A Wireless System for Reducing Response Time in Urban Search & Rescue

Harsha Chenji, Wei Zhang, Myounggyu Won, Radu Stoleru, Clint Arnett*
Computer Science and Engineering, Texas A&M University
**TEEX Emergency Services Training Institute*
{*cjh, wzhang, mgwon, stoleru*}*@cse.tamu.edu, clint.arnett@teex.tamu.edu*

***Abstract*—Time is a critical factor in the Urban Search & Rescue operations immediately following natural and man-made disasters. Building on our collaboration with first responders we identify *a set of areas for improving response times*: victim detection in collapsed buildings, information storage and collection about buildings (collapsed or not), detection of first responder team separation and lost tools, and throughput and latency of data delivered to first responders. In this paper, we present the design (i.e., software/hardware architectures, and the guiding design principles), implementation and realistic evaluation of DistressNet, a system that targets the aforementioned areas for reducing the Urban Search & Rescue response time. DistressNet, built on COTS hardware and on open standards and protocols, pushes complexity that the very diverse Urban Search & Rescue scenarios pose, to user level applications (apps). Apps in DistressNet run on unmodified hardware ranging from smartphones, to motes and wireless routers. For the benefit of the research community, we also share some lessons learned during our experiences in the design, building and evaluation of DistressNet.**

***Keywords*-Wireless sensor networks, Disruption tolerant networking, Mobile ad hoc networks**

I. INTRODUCTION

Natural and man-made disasters in large urban areas cause increased loss of life and property due to the population density. Urban Search and Rescue (US&R) personnel specialize in the "location, rescue (extrication), and initial medical stabilization of victims trapped in confined spaces" [1] especially in an urban/industrial setting (as opposed to ground, mountain and battlefield search-and-rescue). The Federal Emergency Management Agency (FEMA) of the USA has created several task forces (28 TFs currently) specialized in US&R operations. During the Haiti earthquake, FEMA/US&R were responsible for 47 of 134 lives rescued as mentioned in [2]. Unfortunately, it also mentions that many issues during the response were related to communication and situational awareness.

Computer scientists have looked at improving the ways in which first responders interact with mission critical hardware through human-computer interfaces. Recent advances in computer networking research has resulted in many network architectures and protocols focused on improving the disaster response process. However, it is our opinion, formed as a result of close collaboration with first responders, that the field of US&R has not been sufficiently addressed. The primary goal of DistressNet is to reduce the response time of US&R personnel so that more lives can be saved. Our approach to meet the goal is to eliminate practical roadblocks and enhance the situational awareness of first responders, using a computer networking and wireless sensing paradigm. The high level challenges faced are: 1) providing a robust and reliable ad-hoc infrastructure that integrates a diverse range of battery powered devices as well as third party solutions, 2) high network and sensing performance (high throughput, low delay, high accuracy) in such an ad-hoc environment where mobility is low and inter-node contacts are scarce, and 3) low resource consumption (storage capacity, energy) while achieving high performance in such a system.

The above challenges are too diverse to fully address in a single paper. Instead, we tackle the research questions in some of the challenges represented by four responder requirements, gathered from interactions with first responders. Qualitative requirements like smart seismic monitoring for victim detection under a rubble pile and the digitizing of FEMA US&R wall markings are proposed. Quantitative requirements include the design of an algorithm for detecting network topology fragmentation with minimal communication overhead, yet achieving a bounded delay. High situational awareness can be realized with large network throughput and reduced delay, which necessitates the need for optimal placement of limited hardware in the network. The design principles behind DistressNet and the lessons learned during its evaluation are also discussed.

Concretely, the contributions of this paper are: 1) the first (to the best of our knowledge) system which addresses US&R requirements, 2) a set of applications motivated by requirements gathered from interactions with first responders and the few available FEMA/US&R documents, 3) a set of design principles for mitigating issues encountered in previous deployments, 4) evaluation in a realistic setting and 5) lessons learned during the design and evaluation process.

The remainder of this paper is organized as follows: Section II presents the responder requirements and discusses related work. Section III details the design principles and implementation. Applications are discussed in Section IV and evaluated in Section V. Sections VI and VII conclude the paper with lessons learned during our system implementation and evaluation, and conclusions.

978-1-4673-4883-6/12/$31.00 ©2012 IEEE

II. MOTIVATION AND RELATED WORK

DistressNet targets US&R operations over the geographical area of a large scale disaster, and not a medical triage. Unlike a block level emergency, there may be no usable communication or power infrastructure in the stricken area. The command and control center (C2) is situated outside the area and has limited access to infrastructure. Due to cost/energy considerations, mobile satellite internet access points (e.g., MERS) are uncommon. In this section, we list the responder requirements in such a setting, and how they affect the last victim rescue time (LVRT). We define LVRT as the time until the last victim is rescued, reckoned from the start of the disaster.

A. US&R Responder Requirements

DistressNet was built over two years, based on inputs from first responders as well as iterative improvements from implementing various design choices. The FEMA equipment cache list [3] gives the reader an idea for the size, cost and bulk of equipment currently used by US&R teams. Based on this list and interaction with Texas Task Force 1 US&R we outline the responder requirements, how they affect the LVRT, and their relation to computer network metrics.

Qualitative Requirements

These requirements indirectly reduce LVRT by reducing the time required for personnel to perform their tasks, by enhancing the quality of available data using new methodology or technologies and also by removing practical roadblocks.

REQ1: Smart Victim Detection Under Rubble: Highly sensitive seismic sensors that pick up vibration from a rubble pile were used during the 9/11 emergency to locate trapped victims [4]. First responders can listen for human voices or activity through attached headphones, and can locate them by asking victims to tap on nearby pipes. However, the low frequency sound created by shaking buildings and nearby human activity interferes with this detection process [4]. *Filtering the noise automatically reduces the LVRT by 1) eliminating the "All Quiet" condition required for seismic sensor use (which halts rescue efforts in the immediate surroundings); and 2) enabling the re-deployment of on-site personnel to other areas of the disaster.*

REQ2: Digitized Building Information: Whenever US&R teams search buildings, the search status of the building is indicated using markings (called X-codes, FEMA/INSARAG format) painted in day-glo orange on walls (tagging), for the benefit of other teams. This includes data like the last search date/time, presence of hazards etc. Digitizing such tags using low power motes will provide the C2 with high situational awareness due to the variety of information that can be sensed on motes. *By digitizing building tags and enabling automated data collection, resources can be efficiently allocated by the C2, thus improving the LVRT.*

Quantitative Requirements

These requirements reduce the LVRT by improving existing metrics, such as aggregate network throughput and the time taken to detect a separation in the team. In DistressNet, they are networking related solutions and improve metrics typically addressed by networking research.

REQ3: Fast Team Separation Detection: During US&R operations in a collapsed building, team members may become separated from each other due to falling beams, or they may lose vital tools accidentally. The LVRT increases due to such issues, primarily because of the delay in noticing separation and accounting for missing tools. We present an algorithm that lets each team member know of any separation in the team independent of the team size, even when the separation or "cut" occurs many hops away. Team separation detection delay is measured in seconds. *An app installed on a smartphone, alerts a first responder immediately after a missing tools or team members, enabling recovery from the situation within seconds and hence improving the LVRT.*

REQ4: Improved Situational Awareness: Situational Awareness can be improved with a large amount of accurate data at a high temporal and spatial resolution being made available periodically with the least delay. However, this task is challenging because the network is fragmented due to the size of the area. Much of previous research has been devoted to improving network performance metrics like throughput, packet delivery ratio and delivery latency. We quantitatively measure situational awareness by aggregate throughput in Kbps and delivery latency in seconds. *By placing additional hardware, the aggregate throughput of the network is increased and the end-to-end delay is decreased, providing the C2 with increased situational awareness and hence improving the LVRT.*

B. Related Work

In this section we review the large body of work related to disaster response and the four requirements listed above. While the medical triage area in a disaster has received much attention from academia as well as industry, we feel that needs of *urban search and rescue personnel* in the "field" have not been sufficiently addressed.

The Wireless Internet Information System for Medical Response in Disasters (WIISARD) [5][6][7] is a 802.11 based wireless mesh network (WMN) tailored to provide effective medical response in the event of a disaster. [5] proposes the WIISARD Communication Protocol (WCP) which is a gossip based protocol for data dissemination, and shows that link properties vary between the different stages of the rescue drill. Mobile clients like PDAs and laptops roam around the geographical area while being connected to the Internet via multiple backhaul connections [8]. Digital tags on patients [9] are read by medical personnel using PDAs. Changes to such digital records are tracked and can be easily rolled back in case of conflict due to multiple simultaneous

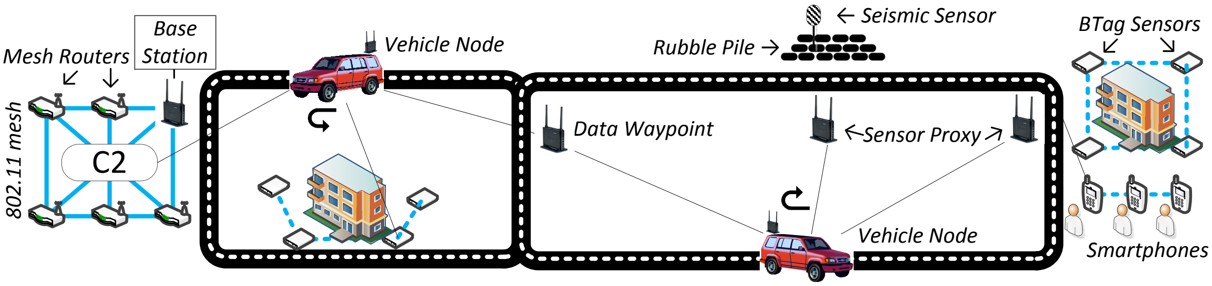


Figure 1. Schematic of a DistressNet deployment showing all components. Data generated by BTag and Seismic Sensors is ferried to the Base Station using Vehicle Nodes. A Data Waypoint improves the data transfer process by creating a contact opportunity between two Vehicle Nodes.

editing. The primary difference between DistressNet and the latest version of WIISARD [5] is that DistressNet is designed for the needs of urban search and rescue personnel when operations occur over a large geographical area. While medical triage is handled by WIISARD, DistressNet address US&R operations like searching for survivors building rubble and building monitoring using low power 802.15.4 devices. As stated in [5], the major contribution is not WCP itself, but the characterization of link quality and human mobility patterns during the medical triage phase.

Project RESCUE [10] provides an overview of a Wireless Mesh Network (WMN) for effective emergency response. [11], [12] argue for a WMN to be used in disaster response. It cites several shortcomings in several real use cases which provide a baseline comparison to such systems. In [13], a hybrid WMN makes use of wireless WANs as a back haul link to access traditional networks. Several portable networked devices make use of routers affixed to lamp posts in order to achieve network connectivity. The SAFIRE project [14] deals with situational awareness for firefighters. Among the many problems dealt with are reliable data dissemination over ad hoc networks. Responders use a WiFi enabled tablet which uses a central push-pull method of data movement. The intended purpose is for use in a local emergency, and not a region wide disaster. [15] has commercial offerings which accomplish network centric warfare. Based on the limited details available, the system offers robust middleware based on 802.11 and/or WiMax based networking. To the best of our knowledge, these systems assume an AC powered connected network and do not offer integration of low power smart devices.

[15] has commercial offerings which accomplish network centric warfare. To the best of our knowledge, these systems assume a powered, connected network and do not offer integration of low power smart devices. Additionally, they address spatially localized disasters or the triage area of disasters which occur over a large area.

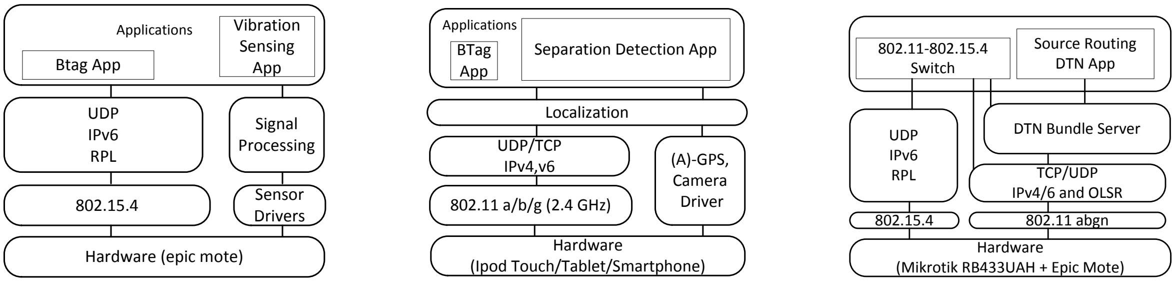
Seismic sensing has been used alongside WSNs to predict volcano activity [16], perform intrusion detection based on footstep detection [17] and heritage building monitoring [18]. In [19], footstep data is analyzed in a variety of environments. In this paper, we do not address footstep detection, but instead evaluate an unsupervised, autonomous noise filtering algorithm using a KNN classifier on a resource-limited embedded device attached to a seismic sensor.

Delay tolerant networking (DTN) has been proposed as a solution for fragmented networks spread over a large area. [20] proposes the use of DTN in a network architecture for disaster response from an information sharing perspective, but does not provide an implementation. Dieselnet and the DOME testbed [21] provide rich information about implementing routing protocols [22]. In [23], data is collected from sensors deployed in a wildlife tracking environment leveraging the frequent movement of zoologists and scientists in the area. However, these systems do not address disaster response.

The problem of intelligent placement of relays to improve the performance of mobile DTNs have been studied [24], [25]. [24] presents a scheme to deploy relays, called throwboxes, in mobile DTNs to maximize data rate between mobile nodes. In [26], analysis on the performance of different relay strategies is presented. [25] later considers other types of infrastructures such as mesh networks and provide cost-performance trade-offs. In this paper, we show that the amount of data transferred between a static and a mobile node depends on the MTU at the network layer, and factor it into the mathematical theory behind waypoint placement. We consider maximization of the end-to-end aggregate throughput in the network and not the data rate between nodes. It is important to note that the amount of unique data passed between nodes has to be maximized since the total data that can be passed is bounded. Additionally, we implement the resulting source routing protocol and waypoint placement algorithm in a realistic scenario.

III. DISTRESSNET DESIGN AND IMPLEMENTATION

In this section we describe the hardware, software and network architectures of DistressNet, and the principles employed during its design. A typical deployment scenario involving all components (Table I) is depicted in Figure 1. These components can be classified into three classes A, B and C (Figure 2).



(a) (b) (c)

Figure 2. DistressNet software architecture: (a) Class A: Sensing; (b) Class B: End User Interactive; (c) Class C: Network Backbone.

Component	Class	Function
BTag Sensor	Class A	Sensing
Seismic Sensor	Class A	Monitoring
Smartphone	Class B	End User Interface
Data Waypoint	Class C	Networking
Base Station	Class C	Networking
Vehicle Node	Class C+A	Networking & proxy
Sensor Proxy	Class C+A	Networking & proxy

Table I
DISTRESSNET COMPONENTS

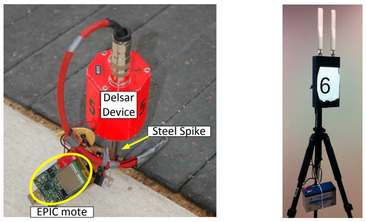


Figure 3. (a) Delsar life detector with steel spike driven into the ground, and interfaced with an EPIC mote; (b) Mikrotik RB433UAH wireless router mounted on a tripod and powered by a battery.

A. *Design Principles*

Based on first hand accounts of US&R deployments and responder requirements, we decided on a set of principles that shall govern our design of DistressNet. A list of applicable system design principles can be found in [27].

PRI1: Unmodified COTS Devices Governmental organizations are increasingly adopting COTS devices because of the available support and software, at fairly economical prices as compared to a custom platform. In many instances, US&R responders have used their own personal iPhones during disasters to email photographs of rubble piles. In any case, one cannot assume a “jailbroken” device where one can have complete control, as is fairly common with hardware platforms used in academic research. Instead, the system has to be designed such that stock capabilities of popular COTS devices are sufficient, so that devices can be borrowed and setup easily. A custom routing protocol or user replaceable batteries are not possible on the iPhone, as an example.

PRI2: Open Standards and Protocols Standardized protocols, preferably of international scope, are emphasized. Certain WiFi channels are allowed in Japan but not in the USA; such issues should be planned for. At every layer of the system, open formats and widely supported protocols make integration of hardware with other international teams much easier.

PRI3: App Oriented Design Because complexity is pushed towards the application layer, updating the system becomes easy and does not need recompiling/reflashing the entire device, especially during disasters. When deployed on a large scale over a variety of heterogeneous devices, PRI3 will significantly reduce roadblocks encountered in platform adoption. At the same time, simplicity in these complex apps is necessary: when a human-computer interface is present, having more than three buttons will cause the device to be left behind in a vehicle, instead of being used by first responders.

PRI4: “Premature Optimization is the Root of All Evil” With DistressNet we first build a proof-of-concept implementation that captures most of the required functionality, and then iteratively optimize the system based on deployment experiences. For example, we trade performance for simplicity in the design of the Source Routing App, by using a simplistic vehicle movement model. The gained simplicity makes it easier to deploy DistressNet as a whole with limited manpower, providing us with valuable experience which we can then use in the next iteration of the Source Routing App.

B. *Implementation*

Class A: Sensing/Monitoring A *Seismic Sensor* (a Delsar Life Detector + EPIC mote, as shown in Figure 3(a)) is a device which monitors rubble piles for vibrations, while a *BTag Sensor* (BTag stands for “Building Tag”) is an EPIC mote which digitizes building tags, implementing REQ1 and REQ2 respectively. They are classified as Class A hardware (low power, battery powered, 802.15.4, no 802.11). The software stack uses UDP, IPv6, Blip2/RPL and TinyOS (Figure 2(a)).

Class B: End User Interactive *Smartphones* are used in DistressNet to access field data. A BTag App (Figure 2(b)) is used to program BTag Sensors once they are deployed, with relevant information, while the Separation Detection App

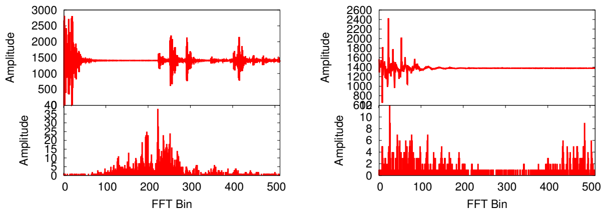
(Figure 2(b)) monitors a team's separation status. They implement part of requirements REQ2 and REQ3 respectively. A new class of hardware, Class B, designates them (they last a few days on battery, 802.11/3G very common, HCI present). iPod Touches were used as smartphones because of the lack of cell tower infrastructure.

Class C: Network Backbone *Sensor Proxies*, *Base Stations*, *Data Waypoints* and *Vehicle Nodes* are Class C components in DistressNet (last a few days on car batteries, 802.11, extensible via USB ports for 802.15.4, ample storage). Because of their dual 802.15.4 and 802.11 functionality (Figure 2(c)), they provide an interface into DistressNet for Class A devices ("802.11-802.15.4 Switch" in Figure 2(c)). A Mikrotik RB433UAH board (Figure 3(b)) has 2 USB slots and 3 MiniPCI slots, allowing for a wide range of field-installable functionality (plugging in a USB GPS receiver can transform a Sensor Proxy into a Vehicle Node). The routers have dual radios, operating in both 2.4GHz and 5GHz (802.11 IBSS mode). Smartphones and laptops connect via the 2.4GHz radio since it is more common, while the other radio is used for DTN and mesh routing. DTN is implemented using IBR-DTN (Bundle server in Figure 2(c)), while the OS is Linux-based OpenWRT.

Design Principles Devices from all three classes need no modification and can be used off the shelf. 802.11 ad-hoc mode is not supported on Android devices (without jailbreaking), which made us choose iOS devices (PRI1). Epic motes and Mikrotik routers do not need any modification. The protocols, connectors and multimedia containers used in the software stack of these devices include 802.11, 802.15.4, TCP/UDP/IPv4/IPv6, RPL, USB, OpenNMEA, micro-SD, MiniPCI, MP4, M4V etc., all of which are internationally accepted open standards or are candidates for standardization (PRI2). DTN follows RFC 5050, OLSR mesh routing is RFC3626. Collaboration between international teams becomes easy, as does third party hardware integration. Smartphone apps can be provisioned on multiple devices irrespective of hardware generation (PRI3). A huge ecosystem of mobile apps is available for first responders to use in the field, due to the popularity of iOS and Android.

IV. DISTRESSNET APPLICATIONS

In this section we discuss the applications, their functioning, and how they address responder requirements. The Vibration Sensing App (Figure 2(a)) implements REQ1 by using a KNN classifier to accurately detect victim responses and filter out noise. The BTag app (Figure 2(b)) on Smartphones presents an interface to the digitized BTag implemented on a Class A device ("Btag" in Figure 2(a)), together implementing REQ2. The Separation Detection App (Figure 2(b)) on Smartphones instantly alerts US&R team members of lost tools or physical separation from other members, thus implementing REQ3. The Source Routing app (Figure 2(c)) on Class C devices implements REQ4 by increasing situational awareness quantitatively, through increasing aggregate throughput.



(a) (b)

Figure 4. Spectrum and signal of (a) stone drop (b) footstep.

A. Vibration Sensing App (REQ1)

The FEMA US&R equipment cache list [3] mentions Delsar Life Detection sensors: a steel spike that is driven into rubble can then be monitored by responders for voices or knocks from victims. Upon manually probing the rubble at different places, the victim can be localized and rescue operations can commence. Since these sensors do not have any native networking capabilities, EPIC motes are used to provide an interface, creating a *Seismic Sensor* component. However, there are sources of noise like footsteps and vibration from nearby vehicles which are also picked up. The goal of the Vibration Sensing App is to automatically detect and classify the source of vibration. To profile these sources the steel spike of the sensor was driven into a small wedge in a pavement outside our building on campus. Three sources of noise/data are profiled: a stone dropped from a height, human footsteps, and a knock made by a hammer on a pavement. The fixed-point in-place 1024-bin FFT of two of the sources is shown in Figure 4.

It is important to note that the amplitude of the signal alone cannot be used to classify a source. Hence, two features were extracted from the FFT: (i) average value of the frequencies weighted by their respective amplitudes (f_1) and (ii) the mean amplitude of the frequencies (f_2). We then used these features in a simple KNN (k-nearest neighbor) classifier, motivated by the fact that classification has to be done on a resource constrained *Sensor Proxy*. Suppose that we have g different types of data $G_1 \ldots G_g$ and n samples for each group, for a total of gn samples $s_1 \ldots s_{gn}$. Let each sample be a vector consisting of two features $[f_1, f_2]$. The KNN classifier first needs to be trained using these samples. Training consists of storing each sample and its corresponding group in memory. Now, given a new sample $S = [F_1 F_2]$ that needs to be classified, $d_i = \sqrt{(F_1 - s_i^{f_1})^2 + (F_2 - s_i^{f_2})^2}$ is computed for each of the gn samples. The k smallest d_i are chosen and the corresponding group for each sample is identified. The most common group in the set of groups is returned as the solution.

ID: 1
test
Living: 5
Dead: 5
Search Code: 5

(a) (b)

Figure 5. (a) Markings indicate that the buildings have been searched; (b) Screenshot of a BTag app on an iPod Touch.

B. BTag App (REQ2)

BTag Sensors are low power devices which manage metadata related to a building from a search and rescue viewpoint. The primary motivation for this component was the current state of art, where US&R personnel use paint on walls (Figure 5(a)) to store information about the current search status of a structure. This includes information like the number of survivors inside, the location of chemical hazards if any and the most recent date/time that the structure was searched. This information is most likely to remain constant and not change very often. Any vehicles in the vicinity which drive by can electronically gather data from the BTag Sensors, especially if they are outfitted with special chemical or air quality sensors. These tags are first programmed by search and rescue personnel once the search is complete, by using the “BTag App” (Figure 2(b)). A screenshot of the app running on an iPod Touch is shown in Figure 5(b).

C. US&R Team Separation Detection App (REQ3)

US&R operations in an unexplored large areas with low visibility and potential hazards (e.g., collapsed tunnel, chemical spills) are dangerous. Any incidents involving team member separation or loss of vital tools (a “cut” in the network) can slow down the victim rescue process because of unnecessary delays. To meet the need for a separation detection method, we develop an iOS application on an iPod touch that enables each team member to monitor connectivity to a team leader *multiple hops away*, and warns a team member of physical separation from the team leader.

This app is inspired by the distributed cut detection algorithm presented in [28]. A node in an electrical network containing a current source will see a change in its potential when there is a partition in the network, enabling it to detect changes in network topology. Similarly, every node n in a computer network maintains a positive scalar value called the “state” $st(n)$, updating it using the formula $st(n) = (\sum_{N(n)} st(i))/(|N(n)|+1)$, where $N(n)$ is the set of one hop neighbors of node n. A special source node S injects a value I by updating its own state using the formula $st(S) = \frac{I+\sum_{N(S)} st(i)}{|N(S)|+1}$. The state of each node converges, given a network topology. A node in the same partition as S after a cut will see its state converge to a higher value, otherwise it drops to zero. The convergence time is fast and the maximum delay in experiencing such a change is bounded, as shown in [28]. Thus, this algorithm helps first responders detect accidental separation in the team by using only one hop communication.

D. Source Routing App (REQ4)

Data Model in DistressNet In DistressNet, mobile teams visit multiple *points of interest* like collapsed buildings in the affected area. These points of interest can be established by the Incident Commander, in keeping with the Incident Command System which is a subcomponent of the National Incident Management System (NIMS) proposed by FEMA[29]. These points of interest eventually act as *sources* and *destinations* of data in DistressNet (for e.g., data generated by Seismic Sensors and BTag Sensors which are placed at specific locations). A special point of interest in DistressNet is the *C2*, which is the base of operations and collects all data from the field. Vehicles used in the environment act as carriers of data between these sources and destinations, either directly or through inter-vehicle data transfer. These operations typically last for days and indicates that we may be able to take advantage of the (planned) paths of vehicles in order to move data more efficiently in DistressNet. Teams are assigned to search areas by a dispatch command using a common vehicle pool. Our assumptions about this model include the fact that vehicles always move in loops on paths established by the Incident Command at a fairly constant speed (like the local speed limit), that all devices in the model have enough on board storage (expandable via USB hard drives, for example), and that vehicles do not arbitrarily change paths or schedules unless planned. The relaxation of these assumptions and solving the routing problem with minimal *a priori* deployment knowledge is ongoing work.

The Source Routing App performs DTN source routing when given a set of waypoints (Class C routers) placed in the area of the disaster. These waypoints act as relays, create contact opportunities between vehicles and hence increases the aggregate throughput. In this subsection we present a mathematical model that globally optimizes aggregate throughput in the network by placing waypoints, and hence makes source routing possible as a consequence. An important feature of this model is that it considers the data transferred per contact to be a function of the MTU size as well as the vehicle’s speed, based on outdoor experiments.

Preliminaries: The amount of data transferred per contact (DTC) between a stationary node and a mobile one, either over 802.11 or 802.15.4, depends on the MTU size at the network layer, in our case the DTN bundle size over TCP. The DTC also depends on many factors including the speed of the vehicle as well as the contact duration. It is therefore important to consider these factors as variables when modeling aggregate throughput in DistressNet. Consider vehicles $V = \{V_1 \dots V_n\}$ in DistressNet, with the path of each vehicle being a loop and hence representable by a closed polygon $Path(v)$ on which it travels at $Speed(v)$. The time taken by a vehicle to go from point A to point B both on $Path(v)$ and along it, is

$$Time(A, B, Path(v)) = Dist(A, B, Path(v))/Speed(v)$$

The known path and speed is partly justified since disaster response scenarios involve organized motor pools and coordinated movement in the area. Let the set of data sources be S and the set of destinations, D.

Any deployment additionally has a set of "flows" Z, with each Z_i having a data source $Z_i^{src} \in S$, a destination $Z_i^{dst} \in D$ and the size of the data Z_i^{data} that can be sent from the source to the destination. Note that a node may act as a source as well as a destination in different flows. A "waypoint" is a router placed at the intersection of the paths of two vehicles such that data can be dropped by one can be picked up by the other. Let X be the set of all possible waypoint locations (areas where at least two paths intersect). A "solution set" for each flow $Z_i \in Z$ means a sequence of alternating vehicles and data waypoints that are capable of carrying data from the source to the destination:

$$\{Z_i^{src}, v_i^1, x_i^1, v_i^2, x_i^2 \dots Z_i^{dst}\}, v_i \in V \text{ and } x_i \in X$$

The time taken for data to flow using the solution set for a Z_i will then be

$$T(Z_i) = T(Z_i^{src}, x_i^1, Path(v_i^1)) + \dots + T(x_i^n, Z_i^{dst}, Path(v_i^n))$$

Z_i^{data} can now be defined as $Z_i^{data} = min(con(v_i^1), con(v_i^2) \dots con(v_i^n))$ where $con(v)$ is the maximal DTC that is possible between a node and the vehicle v traveling at a speed $Speed(v)$. The Waypoint Placement Problem can now be formulated as follows:

> Given an upper bound X_{max} on the number of waypoints (e.g., limited available hardware), choose $X^* \subseteq X$ such that 1) for each flow $Z_i \in Z$, the solution set contains vehicles in V and waypoint locations in X^*, 2) the aggregate throughput $\sum_{z \in Z} \frac{Z_i^{data}}{T(z)}$ is maximized, and 3) the cardinality of X^* is less than X_{max}

Solution: A representative graph G where each element of $S \cup D \cup X$ is a node can be constructed by drawing an edge between any two vertices m_i, m_j visited by a single vehicle v. The weight of this directed edge is $Time(m_i, m_j, Path(v))$. Considering random arrival times, the weight becomes $arr(v) + Time(m_i, m_j, Path(v))$ where $arr(v) < \frac{Path(v)}{Speed(v)}$. $T(Z_i)$ can be modified similarly. Let a binary selection vector $\mathbf{c} = [c_0, c_1 \dots, c_{|X|}]$ denote whether a possible location $x_i \in X$ is chosen to be a data waypoint (1) or not (0). Let the subgraph G^* denote the graph formed by G by removing vertices indexed by a 0 in $\mathbf{c}$. The problem is now to find a binary vector $\mathbf{c}$ such that, operating on G^* we:

$$\text{maximize} \quad \sum_{i=0}^{|Z|} \frac{Z_i^{data}}{T(Z_i)} \qquad (1)$$

$$\text{subject to} \quad T(Z_i) \neq \infty \qquad (2)$$

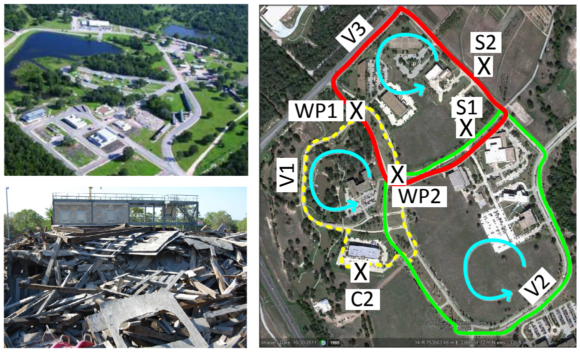
$$\sum_{i=0}^{|X|} c_i \leq X_{max} \qquad (3)$$

Algorithm 1 Polynomial Placer

```
1: $X^* \leftarrow \phi$
2: for each $Z_i \in Z$ do
3:     $paths \leftarrow$ depth-first-search in $G$ between $Z_i^{src}$ and $Z_i^{dst}$
4:     $path \leftarrow$ the path in $paths$ with maximum $\sum \frac{Z_i^{data}}{T(Z_i)}$
5:     Add all vertices in $path$ to $X^*$
6:     Remove $Z_i^{src}$ and $Z_i^{dst}$ from $X^*$
7: end for
8: $\mathbf{c} \leftarrow \mathbf{0}$
9: for each $x_i \in X$ do
10:     if $x_i \in X^*$ then
11:         $c_i \leftarrow 1$
12:     end if
13: end for
```

Constraint 2 ensures that there is always a path in G^* between the source and destination for each flow. This is because the delay for a nonexistent edge will be set to `inf`, or equivalently, G^* can be made fully connected with the newly created edges having a very large weight. Constraint 3 ensures that the number of waypoints deployed is less than or equal to the maximum possible. This problem can be recognized as a binary integer programming problem and is thus NP-hard. Popular heuristics include the branch-and-bound algorithm, which is available in MATLAB as `bintprog` or various algorithms available in ILOG/CPLEX. Once the selection vector is available, **source routing** can be performed by building the optimal path for each flow in G^* and noting the waypoints used in that flow.

A Polynomial Heuristic can be admitted if X_{max} is unbounded. The heuristic in Algoritm 1 chooses $\mathbf{c}$ such that the aggregate throughput is maximum. First, all simple paths between the source and destination for each flow is computed using a depth first search algorithm (Step 3). For each path, the throughput is computed by dividing the time taken for a path (sum of edges) into the maximum data that can be transferred on that flow, by considering per-contact data transfer. The path with maximum throughput is then chosen (Step 4). All the vertices in this path excluding the source and destination are added to a previously empty set X^* (Steps 5, 6). This process is repeated for each flow. The set X^* will then contain the set of vertices which will maximize the aggregate throughput by maximizing the throughput for each flow, given that X_{max} is unbounded.



(a) (b)

Figure 6. (a) (Top) Disaster City and (Bottom) A wood rubble pile; (b) Map of Deployment: S1,S2 are sources, V1,V2,V3 are vehicles, WP1,WP2 are Data Waypoints, X are locations.

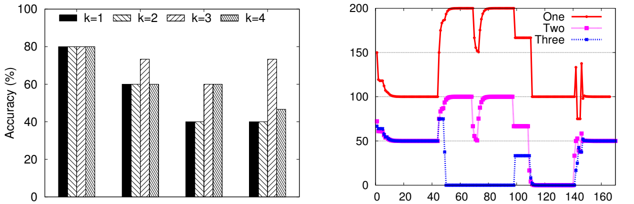
V. EVALUATION

DistressNet is a complex wireless, sensor, ad hoc and delay tolerant system. In this section we present the performance evaluation of the DistressNet apps which fulfill requirements REQ1, REQ3 and REQ4. REQ1 has been evaluated in the rubble piles of Disaster City (Figure 6(a)) near College Station, which is a comprehensive 52-acre training facility for emergency responders with a realistic disaster environment, including several rubble piles of wood and concrete. Due to the functional and qualitative nature of the Building Tag App (REQ2), it is used as a data source in REQ4's evaluation and is not evaluated on its own.

A. *Vibration Sensing App (REQ1)*

We evaluated our Vibration Sensing App in Disaster City on three different rubble piles: one consisting of wooden rubble, one of concrete, and another with a combination of concrete and mud. In the latter one, the soft mud dampens the vibrations caused inside the pile and hence makes detection with a seismic sensor more challenging. Samples for different types of events were gathered at each of these piles: a stone drop, a footstep and a hammer strike. Half of the samples were used to train the KNN classifier, and the other half to evaluate performance. All samples were taken at slightly different strike intensities and distances from the sensor.

Results are shown in Figure 7(a). "wood1" represents samples taken at the wooden pile with the default sensitivity threshold of 25 and "wood2", at a threshold of 50. A higher threshold implies lower sensitivity. This higher threshold was not possible on the two other piles since the sensor could not register soft knocks and events. We conclude that a $k = 3$ provides for optimal performance from the KNN classifier with an average detection of accuracy of 73.33% independent of type of rubble, strike intensity and the distance from the seismic sensor. This evaluation is qualitative in that the LVRT depends upon the number of sensors deployed and their location, indirectly helped by a high detection accuracy.



(a) (b)

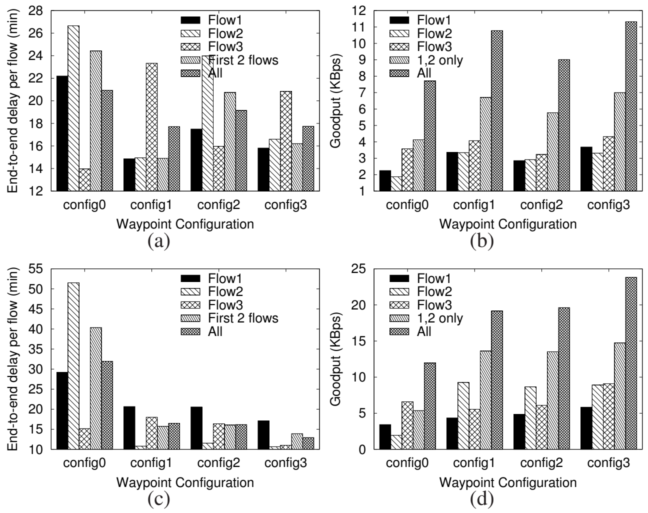
Figure 7. (a) KNN classifier accuracy as a function of k; and (b) Graph of state versus time for a team of three responders.

B. *Separation Detection (REQ3)*

The effect of separation upon the state of a team member is shown in Figure 7(b). An experiment was conducted inside an urban building where iPod touches running the separation detection app were given to each member. "One" is the team leader and hence injects a constant state into the network. Initially, all the team members were present in a single room until time 30. Then, One and Two separated from Three by going into another room. As a result, the state of Three drops to zero since it is no longer connected to One, and the states of Two as well as One increase and converge (time $40-60$). Then, One and Two move around in the large room with lot of metallic wall sized objects, causing disconnection. This disconnection is temporary and does not signal a separation. Later, Two returns to the same room as Three at time 95. As a result, the state of Three increases for time $100-110$ due to the residual state brought by Two, but both of them quickly decrease to zero at 110 since they are no longer connected to One. Finally, One reunites with Two and Three at 140 causing all of their states to converge once again to their initial values. The average detection delay, looking at each of the three separation events and the corresponding state at that time, is $\frac{(45-30)+(98-95)+(143-140)}{3} = 7s$. The detection delay for separation as opposed to rejoining is a little longer because of the guard interval before a node declares a neighbor as disconnected.

C. *Source Routing App (REQ4)*

We designed a deployment (Figure 6(b)) involving three cars and three flows with three data producing/consuming nodes. Flow1 in the following text refers to Seismic Sensing data sent from Source1 to the C2, Flow2 is BTag Sensor data from Source2 to the C2, and Flow 3 is Seismic Sensing data from Source1 to Source 2 (representing sound samples being sent to a subject matter expert). The two possible waypoint locations were WP1 and WP2. There are four configurations possible with two locations: none (config0), both (config3), WP1 only (config1) and WP2 only (config2). The goodput for each flow was experimentally measured for all possible waypoint configurations. Because a large number of Seismic



(a) (b)

(c) (d)

Figure 8. Expt. 1: Epidemic routing performance in terms of (a) latency and (b) goodput; Expt. 2: Source routing performance in terms of (c) latency and (d) goodput.

Sensors and BTag Nodes are required to impose a significant load on the system, we decided to synthetically generate data at high rates on the respective Sensor Proxies.

[Expt. 1: Epidemic Routing/No DTC optimization] For this experiment all nodes performed epidemic routing. Results for the goodput and delay are presented in Figure 8(a,b). Payload size was chosen to be 100KB, each flow generated data at every 20 seconds, vehicles moved at around 20mph. It has to be noted that the sources themselves act as data waypoints due to the nature of Epidemic routing - hence, the goodput improvement between the configurations is not that high as compared to the following experiment. config3 proved to be optimal by providing the highest aggregate goodput across all the flows. If we consider only flows 1 and 2, since Flow3 does not need any additional waypoints (since the same vehicle passes through sources 1 and 2), configs 1 and 3 are almost equal.

[Expt. 2: Source Routing/With DTC optimization] For this experiment, values from the WiFi and 802.15.4 contact experiments were used to determine the payload sizes of flows. In addition to source routing replacing Epidemic routing of the previous experiment as well as choosing an optimal bundle size, security at the DTN layer was enabled. Flow2 was converted to a 802.15.4 flow with the vehicle picking up data from Source2 using 802.15.4 instead of Wifi. Flow1 was still WiFi based - this meant all deliveries to Source2 were made over Wifi. The payload sizes for Wifi and 802.15.4 were chosen to be 300KB and 90B respectively. However, once a vehicle picked up 90B packets, they were marshaled into 300KB DTN bundles. Data was generated every 30 seconds. The maximum delay in config0 is high because there is no data replication (and only opportunistic contact between vehicles), but when waypoints are present, the delay is comparable in spite of the increased payload size and overhead due to security.

The aggregate goodput increases by 110% (11.321 to 23.833 KBps) when source routing is used since unnecessary copies of bundles are not created, leading to efficient and non-redundant per-contact data transfers. A 27.32% decrease in latency (17.75 to 12.9 min) is observed because of the non-redundancy. Thus, the C2 is able to obtain data from the field and achieve a high level of situational awareness by making use of the vehicle movement patters in the field by distributing source routes to nodes.

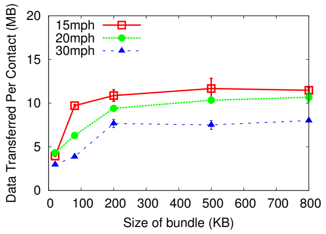


Figure 9. DTC vs Bundle size for WiFi over TCP/IPv4

VI. LESSONS LEARNED

Observations made during the deployment (noticing that DTC depends on the MTU), lead to changes in the theory behind the design. In this section we share those observations, how our hardware requirements have evolved and our more recent efforts on energy efficiency in DistressNet.

Hardware Evolution: Initial versions of DistressNet attempted to use Linksys WRT54GL because of its popularity in academic research. It has a single WiFi interface and poses problems for DHCP and other network configuration issues. Lack of 802.15.4 functionality was a problem and prevented seamless integration of BTag Sensors/Seismic Sensors. The Netgear WNDR3700v1 (1 USB port, dual radios) was then chosen to fix the problem. All the 5GHz interfaces of all the routers belong to the same subnet (192.168.50.x), while each 2.4GHz interface of a router has its own subnet (192.168.x.0), making DTN and meshing very easy. However, limited on board flash memory meant that adding fat libraries like Boost was problematic. The Mikrotik RB433UAH has 512MB ROM/128MB RAM. This platform allowed for luxurious debugging as well. *Decisions about upgrading the hardware saved a lot of time by preventing many practical roadblocks.*

Data Transferred Per Contact In our experiments we noticed that the DTC depends on the MTU at the network layer (both over 802.15.4/UDP/RPL and 802.11/TCP/DTN). In DistressNet, DTN routing is implemented in the application layer by design, using "convergence layers" (RFC 5050) for network compatibility. Vehicles pick up data from roadside nodes like Sensor Proxies over the DTN layer, which happens over TCP as UDP does not support bundle sizes greater than the MTU. Since the Bundle Server has to look at its local cache and decide whether to transfer a particular bundle to this vehicle or not, the DTC depends on the size of the bundle. The optimal bundle size turns out to be $> 200KB$ for a variety of vehicle speeds (Figure 9). Interestingly, drive-by data pickup over 802.15.4 also exhibits the same behavior, except that in this case, the maximum MTU size yields the highest DTC.

802.11 PSM and IBSS: In order to truly extend the lifetime of the system, duty cycling the 802.11 radios which draw around 200mA when active is necessary. Because the majority of the devices in DistressNet operate in the IBSS mode, power saving mode (PSM) support for IBSS mode in the hardware is essential. However, implementing this functionality in the linux ath9k drivers (used by the R52Hn WiFi cards) is not a trivial task. Hence, we could only enable PSM functionality for Smartphones and laptops by setting the 2.4GHz interface of Class C routers to operate in AP mode instead of ad-hoc. Experimental verification was performed using MiniPCI extenders which allowed us to isolate and measure the current drawn by a MiniPCI card, used in the RB433UAH as well as older laptops. *Once IBSS mode PSM is implemented, rudimentary energy savings can be achieved by simply enabling it in the OS.*

VII. FUTURE WORK AND CONCLUSIONS

DistressNet reduces the disaster rescue time by fulfilling first responder requirements. A Vibration Sensing App running on an EPIC mote is able to identify victim-specific sounds with 73.3% accuracy, while a BTag App digitizes search statuses as well as survivor data, thus enabling automated data collection. Requirements are fulfilled quantitatively as well, with the Team Separation detection algorithm running on an iPod Touch being able to detect separation within seconds. The Source Routing App increases throughput by 110.52% while reducing latency by 27.32%. The whole system is designed for the integration of heterogeneous technologies, using only battery powered devices, and relevant subsystems have been evaluated in realistic conditions.

Acknowledgements: The authors acknowledge the NSF's support through grants #1127449, #1145858 and #0923203.

REFERENCES

[1] http://www.fema.gov.

[2] www.llis.dhs.gov/displayContent?contentID=43962.

[3] http://www.fema.gov/emergency/usr/equipment.shtm.

[4] http://www.designnews.com/document.asp?doc_id=213077.

[5] O. Chipara, W. G. Griswold, A. N. Plymoth, R. Huang, F. Liu, P. Johansson, R. Rao, T. Chan, and C. Buono, "WIISARD: a measurement study of network properties and protocol reliability during an emergency response," in *MobiSys*, 2012.

[6] S. W. Brown, M. William, G. Griswold, B. Demchak, B. Leslie, and A. Lenert, "Middleware for reliable mobile medical workflow support in disaster settings," *AMIA Annu Symp Proc*, 2006.

[7] M. Arisoylu, R. Mishra, R. Rao, and L. A. Lenert, "802.11 wireless infrastructure to enhance medical response to disasters," *AMIA Annu Symp Proc*, 2005.

[8] J. P. Killeen, T. C. Chan, C. Buono, W. G. Griswold, and L. A. Lenert, "A wireless first responder handheld device for rapid triage, patient assessment and documentation during mass casualty incidents," *AMIA Annu Symp*, 2006.

[9] L. A. Lenert, D. A. Palmer, T. C. Chan, and R. Rao, "An Intelligent 802.11 Triage Tag for medical response to disasters," *AMIA Annu Symp Proc*, 2005.

[10] S. Mehrotra, C. Butts, D. Kalashnikov, N. Venkatasubramanian, R. Rao, G. Chockalingam, R. Eguchi, B. Adams, and C. Huyck, "Project rescue: Challenges in responding to the unexpected," in *SEISC*, 2004.

[11] R. B. Dilmaghani and R. R. Rao, "An ad hoc network infrastructure: Communication and information sharing for emergency response," *IEEE ICWMCNC*, 2008.

[12] B. Manoj and A. H. Baker, "Communication challenges in emergency response," *Commun. ACM*, vol. 50, 2007.

[13] R. B. Dilmaghani, B. S. Manoj, B. Jafarian, and R. R. Rao, "Performance evaluation of rescue mesh: A metro-scale hybrid wireless network," in *WiMesh, SECON*, 2005.

[14] B. Xing, S. Mehrotra, and N. Venkatasubramanian, "RADcast: Enabling reliability guarantees for content dissemination in ad hoc networks," in *INFOCOM*, 2009.

[15] "Network-centric warfare and wireless communications." http://goo.gl/xYZtw.

[16] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *EWSN*, 2005.

[17] G. Mazarakis and J. Avaritsiotis, "A prototype sensor node for footstep detection," in *EWSN*, 2005.

[18] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment," in *IPSN*, 2009.

[19] A. Pakhomov and T. Goldburt, "Seismic signal and noise assessment for footstep detection range estimation in different environments," in *SPIE*, vol. 5417, 2004.

[20] K. Fall, G. Iannaccone, J. Kannan, F. Silveira, and N. Taft, "A disruption-tolerant architecture for secure and efficient disaster response communications," in *ISCRAM*, 2010.

[21] H. Soroush, N. Banerjee, A. Balasubramanian, M. D. Corner, B. N. Levine, and B. Lynn, "DOME: A Diverse Outdoor Mobile Testbed," in *HotPlanet*, 2009.

[22] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: robust replication routing in wireless networks with diverse connectivity characteristics," in *Mobicom*, 2011.

[23] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pásztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef, "Evolution and sustainability of a wildlife monitoring sensor network," in *SenSys*, 2010.

[24] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, "Capacity enhancement using throwboxes in dtns," in *MASS*, 2006.

[25] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: enhancing mobile networks with infrastructure," in *Mobicom*, 2008.

[26] M. Ibrahim, P. Nain, and I. Carreras, "Analysis of relay protocols for throwbox-equipped dtns," in *WiOPT*, 2009.

[27] A. Bahill and R. Botta, "Fundamental principles of good system design," *EMJ - Engineering Management Journal*, vol. 20, no. 4, pp. 9–17, 2008.

[28] P. Barooah, H. Chenji, R. Stoleru, and T. Kalmar-Nagy, "Cut detection in wireless sensor networks," *IEEE TPDS*, vol. 23, pp. 483 –490, march 2012.

[29] http://training.fema.gov/EMIWeb/IS/ICSResource/index.htm.