# A Query Based Trust Evaluation Scheme for Emergency Response Communication Networks

Muhammad Ibrahim Channa
Asian Institute of Technology
Thailand

Samad Baseer
Asian Institute of Technology
Thailand

Kazi M. Ahmed
Asian Institute of Technology
Thailand

## ABSTRACT
The natural or man-made disaster demands an efficient communication and coordination among first responders to save life and other community resources. This requires the generation and exchange of current information among first responders and emergency management centers in real time for making life saving decisions. Normally, the traditional communication infrastructures such as landline or cellular networks are damaged and don't provide adequate services to first responders for exchanging emergency related information. Mobile ad hoc networks are commonly used as communication means during emergency response operations. Reliable and robust communication is vital for efficient emergency response operations. In large scale emergency response, various rescue teams from different rescue organizations participate for controlling the emergency situation. As the nodes from different rescue organizations join the same emergency response network, there is a possibility that some of the nodes may demonstrate selfish or malicious behavior. A node may experience some damage during emergency management operations that prevents it from forwarding the packets successfully. The communication interruptions among first responders and emergency management centers result into mismanagement of emergency response efforts causing more loss of human lives and other community resources. We propose a query based trust evaluation scheme for ad hoc emergency response networks that keeps track of faulty, selfish and malicious nodes. This information may be used by routing protocols to isolate faulty, selfish and malicious nodes during route setup process for providing reliable and robust emergency response communication services.

## General Terms
Trust Management.

## Keywords
Emergency response, Trust evaluation, Mobile ad hoc networks, Faulty nodes, Selfish nodes, Malicious nodes, Reliability.

## 1. INTRODUCTION
It is a great challenge for public emergency services to cope with the crisis situations that arise due to natural or man-made disasters, such as earthquakes, floods and nuclear explosions. An optimal provision of relevant information concerning the emergency situation is essential for coping with such disasters in an effective and coordinated manner [1][2][3][4]. As coordination requires current information within and among various rescue organizations in real time, it is necessary to deploy an integrated information and communication system for disaster management, which provides an efficient, reliable and secure exchange of relevant information [5]. Depending upon the intensity and coverage area of a disaster, it might be a multi-organizational operation involving government authorities, public authorities, volunteer organizations and the media. These entities work together as a virtual team to save lives, and other community resources [6].

In emergency situations, the availability of telecommunication services is of great importance. These systems provide a means of communication among first responders, affected people and emergency management centers. During Hurricane Katrina [7][8], several wireless base stations were taken out and various communication cables were damaged. The remaining parts of the network were not able to provide adequate communication services to first responders [9]. Mobile ad hoc networks [10][11] are commonly used as ad hoc communication networks in emergency response operations. These networks are easily deployed without need of any existing telecommunication infrastructures. These networks automatically configure when new nodes join or leave the network dynamically.

Reliable and robust communication is vital for successful emergency response operations [9]. During emergency situations, the deployed ad hoc communication network might itself be prone to faults and vulnerable to security threats [6]. In large scale emergency response operations, the participating nodes belong to several rescue teams and there is a possibility that some of the nodes may demonstrate selfish or malicious behavior. A selfish node may drop all received packets for saving its energy and bandwidth resources. A malicious node may modify received packets for corrupting their information. A node may also experience some fault during field level emergency management operations and drop random number of received packets. These node behaviors may affect the reliability of emergency response communication networks.

In ad hoc networks, *trust* is a belief level of a node on another node for a specific action based on its direct and/or indirect observations on the behavior of the observed node. The nodes evaluate trust of each other and form trust relationships with each other. In wired-networks, trust is usually achieved through indirect trust mechanisms such as digital certificates issued by certification authorities and authentication servers. This is not true for ad hoc networks as there is no direct access to certification authorities all the times. These networks are based on "trust-your-neighbor" relationships, which originate, develop and expire on the fly [12]. We propose a query based trust evaluation scheme that keeps track of faulty, selfish and

malicious nodes in emergency response networks. This helps the routing protocols to isolate these nodes during route setup process and improve the reliability of emergency response networks.

The rest of the paper is organized as follows. Section 2 presents an overview of the proposed query based trust evaluation scheme. The components of the proposed scheme are described in section 3 and section 4 concludes the paper.

## 2. PROPOSED TRUST EVALUATION SCHEME

The proposed query based trust evaluation scheme comprises of three major components namely the Observer, Evaluator and Trust Database as shown in figure 1. The *Observer* is responsible for observing the behavior of neighbor nodes by exchanging request and response messages. The *Evaluator* evaluates the trust category of neighbor nodes based on the information provided by the Observer. The *Trust Database* stores the trust information about neighbor nodes. The system works as follows. When a new node joins the network or changes its position to have new neighbors, its neighbors believe it to be trustworthy until observed by them. When a node sends some packets to its neighbor node, it sends a request message to its neighbor for the forwarding information of the sent packets. The requested node broadcasts the response message holding the forwarding information of the sent packets along with the IDs of the requesting node and its next hop, where the packets have been forwarded by the requested node. The response message is only processed by the requesting node and the next hop of the requested node. All other nodes ignore the response message. This information is then processed by the evaluators at the requesting node and the next hop of the requested node to identify the trust level of the requested node. The trust information is then stored in the Trust Database and is used by the routing protocols for isolating faulty, selfish and malicious nodes during route setup process for reliable and secure emergency response communications.

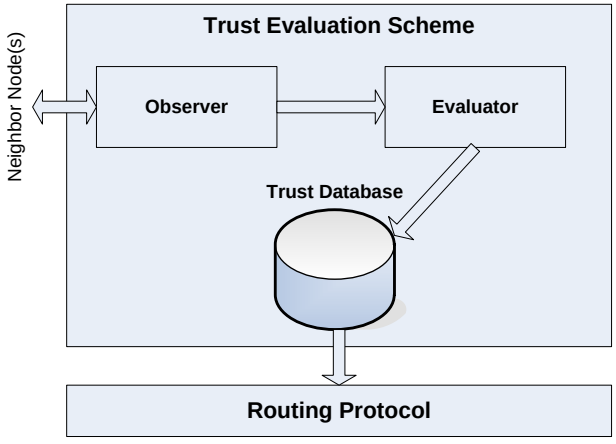


**Figure 1: Query based trust evaluation scheme**

## 3. COMPONENTS OF THE PROPOSED SCHEME

This section describes the functionality of various components of the proposed query based trust evaluation scheme.

### 3.1 Observer

As described earlier, the Observer is responsible for monitoring neighbor nodes in terms of their packet forwarding behavior by exchanging request and response messages. We extend the WatchAnt [13] scheme for identifying faulty, selfish and malicious nodes. Each Observer maintains two types of buffers namely the send buffer and the receive buffer. The send buffer stores the information about the packets sent to the next hop and the receive buffer contains the information about the packets received from the previous hop. The information maintained by the send buffer includes the receiver ID, sent packet ID and hash of the immutable fields of the sent packet. Similarly, the information maintained by the receive buffer includes sender ID, received packet ID and hash of the immutable fields of the received packet. The structure of the send and receive buffers for nodes *i*, *j* and *k* is shown in figure 2. The length of the send and receive buffers may be limited to certain number of entries.

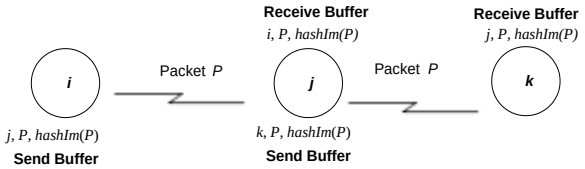


**Figure 2: Send and receive buffers**

The Observer uses a request/response mechanism for identifying the behavior of the neighbor nodes. The *request message* consists of the requested node ID and the IDs of the most recently sent packets whose forwarding information is desired. The *response message* comprises of the requesting node ID, ID of the node where the packets have been forwarded by the requested node and the packet IDs and hash values of immutable fields of the forwarded packets. Referring to figure 2, if node *i* sends a packet *P* to node *j* and asks it for forwarding information of packet *P*, then the format of the request and response messages is as follows.

**Request** ($j$, $P$)

**Response** ($i$, $k$, [$P$, $hashIm(P)$]) (1)

If the forwarding information of $n$ packets is desired, the format of request and response messages is as follows.

**Request** ($j$, [$P_0$, $P_1$,.....$P_n$])

**Response** ($i$, $k$, [$P_0$, $hashIm(P_0)$], [$P_1$, $hashIm(P_1)$],…., [$P_n$, $hasIm(P_n)$]) (2)

The response message is processed by the evaluators at the requesting node and the next hop of the requested node. This helps in evaluating the true behavior of the requested node by at least one of its neighbor nodes.

## 3.2 Evaluator

The Evaluator categorizes a node into one of the four categories such as reliable node, faulty node, selfish node and malicious node. A reliable node successfully forwards all received packets to its next hop. A faulty node may experience some hardware/software problems and may not forward all received packets successfully. A selfish node drops all received packets to save its battery life and a malicious node tempers with the contents of the received packets. It is also assumed that there is no congestion or relevant channel problems in the network. We consider the scenario described in section 3.1, where node *i* is the requesting node, node *j* is the requested node and node *k* is the next hop of the requested node *j*, where node *j* has forwarded the packets received from node *i*. The Evaluator at the requesting node *i* manipulates the information from request message, response message and its send buffer to evaluate the trust level of the requested node *j*. Similarly, the Evaluator at the next hop *k* processes the information from the response message and its receive buffer to evaluate the trust level of the requested node *j*. We describe the trust evaluation process of the Evaluator at the requesting node *i* and next hop *k* for requested node *j* as follows.

If node *i* receives the response message from node *j*, the Evaluator at node *i* compares the packet IDs and their hash values in the request message and send buffer with corresponding entries in the response message. If there is a match, node *i* treats node *j* as a reliable node. Similarly, the Evaluator at node *k* compares the packet IDs and their hash values in the response message with corresponding entries in its receive buffer. If there is a match, node *k* also assumes node *j* as a reliable node.

If node *i* receives the response message from node *j*, the packet IDs in the request message are considerably greater than the same in the response message and the hash values of the packets in the response message and their corresponding entries in the send buffer matches, node *i* assumes node *j* as a faulty node as node *j* might have dropped several received packets. Node *k* treats node *j* as a reliable node if the number of packet IDs and their corresponding hash values in the response message matches it's receive buffer.

If node *i* does not receive the response message from node *j*, it assumes that node *j* being a selfish node might have dropped the request message along with other sent packets. Node *k* does not perform any action if it does not receive the response message from node *j*.

If node *j* tempers with the packet contents and generates a fake response message indicating successful delivery of the packets to node *k*, it can deceive node *i* but not node *k*. Node *i* may assume node *j* as a reliable node but node *k* can identify the malicious behavior of node *j* by comparing the packet IDs and their hash values in the response message with corresponding entries in its receive buffer. Node *k* identifies that there is no match in the packet IDs and their hash values in the response message and its receive buffer and assumes node *j* as a malicious node. The Evaluator performs several observations on the packet forwarding behavior of neighbor nodes and aggregates them to identify true behavior of the neighbor nodes. The functionality of the Evaluator at the requesting node *i* and the next hop *k* is described in figure 3 as follows.

**1: if** (Node *i* receives response message from node *j*) **then**
**2:** **if** (Packet IDs in request message match with corresponding
**3:** entries in response message) && (Hash values of packets in
**4:** send buffer match with corresponding entries in response
**5:** message)
**6:** **then**
**7:** Node *i* assumes node *j* as *reliable*
**8:** **endif**
**9:** **if** (Packet IDs in request message considerably greater than
**10:** the same in response message) && (Hash values of packets
**11:** in send buffer match with corresponding entries in response **12:** message)
**13:** **then**
**14:** Node *i* assumes node *j* as *faulty*
**15:** **endif**
**16: else**
**17:** Node *i* assumes node *j* as *selfish*
**18: endif**

**(a) Evaluator at node *i***

**1: if** (Node *k* receives response message from node *j*) **then**
**2:** **if** (Packet IDs in response message match with corresponding **3:** entries in receive buffer) && (Hash values of packets in
**4:** response message match with corresponding entries in
**5:** receive buffer) **then**
**6:** Node *k* assumes node *j* as *reliable*
**7:** **else**
**8:** Node *k* assumes node *j* as *malicious*
**9:** **endif**
**10: endif**

**(b) Evaluator at node *k***

**Figure 3: Trust evaluation process of Evaluator**

## 3.3 Trust Database

The trust database stores the trust information about neighbor nodes. The trust information comprises of the node ID flowed by its trust category as shown in figure 4. The trust categories of the nodes may be represented by values such as *r, f, s* and *m* for reliable, faulty, selfish and malicious nodes respectively.

| ID | Tc |
|---|---|
| | |

**Figure 4: Trust database**

## 4. CONCLUSION

A query based trust evaluation scheme has been proposed for ad hoc emergency response networks that keeps track of faulty, selfish and malicious nodes. The proposed scheme is independent of any specific ad hoc routing protocol, uses simple computations for trust evaluation, consumes negligible storage for storing trust information and puts reasonable burden on channel bandwidth while exchanging request and response messages randomly. The proposed scheme evaluates the trust of its neighbor nodes on direct observations only. We are in the process of implementing and simulating the performance of the proposed scheme under different scenarios. It is also desired to

extend this scheme in such a way that the trust of a node can be evaluated by direct observations and recommendations from other network nodes.

## 5. REFERENCES

[1] Auf der Heide, E. 1989. Disaster Response: Principles of Preparation and Coordination. Online Book: http://coe-dmha.org/dr.

[2] Kyng M., Nielsen E. and Kristensen M. 2006. Challenges in Designing Interactive Systems for Emergency Response. Proceedings of the 6th Conference on Designing Interactive Systems, 301-310.

[3] Risse T. and Kirchner H. 2006. Challenges in Information Systems for Disaster Recovery and Response. Proceedings of the 3rd GI/ITG KuVS Fachges Prah, Berlin, 16-19.

[4] Manoj B. and Baker A. 2007. Communication Challenges in Emergency Response. Communications of the ACM, Vol. 50, Iusse 3, 51-53.

[5] Meissner A., Luckenbach T., Risse T., Kirste T. and Kirchner H. 2002. Design Challenges of an Integrated Disaster Management Communication and Information System. The First IEEE Workshop on Disaster Recovery Networks (DIREN 2002).

[6] Mehrotra S., Znati T. and Thompson C. 2008. Crisis Management. IEEE Internet Computing, Vol. 12, Issue 1, 14-17.

[7] Lueck T. 2005. Grant to Help City Broaden Radio Network. New York Times, September 19, 2005.

[8] Thompson C. 2005. Talking in the Dark. New York Times Magazine, September 18, 2005.

[9] Portman M. and Pirzada A. 2008. Wireless Mesh Networks for Public Safety and Crisis Management Applications. IEEE Internet Computing, Vol. 12, Issue 1, 18-25.

[10] Hubaux J., Gross T., Boudec J. and Vetterli M. 2001. Towards Self Organized Mobile Ad Hoc Networks: the Terminodes Project. IEEE Communications Magazine, Vol. 39, Issue 1, 118-124.

[11] Mobile Ad-hoc Networks (manet) WG, Mobile Ad-hoc Networks (manet) Charter, Wg Charter, IETF. 1999. http://www.ietf.org/html.charters/manet-charter.html.

[12] Pirzada A. and McDonald C. 2004. Establishing Trust in Pure Ad Hoc Networks. Proceedings of the 27th Australian Conference on Computer Science, 47-54.

[13] Mogre, P. Graffi, K. Hollick, M. and Steinmetz, R. 2007. AntSec, WatchAnt and AntRep: Innovative Security Mechanisms for Wireless Mesh Networks. Proceedings of the 32nd IEEE Conference on Local Computer Networks, 539-547.