ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.4

Disaster Recovery Using Networks and Smartphones

Vedaraj[1], Samundeswari.S[2], Nandhini.K[3], Theepigasri.S[4]
Associate Professor[1], Student[2, 3, 4]
Department of Computer Science and Engineering
R.M.D Engineering College, R.S.M. Nagar, Affiliated to Anna University, Tamil Nadu, India

Abstract:
In this paper, we investigate how to network smartphones for providing communications in disaster recovery. By bridging the gaps among different kinds of wireless networks, we have designed and implemented a system called TeamPhone, which provides smartphones the capabilities of communications in disaster recovery. Specifically, TeamPhone consists of two components: A messaging system and a self-rescue system. The messaging system integrates cellular networking, ad-hoc networking, and opportunistic networking seamlessly, and enables communications among rescue workers. The self-rescue system groups, schedules, and positions the smartphones of trapped survivors. Such a group of smartphones can cooperatively wake up and send out emergency messages in an energy-efficient manner with their location and position information so as to assist rescue operations. We have implemented TeamPhone as a prototype application on the Android platform and deployed it on off-the-shelf smartphones. Experimental results demonstrate that TeamPhone can properly fulfill communication requirements and greatly facilitate rescue operations in disaster recovery.

1. INTRODUCTION:

The main aim of this project is to provide communication in disaster area between rescue workers and trapped survivors.

2. EXISTING SYSTEM:

In existing system user cannot communicate with others seeking help due to power failures and network failures. These failures impact the rescue operations in-order to find out victims in different area. During the disaster recovery, communication is mandatory for coordinating the rescue operations. However during earthquakes, the cellular towers will be destroyed and thus cellular communication of smart-phones gets blocked.

3. PROPOSED SYSTEM:

In this paper, we propose Teamphone, a platform for communication in disaster recovery, where smartphones are teamed up and work together to provide data communications. By exploiting Wi-Fi and cellular modules of smartphones .Teamphone seamlessly integrates cellular networking, in infrastructure-constrained and infrastructure-less scenarios .Teamphone also enables energy-efficient methods for trapped survivors to discover rescue workers and send out emergency messages, by carefully addressing the wake-up scheduling of smartphones. The emergency message includes the coarse-grained location and position information of trapped survivors, which is derived from the last known locations of their smartphones and the network formed by these smartphones. We implement Teamphone as an app on the Android platform and deploy it on off-the-shelf smartphones. Experimental results demonstrate that Teamphone can properly fulfill the communication requirements and greatly facilitate rescue operations.

3.1. Advantages

- There has been no change in his outdoor environment, lifestyle or medications.
- Our research is the first step towards evaluating whether access to data related to patient’s living surroundings can help doctors in continuous monitoring of the indoor air quality of their asthma patients and incorporate them with clinical records that contain information on an individual’s asthma triggers, allergies, medications, and past emergency room visits for further insights on the role played by the indoor environment in asthma management
- The VOC reading of the sensor also accounts for carbon monoxide (CO).

4. BLOCK DIAGRAM:

SYSTEM DESIGN

MODULES

1. **Network Formation.**
2. **Wakeup Scheduling**
3. **Emergency Alert**
4. **Initiating Recovery**

Network Formation:

First we can create a network node assume the communication range of a node is finite. By providing distance and range that is Coverage of a particular node. Node in the network would contain unique name and port number to communicate with other node. Node need to find their nearby neighbor before starting any communication. Neighbor is calculated based on the coverage of each node , when the node comes the coverage range of the other node then the two node will consider as the neighbor node .

Wakeup-Scheduling:

In this module we handle with the battery status of trapped survivors. The battery life of smart-phones must last as long as possible, since rescue operations may last for hours or even days. Therefore, the self-rescue system must be energy-efficient. Since trapped survivors are most likely difficult to discover, rescue crews may not infer the location of trapped survivors, even if they have received emergency messages from them. On concerning these problems we use the concept of wakeup scheduling. Here we aggregate the trapped survivors by their disaster type and choose a head among the group based on high battery level. When the emergency button is triggered a group is formed within the trapped survivors and a head node will be selected based on the battery percentage .All the other nodes will be in sleeping state while the head node is in wakeup state will be looking for message transfer. The head node collects all the necessary information such as position and counts of trapped survivors (nodes) within its group and form an emergency message which is to be sent to the nearby Rescue Worker.

Emergency Alert:

Here we look at how the messaging system satisfies the data communication within a routing path. The messaging system get invoked once when the emergency is triggered. A broadcast message is generated automatically in this messaging system. After Wakeup Scheduling the head node started broadcasting a message like “help me !” .When any rescuer node enters into the particular range of the head node ,they will receive a broadcasted message. When the rescuer receives any broadcasted message start scanning the trapped survivors by providing a wifi-hotspot. Thus, the chosen head will send an emergency message with location information to facilitate rescue operations .

Initiating Recovery:

In this final module once after sending the emergency message with trapped survivors position and location information to rescue workers, the rescue worker will forward the message to their nearby command center via opportunistic network using opportunistic routing. The rescuer node search for nearby rescuer node within their range and forwards the info to them. This message transfer runs continuously until it reaches the nearby command center. The command center finds the route between rescue workers in disaster region using AODV routing protocol. After finding the path between rescue workers the command center commands respective rescue workers to travel towards the position of trapped survivors. Thus by using our Team-Phone framework in a mobile ad-hoc network, many trapped survivors will be recovered soon and safely.

CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

●Program should be simple, clear and easy to understand.
●Naming conventions
●Value conventions
●Script and comment procedure
●Message box format
●Exception and error handling

CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

●Program should be simple, clear and easy to understand.
●Naming conventions
●Value conventions
●Script and comment procedure
●Message box format

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

●Proper default values for the variables.
●Proper validation of values in the field.
●Proper documentation of flag values.

SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

● X – User has performed illegal operation.
● ! – Information to the user.

3 TEST PROCEDURES

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

TEST DATA AND OUTPUT:

UNIT TESTING:

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

2 FUNCTIONAL TEST

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

PERFORMANCE TEST:

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

INTEGRATION TESTING

- Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.
- The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested

TESTING TECHNIQUES / TESTING STRATERGIES

a)TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

b) WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

c) BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

d) SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works "outward" toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

e) INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is "putting them together"- interfacing. There may be the chances of data lost across on another's sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

f) PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

g) SECURITY TESTING

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

h) VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

i) USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

10. CONCLUSION:

Thus, to facilitate the trapped servers,we use the technique of monitoring and controlling every individual's mobile.

By knowing their status and by controlling their situations they will be rescuied by rescue workers and control commanders

11. REFERENCES:

[1] K. Lorincz, et al., "Sensor networks for emergency response: Challenges and opportunities," IEEE Pervasive Comput., vol. 3, no. 4, pp. 16–23, Oct. 2004.

[2] S. George, et al., “Distressnet: A wireless ad hoc and sensor network architecture for situation management in disaster response,” IEEE Commun. Mag., vol. 48, no. 3, pp. 128–136, Mar. 2010.

[3] G. Zussman and A. Segall, “Energy efficient routing in ad hoc disaster recovery networks,” in Proc. IEEE INFOCOM, 2003, pp. 682–691.

[4] A. Fujihara andH.Miwa, “Disaster evacuation guidance using opportunistic communication: The potential for opportunity-based service,” in BigData Int Things A Roadmap Smart Environ., 2014, pp. 425–446.

[5] “Powerful quake ravages china, killing thousands,” (2008). [Online]. Available: http://www.nytimes.com/2008/05/13/world/asia/ 13china.html

[6] X. Sun, Z. Lu, W. Hu, and G. Cao, “Symdetector: Detecting soundrelated respiratory symptoms using smartphones,” in Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput., 2015, pp. 97–108.

[7] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, “Medusa: A programming framework for crowd-sensing applications,” in Proc. 10th Int. Conf. Mobile Syst. Appl. Services, 2012, pp. 337–350.

[8] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, “Serendipity: Enabling remote computing among intermittently connected mobile devices,” in Proc. 13th ACM Int. Symp. Mobile Ad Hoc Netw. Comput., 2012, pp. 145–154.

[9] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, “E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures,” in Proc. 20th Annu. Int. Conf. Mobile Comput. Netw., 2014, pp. 617–628.

[10] N. Suzuki, J. L. F. Zamora, S. Kashihara, and S. Yamaguchi, “SOSCast: Location estimation of immobilized persons through SOS message propagation,” in Proc. 4th Int. Conf. Intell. Netw. Collaborative Syst., 2012, pp. 428–435.

[11] H. Nishiyama, M. Ito, and N. Kato, “Relay-by-smartphone: Realizing multihop device-to-device communications,” IEEE Commun. Mag. vol. 52, no. 4, pp. 56–65, Apr. 2014.

[12] M. Ervasti, S. Dashti, J. Reilly, J. D. Bray, A. Bayen, and S. Glaser, “iShake: Mobile phones as seismic sensors – user study findings,” in Proc. 10th Int. Conf. Mobile Ubiquitous Multimedia, 2011, pp. 43–52.

[13] Z. Lu, X. Sun, Y. Wen, and G. Cao, “Skeleton construction in mobile social networks: Algorithms and applications,” in Proc. IEEE 11th Annu. Int. Conf. Sens. Commun. Netw., 2014, pp. 477–485.

[14] Z. Lu, X. Sun, Y. Wen, G. Cao, and T. La Porta, “Algorithms and applications for community detection in weighted networks,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 11, pp. 2916–2926, Nov. 2015.

[15] B. Wang, “Coverage problems in sensor networks: A survey,” ACM Comput. Surveys, vol. 43, no. 4, pp. 32:1–32:53, 2011.