A Disruption-Tolerant Architecture for Secure and Efficient Disaster Response Communications

Kevin Fall[1] Gianluca Iannaccone[1] Jayanthkumar Kannan[2] Fernando Silveira[3] Nina Taft[1]
[1]Intel Labs, Berkeley [2]University of California, Berkeley [3]UPMC Paris Universitas

ABSTRACT

We consider the problem of providing situational awareness when citizens in a disaster are willing to contribute their own devices, such as laptops and smart phones, to gather data (text, images, audio or video) and to help forward data gathered by others. A situational awareness service processes all received data and creates annotated maps to visualize a disaster site (e.g., the status of the disaster, such as fires or floods, the location of people, food, or water). We discuss the challenges imposed on such an application when 1) the communications infrastructure in the disaster area can only provide intermittent connectivity, 2) anxious victims generate large amounts of redundant content congesting the network, and 3) the sharing of personal devices creates security and privacy threats. We present an architecture that addresses the requirements to support such a service.

Keywords

Situational Awareness, Delay / Disruption Tolerant Networking, Information Sharing, Privacy

INTRODUCTION

During a disaster, *situational awareness* (SA) is one of the most important needs to minimize injury, loss of life and property damage. It includes information about the environment (e.g., location of fires, flooded areas, resources) as well as the status of individuals affected by the disaster (e.g., health monitoring, distress calls). Situational awareness is important for victims as well as for first responders. Citizens inside the disaster area need this information, not only for their own survival, but because they often participate in the emergency response efforts themselves (Palen and Hiltz, 2007). First responders need comprehensive, reliable information in order to make well-informed decisions. It is thus critically important to be able to gather data from inside a disaster area and deliver it to a service where it can be processed, validated, and effectively shared.

People's everyday devices, such as smart phones and laptops, can be used to gather and access critical information in a variety of formats, including text, photos, audio and video. Those affected by the disaster may use these devices not only to produce information that can be incorporated into a shared situation awareness application, but also to validate the authenticity of information they use. An example of an SA application is one that processes and assembles gathered data to create a *situational awareness map* in order to provide a global view of the disaster to victims, first responders and others (e.g., those outside the affected areas). The web application Ushahidi (http://Ushahidi.com) is one such example. It allows individuals to generate reports with text and photos that are aggregated and displayed on a Google map or as a list of text reports.

The crisis-assistance social-networking applications that we have seen (such as Ushahidi, (Lui et al., 2008), (Hughes and Palen, 2009) and Google's PeopleFinder) are "first generation" web applications that have critical limitations. First, they assume that the victims have continuous access to the Internet. Second, they assume that this access will be of sufficient quality to handle whatever data load is required. Finally, they do not verify the integrity of the data they collect using strong methods (e.g., cryptography). Unfortunately, these limitations may become manifest as the result of an earthquake, flood or fire that leads to communication infrastructure disruption. Furthermore, it has been observed (Hughes et al., 2008), that in order to cope with the stress and anxiety of a disaster, victims tend to communicate extensively, thereby exacerbating the demands placed upon the remaining communication resources. Coupled with the lack of strong authentication, communication capacity available during a disaster may be tasked with carrying redundant, unverified data that could degrade or prohibit the effective operation of SA applications. Our goal is to improve the underlying network architecture and security model to better support a broad variety of SA services that need to operate during crises. We first discuss the three limitations in more detail followed by the communication architecture we are pursuing.

Reviewing Statement: This paper represents work in progress, an issue for discussion, a case study, best practice or other matters of interest and has been reviewed for clarity, relevance and significance.

CHARACTERISTICS AND CHALLENGES OF COMMUNICATING DURING DISASTERS

Recent studies have shown that, when citizens in a disaster area have Internet connectivity, they often contribute to SA by sharing photos (Lui et al., 2008), micro-blogging (Hughes and Palen, 2009) and by using other social networking sites. It has also been shown that the information shared using social networking tools is useful in collective problem solving, e.g., figuring out lists of victims during an incident (Vieweg et al., 2008). Naturally, utilizing Internet-based services such as these requires access to the Internet of sufficient quality to support operation of common client software such as web browsers.

In some disasters, Internet connectivity may be only intermittently available or otherwise limited. In these situations, it is much more difficult to collect and distribute SA information. Connectivity can be limited when civilian telephony and Internet infrastructures, e.g., cell phone towers, telephone networks, and wireless access points, are also affected by the disaster. Because connectivity may be in short supply, its efficient use is important. At present, the Internet's primary TCP/IP protocol suite and accompanying web applications provide timely delivery and efficient resource usage only when connectivity is plentiful. New methods that support controlled sharing of network bandwidth while tolerating periods of disruption and delay are needed.

In the last few years, the networking research community has proposed specific protocols that can cope with the challenges of intermittent and degraded connectivity. *Delay and Disruption Tolerant Networking* (DTN) (Fall, 2003) suggests that *store-carry-forward* protocols (where data can be physically carried if necessary) enable communication even in extreme cases of impaired connectivity. For instance, if only one person in a disaster has a phone with an Internet connection (e.g., via satellite), others may use this person's device as a relay to send their messages. More generally, if a neighborhood currently lacks Internet access entirely, data can be collected by one or more devices and physically carried towards a point where a communication infrastructure is available (Chaintreau et al., 2006; Seth et al., 2006).

The existing Internet infrastructure has fault tolerant mechanisms built in, such as secondary links that are activated within milliseconds when a primary link fails. These rely on employing backup infrastructure when primary infrastructure is not available. During a disaster, we cannot assume backup infrastructure will be functional. Consequently, we require a fault and disconnection tolerant network that goes beyond the capabilities of the standard Internet protocols. DTN can tolerate delays of much longer time scales, such as hours or even days, by using persistent storage within the network. In this paper, we propose how an SA service, intended for disaster response communication, should be based upon DTN. Applications utilizing DTN can provide an ad hoc delay tolerant network formed by the personal devices of those within an affected area. Although protocol specifications for DTN have existed for some time, few applications have fully exploited DTNs features, and few operational DTNs have been built and deployed (http://www.dtnrg.org).

In considering how to architect an SA application and its supporting network to be robust to disasters, we first consider the likely behavior of its users. As mentioned above, people tend to communicate a great deal by any means possible during a disaster due to anxiety (Hughes et al., 2008). When victims send "I'm OK" messages, or a photo of their burning house repeatedly to everyone they know, there is an enormous amount of redundant content created and transmitted through the network. This natural behavior aggravates the problem of limited connectivity, and leads to congestion and inefficient use of shared communication, storage, and power resources. Furthermore, congestion of this form may frustrate the use of an SA service by first responders that must make timely decisions based upon the best data available at any moment in time. A requirement for our architecture, then, is to ensure that useful SA information can travel successfully to and from servers implementing SA services even in the face of congestion and limited connectivity. This implies the need for both sophisticated prioritization mechanisms (such as meta data labeling and filtering), and mechanisms to identify and reduce redundant content. DTN implementations today do not support these capabilities and we thus propose to augment DTN's basic message forwarding functionality to handle application-specific requirements such as these. Forwarding based on content is a new paradigm that stands in contrast to the Internet of today where resources are shared by considering the delivery of all bits to be equally valuable.

Even if non-redundant SA information can be made to flow efficiently in challenging environments, critical decision-making requires the underlying information to be trustworthy. This is a special concern when personal devices are used to store and forward the data of others in an ad hoc network. In this context, by "trustworthy" we mean there is an ability to verify the origin and integrity of SA data, and to provide privacy and access control if requested. These are standard challenges in networking, but our most common solutions (e.g., TCP/IP with SSL or IPSec) require re-consideration for use in networks with intermittent and degraded connectivity.

ARCHITECTURE OVERVIEW

Our proposed SA application architecture comprises a cluster of infrastructure-supported back-end servers maintained by some trusted entity (e.g., first responders), and a collection of portable nodes that may suffer intermittent or limited connectivity. The SA service uses a combination of DTN and Internet protocols (and DTN protocols can run atop the Internet if so configured). The goal of the system is to have trustworthy data flow between the portable nodes and back-end servers, even in the absence of a public communication infrastructure (e.g., cell towers, DSL lines, CATV data service). To cope with the lack of infrastructure, DTN performs store-carry-forward communications as necessary. The potential lack of "always-on" connectivity limits the demands applications can make on the network. For instance, we cannot guarantee that a sub-200 millisecond latency VoIP conversation can be sustained using a network fabric that is not always connected. However, for applications where end-to-end interactions do not have to be instantaneous, DTN protocols are not only convenient, but they may be the only choice to allow data to reach their destination. Furthermore, DTN protocols *can* deliver messages with low-latency when network conditions permit.

A DTN implementation *schedules* the order and time messages are forwarded. In most cases, this is performing in a simple *first-come-first-served* (FCFS) fashion. For the disaster scenario we are presently addressing, we add scheduling mediated by origin authentication and prioritization and content of the messages being forwarded. By allowing content to play a role in scheduling decisions, data redundancy can be reduced, leading to a number of potential benefits: more trustworthy data, less congestion, and quicker access to the most important information.

Node Classes and Architectural Functions

Broadly speaking, we consider the network supporting our SA application to comprise both back-end servers and two classes of portable nodes: *producer/consumer devices* (PCDs) used to collect and access data and *field servers* that are set up to act as ad hoc routers and caches for data. Back-end servers are typical servers found in an Internet data center. Field servers are computers that can communicate over the air with nearby nodes and have reasonable storage, power and communication resources. They are expected to be deployed with power supplies of moderate endurance (e.g., on vehicles with generators). Field servers may belong to either citizens or first-responders and are expected to be present in modest number within disaster-affected areas. They are configured to contribute (some of) their resources (bandwidth, battery, and memory) to the overall application. The fraction of resources dedicated to the function of relaying data for other systems is configurable by the device owner, as field servers may be performing other related or unrelated tasks.

PCDs perform a number of user-facing functions. More specifically, they: (1) generate and sign SA information in the form of text, audio or video, (2) sign and attach meta-data to the SA information, (3) access, verify, and display SA information contributed by other nodes, and (4) query for specific types of information. The contributions of a PCD are expected to reach the back-end servers eventually, regardless of any intermittent connectivity problems. PCDs obtain SA information from other nodes, either back-end servers or field servers.

Our architecture utilizes several different functions spread across the various node classes (PCDs, field servers and back-end servers). Generally speaking, each node is of a single class that corresponds to some set of functions (called a *role*), although some nodes may assume more than one at once. Figure 1 shows the mapping between node type and architectural functions. The functions in the "Delay-Tolerant Networking Protocols" box are available on all devices.

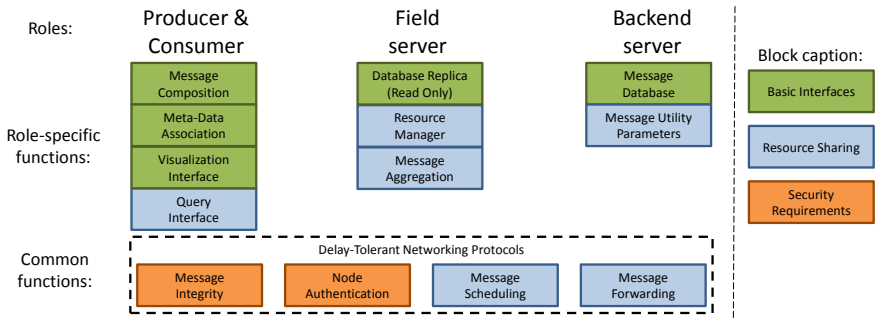

Figure 1. Functions associated with the various node classes.

A producer initially reports information through a *message composition* interface. This produces a message bundle containing an aggregate of text, images, audio, and video. The bundle is next annotated with *meta-data* that provides different types of context information, along with a digital signature. We consider three basic types

of meta-data: space coordinates (e.g., obtained from a GPS receiver), timestamps, and tags that describe the content (e.g., "fire", "S.O.S. request"). Some tags can be generated automatically by applying simple classifiers to the data. For instance, the interface could check for the presence of meaningful keywords such as "fire" or "flood" in text and audio, or recognize specific objects such as victims in photos. The interface also allows a user to manually input meta-data. For example, a user might want his/her message to refer to a location other than his/her current GPS coordinates, the timestamp to refer to an event in the past, or to include tags that are not automatically included by the automated content classifiers.

The presence of different types of meta-data allows us to think in terms of a *meta-data space*, where coordinates are combinations of space, time, and tag values. From this point of view, a message with its associated meta-data can be considered as "covering" parts of the meta-data space. This abstraction is important because it allows us to make decisions about priorities in message forwarding based on characteristics of a message's content and its ability to cover more or less of the meta-data space. For example, a photo of a geographic area for which no data has yet been received could be considered more valuable (and thus worthy of higher delivery priority, if authentic) than a message containing a photo of locations for which data has already been received.

Messages that arrive at the cluster are stored in a *message database*. The next generation of SA services will process, merge and mine the data to extract important information, and present SA information via annotated maps. Consumers should also be able to query the database for specific information. Because data is both produced and consumed by a community of users, this SA service can be viewed as a social networking application. In order to provide SA to victims without continuous connectivity, we believe the database should be replicated and cached on field servers whenever possible. If the server cluster supporting the database and basic SA service is connected to the general Internet, then the information can be made available to the public.

Resource Sharing

A field server acting as a relay must decide how to share its resources when sending and receiving messages. We account for three resources in this case: the bandwidth in transmission opportunities, the memory buffers used to store messages that need to be sent, and the battery power consumed by sending and receiving the messages (if constrained). The node's owner is able to specify what fraction of each resource s/he is willing to give towards the task of forwarding messages belonging to others. Network software keeps track of how much each message "costs" in terms of each resource. The fraction of available resources and the cost of each message are handled by a *resource manager*. When an opportunity to forward a message (either to the cluster or to another relay) arises, the resource manager it must take into consideration how much energy and bandwidth it should use, as well as the amount of memory available at the next hop to store messages.

The scheduler must then decide when and in which order to send messages. To do this, each message is assigned a *utility*—a value essentially indicating its delivery importance. The utility can come from two sources. The servers that maintain the SA information can send requests to the portable nodes for missing information (i.e., areas not "covered"). Server requests for missing information increase the utility of messages that carry the information. Consumers may create queries, delivered via the field servers, to the SA service for certain information. If multiple consumers request the same information, the utility of the response messages can be increased. On the other hand, content identified as redundant will have its utility decreased. Utility can also be influenced from configured policies that indicate the criticality of a message (e.g., as determined by its tags, author, or timeliness). We model the scheduling decision as an optimization problem. The scheduler in each field server determines the utility of the messages stored in its buffer and runs an optimization solver to identify the order and time of transmissions that maximizes the delivery performance of high-utility messages.

Field servers can also perform *message aggregation* to help eliminate redundant content. For instance, suppose several producers generate messages reporting the same event, e.g. a fire that starts in a house. If several of these messages reach the same node, they can be aggregated in some algorithmic fashion, e.g., discarding multiple photos of the fire. In order to perform aggregation of this kind, it is important (and challenging) to (1) recognize that two or more messages have redundant content, and (2) pick an appropriate strategy to aggregate the messages (e.g., discarding, merging). Recognizing redundancy depends on the specific data types but it can be done with data mining techniques such as clustering or binary classifiers. As some of these techniques can be too computationally intensive for certain devices, the aggregation functionality is likely to be activated only in nodes with more power, e.g., computers installed in powered vehicles such as cars, boats or aircraft.

Security Requirements

There are four main security requirements in our architecture, as follows. (1) Before adding freshly received information to the message database, the origin and integrity of the information should be verified. This can be

used to help assess the data's trustworthiness, and for after-event analysis and attribution. (2) A user who accesses and displays SA information should be able to verify the information originated from a trustworthy source and has not been modified in transit. (3) In some cases, the information sent by a producer may be confidential and/or anonymous, and either the application or the producer might wish to keep it private. (4) The message scheduler might wish to verify the author of information to perform differential scheduling of resources among multiple competing parties by altering the utility computation discussed above.

Regarding the information origin requirement, our architecture supports two main mechanisms. First, we expect some information producers to be operated by first responders or other trusted parties for enhancing their own SA. We assume such producers have associated public key credentials from a common public key infrastructure (PKI) available to the service. These PCDs are capable of producing signed (or encrypted and signed) messages that the back-end servers are able to verify (and decrypt). Second, in cases where a common PKI is not available, we at least aim for accountability by binding the information received by the service to some (hopefully un-forgeable) identity (such as a cellular phone SIM) associated with the PCD. In cases where even this is lacking, other meta-data (location, host address, etc.) may be made selectively available subject to user anonymity/privacy policies. Fortunately, it is possible to indicate in the message database and with associated visualization tools which information is deemed the most and least trustworthy (with anonymous data likely being considered the least likely to be trustworthy).

For receiving PCDs, we rely on standard PKI techniques by assuming that the service has a public key signed by a well-known certification authority. Browsers on the PCDs already come pre-loaded with keys for many certification authorities, enabling them to validate the service's certificate. The third requirement is addressed using standard end-to-end encryption techniques based on, once again, a common certification authority. The fourth requirement implies an on-path relay must be able to validate the origin of a message. To achieve this, a producer attaches an integrity signature to any data it originates; this integrity signature is verified by relays by relying on a common certificate authority. Once the source of a message is verified, a relay can then perform policy-based scheduling of its resources.

FUTURE WORK

We plan to build a prototype of the system described above. In doing so, we will evaluate the practical requirements of using technologies such as natural language processing, object recognition and image compression to deal with identifying redundancies (including semantic ones) and finding intelligent ways to aggregate or compress redundant information. We will conduct a comparative study of scheduling policies and study the performance impact on the coverage of the surveillance map and resource utilization. A longer-term goal is to create a purely distributed version of our surveillance map that can be supported by victims alone, and does not require first responders to support a cluster of servers.

Acknowledgements

The authors would like to thank Megan Finn of UC Berkeley for her helpful feedback on the paper.

REFERENCES

1. Fall, K. (2003). A Delay-Tolerant Network Architecture for Challenged Internets. In Proceedings of ACM SIGCOMM 2003.
2. Seth, A., Kroeker, D., Zaharia, M., Guo, S., and Keshav, S. (2006). Low-cost Communication for Rural Internet Kiosks Using Mechanical Backhaul. In Proceedings of ACM MOBICOM 2006.
3. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J. (2006). Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms. In Proceedings of INFOCOM 2006.
4. Palen, L. and Liu, S. (2007). Citizen Communications in Crisis: Anticipating a Future of ICT-Supported Participation, In Proceedings of ACM CHI 2007.
5. Palen, L., Hiltz, S., and Liu, S. (2007). Online Forums Supporting Grassroots Participation in Emergency Preparedness and Response. Communications of the ACM, 50 (3) (Mar. 2007): 54-58.
6. Liu, S., Palen, L., Sutton, J., Hughes, A., Vieweg, S. (2008). In Search of the Bigger Picture: The Emergent Role of On-Line Photo-Sharing in Times of Disaster. In Proceedings of ISCRAM 2008.
7. Vieweg, S., Palen, L., Liu, S., Hughes, A., Sutton, J. (2008). Collective Intelligence in Disaster: An Examination of the Phenomenon in the Aftermath of the 2007 Virginia Tech Shootings. In Proceedings of ISCRAM 2008.

8. Hughes, A. and Palen, L. (2009). Twitter Adoption and Use in Mass Convergence and Emergency Events. In Proceedings of ISCRAM 2009.

9. Hughes, A., Palen, L., Sutton, J., Liu, S., and Vieweg, S. (2008) "Site-Seeing" in Disaster: An Examination of On-Line Social Convergence. In Proceedings of ISCRAM 2008.