31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

DiNAR: Health Monitoring of IT Systems in Emergency Response

Kartik Deshpande and Aura Ganz, *IEEE Fellow,* University of Massachusetts Amherst

Abstract—**Disaster management and emergency response mechanisms have come of age post 9/11. Undoubtedly the introduction of Information Technology (IT) in emergency response management has been revolutionary from the perspective of better management of the emergency response. However, the various components which make up the IT infrastructure, such as sensors, computers, PDAs, network devices, etc, are inherently volatile in nature. To overcome this volatility and increasing the robustness of the emergency system we designed and developed a health monitoring tool, denoted DiNAR, for disaster systems. This tool will help the incident commander to actively monitor the state of the IT infrastructure which has been deployed in the field as well as the applications running at remote sites. DiNAR uses Manager-Agent based architecture and a novel agent initiated message exchange.**

1. INTRODUCTION

Emergency response management has over the years seen significant changes in its procedures and technology. In [1] the authors identify 9/11 as a key turning point in emergency response management. Post 9/11 there was a concerted effort to improve the ability to respond to emergency situations. Specifically, major efforts started aimed at improving the survivability and availability of underlying networks [1,2]. Along with this, research on modern methods of information collection and transmission using ad-hoc wireless networks, handheld devices, sensors and web based resources was also encouraged [1].

Undoubtedly the introduction of IT in emergency response management has been revolutionary from the perspective of better management of emergency response systems. Figure 1 shows an emergency response system with its main components (e.g. the DIORAMA system [3]) such as: 1) wireless network devices deployed at the disaster site (used for bothwireless local area network communication and cellular communication)-such devices interconnect the devices on the disaster site among themselves as well as relay the information from the disaster site to the remote site, 2) computing and communication devices at the disaster site such as sensors, PDAs, laptops, RFID readers etc-such devices collect the relevant information from the disaster site, process it and transmit it to the local and remote servers through the wireless network, 3) servers at remote sites which process, store, manage and display the information collected at the disaster site, and 4) applications which reside on computing devices on the disaster site (e.g. PDAs and laptops) as well as on remote computing devices (e.g. servers) that collect, process, transmit and display the information from the disaster site.

It is important to notice that the components which constitute the IT infrastructure are inherently volatile in nature. Devices like sensors, laptops, and PDAs are subject to bootups, power outages etc. Applications are subject to crashes and unresponsiveness. The network as a whole (Internet + Ad-hoc networks in the disaster area) is subject to various outages and communication failures.

Maintaining the integrity of emergency response systems faces the following challenges:

- Since these emergency response systems operate under ad hoc chaotic scenarios, the probability of device failures/outages and application crashes significantly increases.
- The personnel setting up the IT infrastructure in the disaster site is not technically trained to handle and troubleshoot the system failures.
- Maintenance of the IT infrastructure may not be a first priority in disaster management, i.e., there may be no trained personnel on site to monitor, diagnose and repair the IT infrastructure.

Therefore, there is a need of automated management of the whole system, which includes the device, applications and networks. Even enterprise networks with a wealth of IT expertise still need automated network fault and performance management software to respond to network and device failures.

In this paper we introduce DiNAR tool which automates the network monitoring task, enabling the incident commander and/or the remote IT personnel to diagnose the emergency management network deployed at disaster sites as well as at remote sites.

The remainder of the paper includes the following sections. Section 2 introduces the proposed DiNAR architecture and Section 3 concludes the paper.

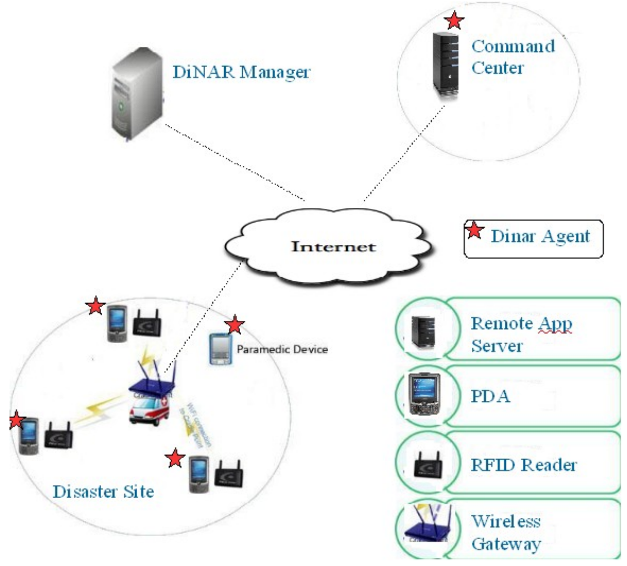

Figure 1 – Emergency IT infrastructure and DiNAR Manager-Agent setup

2. DiNAR ARCHITECTURE

DiNAR architecture follows a Manager-Agent based model (see Figure 1) and draws inspiration from WBEM [4] and YAP [5] in the way management information is represented and collected. DiNAR architecture consists of two main entities, the DiNAR manager and agent. An agent is a daemon service installed on all managed nodes in the field like PDAs, laptops, wireless routers etc.. While a Manager which resides on a server in the remote control center is a central application to which agents report the management information. As shown in Figure 1, every computing system hosts an agent as shown by the red star. The manager and the agents communicate using HTTP calls over the Internet. Thus the manager is expected to run on a known URL.

We next provide more details on the DiNAR agent design, DiNAR manager design, Manager-Agent interaction and the data model.

Figure 2 shows a block diagram view of an agent which is an independent application running on the device. The functionality of an agent has been split into multiple components with a motive to keep things simple and modularized. Next, we describe each component:

1) **Agent Daemon:** This is the main backbone process of DINAR agent and controls all the other components of an agent. One of the initial tasks for this daemon is to locate the manager and initiate a contact. After the initial contact with the manager, the agent starts the information engine and receives regular updates from it. This update information is structured and passed onto the manager at regular intervals.

2) **HttpClient module:** This is the communication engine of the agent. The agent communicates with the manager using the Http protocol. The HttpClient module manages the establishment and maintenance of the Http Connection with the manager. It uses the Apache HttpClient 3.1 library to perform all the Http operations. It also tries to maintain only one connection to the manager throughout and keeps this connection persistent.

3) Information Engine: The information engine is the Management Information database. It stores all the relevant and needed status parameters and tracks them continuously. As described above, the agent daemon interfaces with the information engine to get hold of the current state of the machine. The information engine acts as a central docking point for all information adapters. Information Adapters are specialized modules for continuously monitoring a certain sub component and getting its state information.

4) Information Adapters: Information adapters are specialized modules for continuously monitoring a certain environment and getting its state information. The number and class of adapters can be customized. If the device is a hand-held device in the field, then it can have adapters for the interfaced gadgets, applications etc. If the device is a central web server, then it would mainly contain application adapters. Apart from all these adapters, all agents must have a mandatory generic adapter. The function of a generic adapter is to gather vital parameters of the health of the machine itself like CPU usage, memory usage, power level etc. Except the generic adapter, the rest are optional. This architectural decision on information adapters will help in making DiNAR apply to a wide variety of scenarios. Addition of new monitoring environment would just need new adapters to be plugged into the information engine.

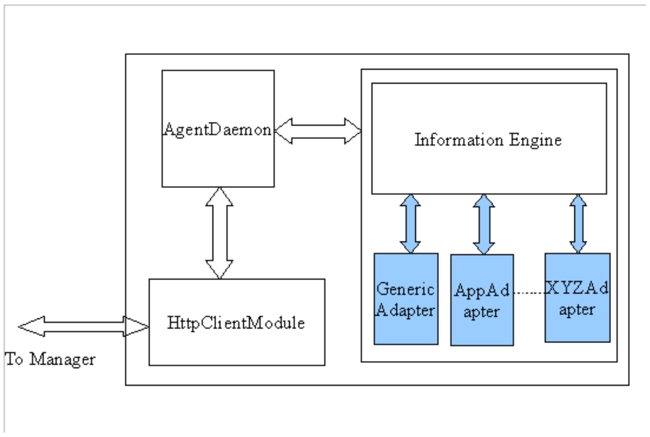

Figure 2 - DiNAR Agent

● *DiNAR Manager design:*

DINAR manager is the focal point of the whole DINAR system. It is the information sink for all the DINAR agents running across the environment.

The manager has to configure and facilitate the data transaction process. In association with various agents it provides seamless flow of management information.

This involves handshakes with every agent, initializing an object repository etc. Figure 3 shows the block diagram view of DINAR Manager and its components.

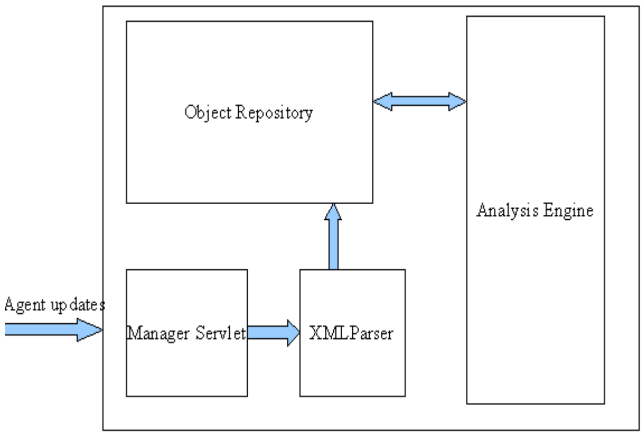

Figure 3 - DiNAR Manager

1) **Manager Servlet:** This module acts as a global interface of the manager. It listens on the assigned manager port and accepts the updates from all the agents. The updates from the agents are in XML format. The servlet extracts the xml data and passes it to the XML parser for further processing.

2) **XML Parser:** This module parses the incoming XML data from the servlet and creates objects for each of the elements in the XML document. It converts the XML document into a set of managed DiNAR Objects (Java) in the system memory.

3) **Object Pool:** DINAR manager looks at every device and its components and attachments as an object. Even the DINAR agent is an object. It instantiates an object for every new device or component it manages. This object is an abstraction of the real device. Whenever updates are received for an already created object, only the property of the object is updated.

4) **Analysis Engine:** Analysis engine will analyze the management information received and correlate it to identify problems and warnings.

● *Manager-Agent Interaction:*

We describe the initial handshake between the agent and the manager. As soon as the agent boots up and knows the URL at which the manager is residing, it first issues an HTTP request with messageType set to “Hi”. Along with this, it also sends a polling interval, which indicates to the manager the frequency with which it should expect updates. If the manager receives all this information correctly, then it would send a “Hi” response. The agent which receives the “Hi” response now knows that the initial handshake has ended. The agent now sends the skeleton XML document which represents the device, applications and the attached gadgets which the agent will be monitoring. There are no status updates in this XML document, just the skeletal XML. After receiving this skeletal document successfully the manager sends back an ACK to indicate to the agent that everything was received successfully. After the ACK is received, the agent starts a timer and after every t secs (equal to the polling interval) it sends out the updated XML document. This continues until the agent or the manager are stopped.

● *Data Model:*

The data model is a very critical component of any system management application. DINAR uses a data model which is an extension of CIM modeling technique in WBEM [4]. The data model represents the class structure which will be used to abstract the real world network. The data model is specific to the actual application scenario and details of the emergency response network. For the emergency system shown in Figure 1, the data model will include classes which represent the disaster site devices and gadgets like RFID readers, RFID tags etc. Figure 4 depicts a piece of this data model.

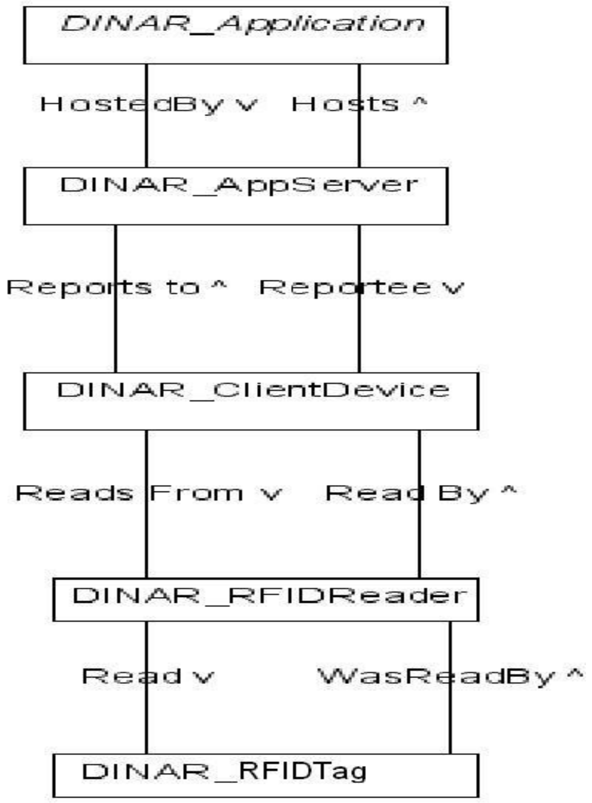

Figure 4- DiNAR Data model

3. CONCLUSION

In this paper we introduced DiNAR tool which performs real time information collection on the health of the disaster system components and applications This tool is crucial for maintaining the health of the disaster system both on the disaster sites as well as on the remote sites.

Acknowledgement: This project was supported in part by the following grants: Grant number ANI-0434985 from the National Science Foundation and Grant Number R21LM008942-01A2 from the National Library of Medicine. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation, National Library of Medicine or the National Institutes of Health.

BIBLIOGRAPHY

[1] M. Turoff, M. Chumer, B. Van De Walle, X. Yao. "The Design of a Dynamic Emergency Response Management Information System". Journal of Information Technology Theory and Application, January 1, 2004.

[2] R. R. Rao, J. Eisenberg, T. Schmitt, "The Role of IT in Mitigation, Preparedness, Response and Recovery", Book ISBN: 0-309-66744-5, National Academics , 2007.

[3] DIORAMA: diorama.ecs.umass.edu

[4] J. P. Thompson. "Web-Based Enterprise Management Architecture", IEEE Communications Magazine, pp. 80-86. March 1998.

[5] C.J. Chiang, et al. "Generic Protocol for Network Management Data Collections and Dissemination", IEEE Military Communications Conference, October 2003.