be more likely to find a viable alternative path sooner than the baseline random heuristic. We therefore let the null hypothesis, H_0, be that the mean utility metrics for both distributions are the same and the alternative hypothesis, H_a, be that these means are indeed different. The results of the tests indicate a statistically significant difference for p(fail) values of 0.1 and 0.2 in both disaster locations. Therefore, we can reject H_0 for these cases and claim with high confidence that ODP improves the expected utility in these scenarios. However, the lack of significant improvement for higher p(fail) values necessitates further testing and refinement of the objective function (perhaps by relaxing the rigidity of the location requirements or assigning more weight to one than the other).

V. Conclusion and Future Work

In this paper, we explored the application of resilient overlay networks to the domain of Internet-connected sensing during regional failures due to large disasters. We developed a formal model for organizing the overlay nodes and choosing alternate paths. We presented the results of our initial simulations and ODP heuristic that prove the utility of this approach.

We plan to continue this project to further refine our techniques and explore alternative approaches. In particular, we are studying the process of building overlay paths a priori, using more advanced heuristics than the ones proposed here (possibly with partial knowledge of the underlying network structure), to improve recovery time during a disaster. We also plan to develop heuristics that make overlay peer choices on-line, using knowledge of perceived failure locations and information exchanged with other nodes during an event to more quickly reestablish failed connections. We are also working towards addressing AS interconnections and the resilience issues inherent with BGP route updates as we continue this project.

In the future, we plan to investigate the possibility of including wireless and mobile devices in the overlay to establish a multi-network environment. This could lead to delay-tolerant protocol designs and would likely increase the effectiveness of the overlay during disaster scenarios, especially when modeling continuing and moving failures.

Acknowledgment

This work was supported by National Science Foundation award nos. CNS 1143705 and CNS 0958520. The authors also thank Mani Chandy and Julian Bunn at Caltech for their discussions related to CSN and this work.

References

[1] E. Cochran, J. Lawrence, C. Christensen, and A. Chung, "A novel strong-motion seismic network for community participation in earthquake monitoring," *Instrumentation Measurement Magazine, IEEE*, vol. 12, no. 6, pp. 8 –15, Dec 2009.

[2] (2012, Jul) Community Seismic Network. http://www.communityseismicnetwork.org/.

[3] (2012, Aug) Pervasive computing for disaster response. http://www.cacr.caltech.edu/projects/PerDis/.

[4] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[5] A. F. Hansen, A. Kvalbein, T. Čičić, and S. Gjessing, "Resilient routing layers for network disaster planning," in *Proceedings of the 4th international conference on Networking - Volume Part II*, ser. ICN'05. Springer-Verlag, 2005, pp. 1097–1105.

[6] J. Sterbenz, E. C andetinkaya, M. Hameed, A. Jabbar, and J. Rohrer, "Modelling and analysis of network resilience," in *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, jan. 2011, pp. 1 –10.

[7] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, no. 8, pp. 1245–1265, jun 2010.

[8] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of the eighteenth ACM symposium on Operating systems principles*, ser. SOSP '01. New York, NY, USA: ACM, 2001, pp. 131–145.

[9] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek, "Measuring the effects of internet path faults on reactive routing," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 126–137, Jun 2003.

[10] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin, "Internet routing resilience to failures: analysis and implications," in *Proceedings of the 2007 ACM CoNEXT conference*, ser. CoNEXT '07. New York, NY, USA: ACM, 2007, pp. 25:1–25:12.

[11] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 4, pp. 175–187, Aug 2000.

[12] K. Nakayama, N. Shinomiya, and H. Watanabe, "An autonomous distributed control method for link failure based on tie-set graph theory," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, no. 11, pp. 2727 –2737, nov. 2012.

[13] D. Stamatelakis and W. Grover, "Theoretical underpinnings for the efficiency of restorable networks using preconfigured cycles (p-cycles)," *Communications, IEEE Transactions on*, vol. 48, no. 8, pp. 1262 –1265, aug 2000.

[14] J. Han, D. Watson, and F. Jahanian, "Topology aware overlay networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, march 2005, pp. 2554 – 2565 vol. 4.

[15] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, ser. Middleware '01. London, UK, UK: Springer-Verlag, 2001, pp. 329–350.

[16] L. Massoulie, A.-M. Kermarrec, and A. Ganesh, "Network awareness and failure resilience in self-organizing overlay networks," in *Reliable Distributed Systems, 2003. Proceedings. 22nd International Symposium on*, oct. 2003, pp. 47 – 55.

[17] K. Kim and N. Venkatasubramanian, in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*.

[18] (2012, Jun) ns-3. http://www.nsnam.org/.

[19] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 1, pp. 2 – 16, Feb. 2004.

[20] (2012, Dec) Geonames. http://www.geonames.org/.

[21] (2010, Apr) Bug 521 - Ipv4 global routing inefficient. https://www.nsnam.org/bugzilla/show_bug.cgi?id=521.