Implementation and Benchmarking of a Novel Routing Protocol for Tactical Mobile Ad-Hoc Networks

Dhafer Ben Arbia, Muhammad Mahtab Alam, Abdullah Kadri, Rabah Attia, Elyes Ben Hamida

Implementation and Benchmarking of a Novel Routing Protocol for Tactical Mobile Ad-Hoc Networks

Dhafer Ben Arbia*†, Muhammad Mahtab Alam*, Abdullah Kadri*, Rabah Attia†, Elyes Ben Hamida*
*Qatar Mobility Innovations Center (QMIC), Qatar University,
PO. Box. 210531, Doha, Qatar. Email: [dhafera, mahtaba, abdullahk, elyesb]@qmic.com, rabah.attia@enit.rnu.tn
†SERCOM Lab, Polytechnic School of Tunisia, University of Carthage B.P. 743. 2078 La Marsa. Tunisia

***Abstract*—Providing efficient routing service over tactical multi-hop ad-hoc networks is a crucial building block in wireless communication networks especially during a disaster relief. To date, there is still a lack of routing standards for such networks. Indeed, in such harsh environment, medical rescue teams, firefighters, military, police and even victims need to be steadily connected to a distant command center (CC) which conducts the rescue operations. In this paper, we propose a new multi-hop routing approach called ORACE-Net. The proposed protocol uses advertisement packets to establish routes from deployed nodes towards the CC (i.e. Direct Route Establishment). Then, it utilizes the data packets to establish reverse routes (from the CC to all nodes in the network). We implemented and evaluated our approach in realistic scenario using tactical and on-body mobile nodes. Our experiments include also an Internet of Thing (IoT) platform and a real-time dynamic topology website which are used for analyzing the behavior of the protocol. The experimental results show that our protocol increases the mobile nodes connectivity and packet delivery rate. Also, it reduces the average round trip time delay for the on-body nodes compared to the tactical deployed base stations.**

***Index Terms*—Tactical Multi-Hop Routing Protocol, Ad-hoc Wifi, Internet of Things, Optimized Routing Approach for Critical and Emergency Networks.**

I. Introduction

Disasters are increasing worldwide with more devastating effects than ever before. In fact, the absolute number of disasters around the world has almost doubled since the 1980s [1]. The growing number of disasters and accidents has a significant impact on humans living conditions, asset safety, as well as the economy. According to the United Nations Development Program (UNDP) Office for Disaster Risk Reduction (UNISDR), during the last decade (between 2000-2012) the overall worldwide estimated damages due to various disasters results in loss of 1.7 Trillion dollars, 1.2 Million loss of lives and overall 2.9 Million affected people [2].

Information and communication technologies (ICT) has a vital role both in early prediction as well as in effective rescue and evacuation phase to minimize the loss of lives and assets. On one hand, the public safety networks (PSNs) are evolving rapidly from the classical land mobile radio systems to the long term evolution (LTE) technologies. On the other hand, there is a growing need for reliable ubiquitous communication system which should be fast and easy to deploy.

In this context, instant mobile ad-hoc networking is emerging as an effective alternative to be deployed during the disaster [3], [4]. In addition to radio technologies inter-operability, coexistence, and energy consumption issues, routing is an important and critical challenge for the tactical emergency networks [5]–[7]. To the best of our knowledge, the existing routing techniques and protocols for PSN are not optimized for the tactical and disaster context.

In this paper, we introduce a tactical and mobile multi-hop routing protocol called Optimized Routing Approach for Critical and Emergency Networks (ORACE-Net). The protocol enables two-way communication to-and-from command center (CC) to the rest nodes in the network. Please note that the CC may include one or more *CC nodes*.

Specific contributions of this paper are listed as follows: 1) We present ORACE-Net approach which is based on the End-to-End Link Quality Estimation ($E2E_{LQE}$) to optimize routing in tactical networks. 2) We discuss the experimental results of the proposed approach implementation in a realistic tactical testbed. 3) We used an Internet of Things platform in the application layer to evaluate the network connectivity and reliability.

II. Related Works

The development of ad-hoc networking via WiFi IEEE 802.11n standard on smart phones is rare. In wireless ad-hoc mode, each device can directly communicate with another device and the management of the ad-hoc network is done through collaboration between the nodes in the network. The authors in [8] present similar networking over WiFi Direct standard on the smart phones. The proposed architecture enables building multiple group formations using multi channels on Wi-Fi Direct for providing multi-hop communications which help to achieve power saving for energy constraint smart phones. This content-centric multi-hop networking study is limited to only energy optimization. Other performance metrics (i.e., communication delay, hop-counts and packet reception rate) are not targeted. Further, similar concept is exploited in [9], where authors propose an energy efficient cluster-based routing protocol (called QGRP), which relies on virtual hierarchical distributed clustered algorithm. The simulation results show that the proposed QGRP is energy effi-

cient. However, the study presents only preliminary simulation results and it misses detailed study including implementation.

The design and implementation of the wireless multi-hop ad-hoc networks using smart phones is presented in [4]. The middle-ware of the mobile Android devices is implemented in the user-space and therefore, kernel modifications are not required. The implementation evaluation results show that the middle-ware achieves around 5 Mbps transmission bandwidth in single-hop and around 4 Mbps bandwidth in two hops communications with strong WiFi links. The experiments were carried out with forced multi-hop setup. Almost 100 ms average round trip delay time is reported. However, the experiments lack real multi-hop communication as it is only limited to small-scale networks with the maximum of two-hops. The Optimized Link State Routing Protocol (OLSR) is implemented in [10]. The implementation includes two android (HTC) smart phones and three PCs. The results show that the OLSR protocol always ensures strong connectivity under dynamic mobility. The implementation study is interesting as it validates the connectivity of the mobile network in heterogeneous devices. However, the performance metrics and their analysis of the protocol at network level are not addressed.

The preliminary results of the prototype implementation on Android smart phone over WiFi using software defined networking (SDN) are presented in [11]. Text messaging and file transfer applications were demonstrated for SDN in ad-hoc networks (SDNAN). This approach is based on three layers i.e., ad-hoc networking layer (which is based on AODV routing protocol), a network operating system layer (which controls the dynamic variations of the routing protocol) and control program (which manages forwarding rules, routing table, and routing protocol on the fly). Android Interface Definition Language (AIDL) is used for inter-process communication which simplifies the code complexity of the interface among three layers.

Some more recent studies (such as [3], [9], [12]–[14]) are adapted to the network dynamic aspects for viable and effective disaster operation. For example, localization-based and network congestion adaptive approach called "DistressNet" [13] is efficient in congestion avoidance in the network during disaster operations. However, this approach creates network sparseness which impacts localization and renders multi-hop algorithms inefficient, especially in the indoor rescue operations. Recent proposed approaches called Reliable Routing Technique (RRT) [3] and TeamPhone [14], are based on Greedy Perimeter Stateless Routing (GPSR) [15] and Delay Tolerant Network version-2 (DTN2) [12] respectively. The authors claim that RRT approach is delay-efficient. However, it has significantly higher energy consumption. On the other hand, TeamPhone is limited in terms of services. In fact, it is based on ad-hoc and opportunistic network and it provides only basic emergency messages with up to one hop only.

To conclude, wireless mobile ad-hoc networking using smart phones is appearing to dominate future rescue and critical operations. In most of the above mentioned implementations and prototypes, network connectivity and mobility under indoor rescue environment are not considered. Additionally, the number of nodes used is too low (around three nodes). To address some of these limitations, in this paper, we consider heterogeneous architecture including smart phones (Android-based), raspberry pi (Linux-based) devices for effective connectivity in ad-hoc tactical network. We mainly considered above protocols in the literature, because they were implemented or evaluated in realistic scenario. Other approaches such as [16] were not covered because they are multi-path and will be a part of our future implementations..

III. Multi-hop Optimized Routing Approach for Critical and Emergency Networks

In this section, we present a new routing protocol called Optimized Routing Approach for Critical and Emergency Networks (ORACE-Net). The main objective of ORACE-Net is to have instant neighborhood links visibility and establish available optimized routes according to the specific link quality estimation metrics. The proposed protocol consists of three main phases: 1) Beacons, Advertisement broadcasts and Link Quality Estimation, 2) Direct Route Establishment (DRE), and 3) Reverse Route Establishment (RRE). These phases are described in the following subsections.

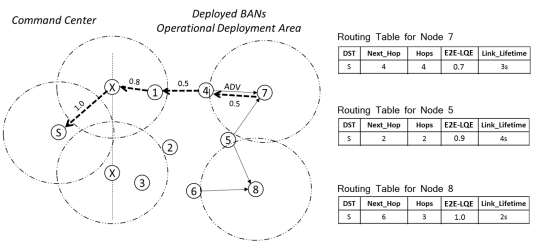
1) Beacons, Advertisement broadcasts and Link Quality Estimation: Each node from ORACE-Net network broadcasts continuously periodic $Hello$ packets for neighborhood discovery according to the standard NeighborHood Discovery Protocol (NHDP) [17]. In addition, $Hello$ packets are used in the link quality estimation for the nodes [18]. Each $Hello$ packet has a sequence number. When a node receives the first $Hello$ packet from a neighbor, this neighbor is inserted into the neighbors table with a Link Quality Estimation (LQE) equal to 1.0. Based on the $Hello$ packets broadcasted every 3s, a node can estimate the number of $Hello$ packets supposed to be received during a certain period of time. The LQE of a one hop neighbor is assigned according to the following equation:

$$LQE = \frac{H_R}{H_E} \tag{1}$$

where H_R is the number of Received $Hello$ packets, and H_E is the expected number of $Hello$ packets to be received which is equal to:

$$H_E = \frac{T_C - T_S}{P_H} \tag{2}$$

where T_C is the current time, T_S is the connection starting time with each specific node, P_H is the $Hello$ period. The *CC node* initializes the connection by broadcasting periodically Advertisement packets (ADV) which are flooded over the entire network as a wave to introduce the *CC node* to all other nodes in the network. A *CC node* is a node deployed by the command center in the closest safe place to the incident area. A node receiving an ADV, processes it and then rebroadcasts it to all of its reachable nodes. The header of the ADV contains a sequence number which is used to discard the duplicated received ADVs. When a node receives an ADV, a route is established towards the *CC node* with the last visited node by the ADV selected as the next-hop. The following



Routing Table for Node 7

DST	Next_Hop	Hops	E2E-LQE	Link_Lifetime
S	4	4	0.7	3s

Routing Table for Node 5

DST	Next_Hop	Hops	E2E-LQE	Link_Lifetime
S	2	2	0.9	4s

Routing Table for Node 8

DST	Next_Hop	Hops	E2E-LQE	Link_Lifetime
S	6	3	1.0	2s

Fig. 1. Routing tables after DRE phase (for Nodes 7, 5 and 8) when the 1^{st} wave of ADV reaches all nodes. Please note that, Nodes Xs are base stations deployed by the rescue teams while they are moving towards the incident area. Route from Node 7 to the *CC node* is represented by the bold dashed line.

received ADV will initiate the second phase of ORACE-Net detailed in the next subsection.

In our proposed approach, ADV broadcasting process has three key roles: 1) it contributes in the conventional neighbors discovery process, 2) it provides routes establishment towards the command center node(s) (*CC node(s)*), 3) it provides also the $E2E_{LQE}$. The proposed approach relies on two main metrics: $E2E_{LQE}$ and the Hop_{Count}. The first metric can be calculated based either on the *Signal Strength Level* (SSL), the *link quality indicator* (LQI), or the *signal to noise ratio* (SNR) measurements [19] [18]. To that end, each ADV contains specific header's entries to track the hop count and the $E2E_{LQE}$ along the traversed route. When an ADV is rebroadcasted, the $E2E_{LQE}$ field in the packet header is updated by multiplying the LQE values recorded at each hop. Figure 1 depicts an example of the ADV broadcasting process. The $E2E^{SD}_{LQE}$ between a source node S and a destination node D is calculated according to the following equation:

$$E2E^{SD}_{LQE} = \prod_{S}^{D} LQE_{ij} \tag{3}$$

where: S is the source of the E2E route, D is the destination, i and j are the visited nodes from the source to the destination. LQE_{ij} is the Link Quality Estimation between node i and j (i.e., on one hop only).

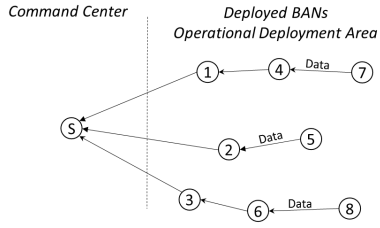
The proposed ORACE-Net routing protocol operates based on two different algorithms. Algorithm 1 is run upon the reception of ADVs (i.e. DRE), while Algorithm 2 is executed upon receiving a $DATA$ packet (i.e. RRE).

2) Direct Route Establishment: The *CC node* broadcasts an ADV, then it waits for a predefined period (i.e. 3s) to broadcast the next ADV with a new sequence number. On the other hand, when a node receives an ADV, it updates both its neighbors and routing tables. Then it rebroadcasts the ADV only once. The direct route establishment algorithm (as explained below) is depicted in Algorithm 1.

The routing table is only updated when the received ADV has better $E2E_{LQE}$ than the one of the current used route. According to Algorithm 1, all the ADV packets are considered (even duplicated), but each ADV is re-broadcasted only one time (based on the sequence number). Indeed, the $E2E_{LQE}(ADV)$ is compared with the $E2E_{LQE}(Route)$. If the first value is higher, then the current route is updated as

Algorithm 1 Direct Route Establishment Algorithm (Node 'i')

1- RX (SRC, DST, Sender, ADV_{Packet})
2- Update_Neighbors_Table(ADV_{Packet})
if $(E2E_{LQE}(ADV_{Packet}) > E2E_{LQE}(Route))$ OR $(E2E_{LQE}(ADV_{Packet}) == E2E_{LQE}(Route)$ AND $Hop_{Count}(ADV_{Packet}) \leq Hop_{Count}(Route))$ **then**
  Update_$E2E_{LQE}(ADV_{Packet})$
  Update_$Hop_{Count}(ADV_{Packet})$
  Update_$Routing_{Table}(ADV_{Packet})$
end if
if ($ADV_{Packet}(Seq_{Number})$ == already_broadcasted) **then**
  3- Drop_Duplicated_$ADV_{Packet}(Seq_{Number})$
  4- Go To 1.
else
  5- TX (SRC=CC-node, DST=Bcast, Sender=i, ADV_{Packet});
  6- Go To 1.
end if{Where: "SRC" is the originator of the packet, and "Sender" is the last visited node.}



Routing Table for CC-Node S

DST	Next_Hop	Hops	E2E_LQE	Link_Lifetime
1	1	1	0.9	2s
2	2	1	1.0	1s
3	3	1	1.0	5s
4	1	2	0.95	2s
5	2	2	1.0	1s
6	3	2	0.9	4s
7	1	3	0.92	1s
8	3	3	0.95	3s

Routing Table for Node 4

DST	Next_Hop	Hops	E2E_LQE	Link_Lifetime
S	1	2	0.9	3s
7	7	1	1.0	3s

Routing Table for Node 2

DST	Next_Hop	Hops	E2E_LQE	Link_Lifetime
S	S	1	1.0	4s
5	5	1	1.0	3s

Fig. 2. Reverse Route Establishment (i.e., RRE) based on data packets.

follows: First, the last visited node by the ADV becomes the next-hop of the route. Second, the $Hop_{Count}(Route)$ gets the value of the $Hop_{Count}(ADV)$, and the destination remains always the *CC node*. If the $E2E_{LQE}(ADV)$ is equal to the $E2E_{LQE}(Route)$, then, the shortest or equal path is considered. For the rest of the cases, the current route is maintained until the route lifetime expires. If it is the case, a new route is created based on the next first ADV received. It is important to note here, that as soon as an ADV is re-broadcasted, the upcoming received ADV (with delay) with the same sequence number are then dropped (Step 3 of Algorithm 1). This feature will trigger the multi-path functionality in the upcoming versions of ORACE-Net. The DRE phase of the protocol ends up by a fresh route towards the *CC node* at every involved node with only one way routes (i.e., from nodes to the *CC node*) as depicted in Figure 1. As a reply to the ADV packets, nodes send back a data packet towards the originator *CC node*, this data packet triggers the next phase called Reverse Route Establishment (i.e., RRE).

Algorithm 2 Reverse Route Establishment Algorithm (Node 'j')

1- RX (SRC, DST, Sender, $DATA_{Packet}$)
2- Update_Neighbors_Table($DATA_{Packet}$)
if $(E2E_{LQE}(DATA_{Packet}) > E2E_{LQE}(Route))$ OR $(E2E_{LQE}(DATA_{Packet}) == E2E_{LQE}(Route)$ AND $Hop_{Count}(DATA_{Packet}) \leq Hop_{Count}(Route))$ **then**
  Update_$E2E_{LQE}(DATA_{Packet})$
  Update_$Hop_{Count}(DATA_{Packet})$
  Update_$Routing_{Table}(DATA_{Packet})$
end if
if ($DATA_{Packet}(Seq_{Number})$ == already_broadcasted) **then**
  3- Drop_Duplicated_$DATA_{Packet}(Seq_{Number})$
  4- Go To 1.
else
  5- TX (SRC, DST=CC_node, Sender=j, To_Nexthop, $DATA_{Packet}$)
  6- Go To 1
end if

3) Reverse Route Establishment: ORACE-Net proposes bi-directional path establishment for efficient routing in public protection and disaster networks. Indeed, the data packets are forwarded hop-by-hop until they reach the *CC node*. The $DATA$ packet header records the $E2E_{LQE}$, Hop_{Count}, the last visited node, and the originator of the packet. If the routing table does not contain a route to the originator of the packet, then a new route is created. Otherwise, if the route already exists and the $E2E_{LQE}(DATA)$ is higher than $E2E_{LQE}(Route)$, or, they are equal and the $Hop_{Count}(DATA)$ is less or equal than $Hop_{Count}(Route)$, fields are extracted from the header to create or update route as follows: 1) The originator of the $DATA$ packet becomes the final destination in this route. 2) The last visited node is the next-hop to reach that final destination. 3) The $E2E_{LQE}$ is updated with the LQE (given by Equation 2) of the link (Current node, last visited node) according to Equation 3, then inserted into the route. 4) The Hop_{Count} is incremented and inserted within the route. Figure 2 illustrates the data packets flow towards the *CC node*. When a node receives a data packet, the node updates its routing table then forwards the packet. A duplicated $DATA$ packet is used to update the routing table and then dropped based on the sequence number. Similarly to the DRE phase, the $E2E_{LQE}$ is calculated and updated using the same process as detailed in Algorithm 2.

IV. Implementation Architecture

In this section, we describe ORACE-Net implementation. Two versions of ORACE-Net are implemented (i.e. Linux and Android). Linux version is developed in C-language whereas Android version is Java-based implementation. A system-oriented stack is depicted in Figure 3a. The stack presents the common system layers (i.e., Hardware, Linux Kernel) and the specific layers for each architecture (i.e., libraries, runtime, environment and user interface).

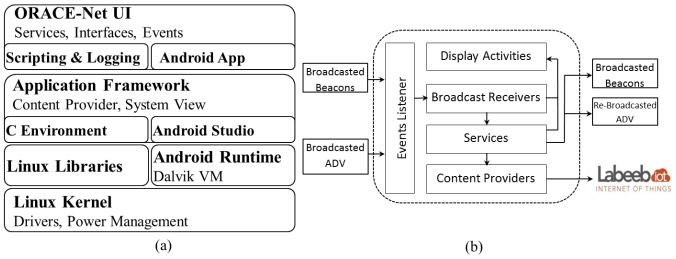
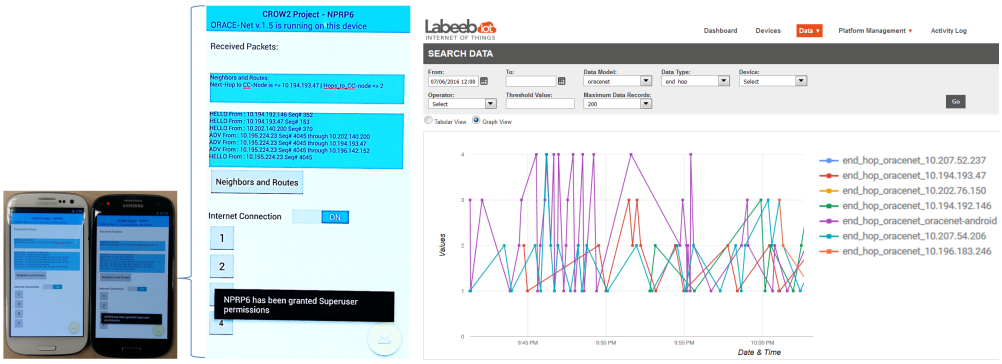


Fig. 3. (a) ORACE-Net System-Oriented Stack over Linux and Android. (b) ORACE-Net Android Application Architecture

With regards to the operational requirements of a disaster context, first rescue teams reaching the incident area, deploy tactical base stations to initiate a wireless tactical network. Each rescuer is equipped with a mobile device. Consequently, an ad-hoc tactical-mobile network will cover the disaster area. All devices are running according to ORACE-Net over wireless ad-hoc network. In the following subsections we describe the different components and devices of the experiments: 1) mobile devices, 2) tactical base stations, 3) Labeeb-IoT platform, and 4) dynamic topology website.

A. Mobile Devices : ORACE-Net Android application

ORACE-Net Android application is implemented on the user's level as depicted in Figure 3a, it exploits the features of Linux operating system at the kernel layer through the Dalvik Virtual Machine. Figure 3b depicts ORACE-Net mobile application components which are: 1) *events listener*, 2) *Broadcast receivers*, 3) *Services*, 4) *Content Providers*, and 5) *Display activities*. The relevant component in the architecture is the *events listener* which triggers the rest of the tasks. An *events listener* is used to catch events (e.g. multicasted or broadcasted packets, clicked button, typed text, etc.). In ORACE-Net java-based Android application, the *events listener* is implemented as a socket with a multicast IP address/Port: *224.0.0.1/10000*. Similar socket is implemented on C-language under Linux for the tactical base stations. Received packets through the *events listener* are handled by the *broadcast receivers* component to be hulled. In fact, each field from the packet is retrieved separately: packet sequence number (Seq_{Number}) $Source$, $Destination$, Hop_{count}, and $E2E_{LQE}$. Non-duplicated received packets are passed to the *services* component to be exploited. The *services* block is also responsible for broadcasting $Hello$ and rebroadcasting the ADV packets. Algorithms 1 and 2 are implemented under the *services* component. Finally, the *content provider* allows the application share the results on other servers or platforms. Both outputs of ORACE-Net Tactical and Android applications are transmitted to an Internet of Things platform called: *Labeeb-IoT* [20]. Figure 4a shows the interface of the Android mobile application displaying the real-time events (i.e., received $Hello$ and ADV packets), the current route used by each device, and the Internet connection status.



(a) (b)

Fig. 4. (a) Testbed: A photo of the ORACE-Net mobile devices displaying the real-time events (received $Hello$ and ADV packets) and the current route. (b) Labeeb-IoT [20] interface (on the right) shows the variation of the Hop_{count} for the connected nodes(mobiles and statics) over the time.

B. Tactical Base Stations: ORACE-Net Linux application

For tactical base stations, we implemented ORACE-Net on Raspbian v8.0, a free operating system based on Debian optimized for the Raspberry Pi hardware. Linux libraries are used to operate the various protocol events (i.e., socket connections, packets encapsulation, multicasting and broadcasting). We use shell scripts to display the status and statistics and to manage the processes of the protocol. The logging system in the tactical devices is based on the operation system logging service *Syslog*. Finally, the data is pushed to *Labeeb-IoT* platform via the machine-to-machine (M2M) Message Queuing Telemetry Transport (MQTT) protocol client.

C. Labeeb-IoT Platform

Internet of Things (IoT) is emerging technologies developed for smart living solutions. IoT solutions are online platforms capable of sensing real-time data from diverse types of devices and sensors that could be deployed in a vast geographic area. This platform collects, stores and publishes the data according to many predefined parameters. With respect to the MQTT standard [21], *Labeeb-IoT* uses a publish/subscribe architecture in contrast with HTTP request/response paradigm architecture. Publish/Subscribe is event-driven and enables messages to be pushed to clients using MQTT protocol. MQTT client communicates with the broker using predefined methods (e.g., connect, disconnect, subscribe, publish). *Labeeb-IoT* offers various APIs and Restful/JSON web services.

In our experiments, ORACE-Net devices (mobile and tactical) push continuously and instantly specific data to *Labeeb-IoT* platform: 1) the device identifier ($Device_{Id}$), 2) the device location ($Location$), 3) the device neighbors list ($Neighbors$), 4) the next-hop to the *CC node* (NH_{CC}), 5) the $E2E_{LQE}$, and 6) the Hop_{count} to the *CC node*. This data is stored into the platform database and then extracted to the dynamic real-time topology website as shown in Figure 5.

D. Dynamic Topology Website

In order to display real-time network dynamic topology, we developed Javascript-based website. This tool reads instant data from *Labeeb-IoT* platform and displays nodes as spots linked to each others and to the *CC node* (a screenshot appears in the experimentation architecture at the right of. Figure 5). In addition to the network dynamic map, this tool displays the next-hop table for each connected node and the variation of the $E2E_{LQE}$. As an example, a snapshot of the ORACE-Net network is depicted in Figure 6. It shows that the mobile node is four hops far from the *CC node*, and the $E2E_{LQE}(M) = 0.78$.

V. Experimental Evaluation

In this section, we describe the settings of our testbed, then we present the adopted performance metrics, followed by the discussion of the obtained results.

A. Experimental Setup

In our experiments, we consider a disaster scenario in our office Qatar Mobility Innovations Center (QMIC) in Qatar Science and Technology Park (QSTP). Our testbed consists of eight raspberry-pi devices model 2-B and two Samsung galaxy S3-I9300 smart phones with ORACE-Net routing protocol implemented on-board. The office map is shown in Figure 5. Rescue teams access to the office from the back-gate (BG). First, they deploy the *CC node* in a trusted and safe location at the gate to be connected to Internet through an Ethernet or WiFi access point. Upon their entrance inside the office, rescuers start deploying tactical base stations in order to have the maximum network wireless coverage above all the operations area. Tactical static base stations are deployed as shown in Figure 5 from 1 to 8. Mobile nodes (smart phones) carried by the rescuers are connected through the tactical network to the *CC node*. Since the experimentation area is limited, we reduced the raspberry-pi's and smart phone's WiFi antennas transmission power to 0 dBm. Disaster area is divided into four zones: incident area (zones 1 and 2), victims waiting for evacuation (zone 3) and evacuation area (zone 4). Rescuers evacuate victims from incident area to the evacuation area. Experimentation parameters and configuration settings are detailed in Table I.

B. Experimentation Results

In this subsection, we present the results of experiments aimed to evaluate ORACE-Net tactical mobile routing protocol in real disaster scenario. In fact, with regards to the disaster context, we consider the following relevant performance

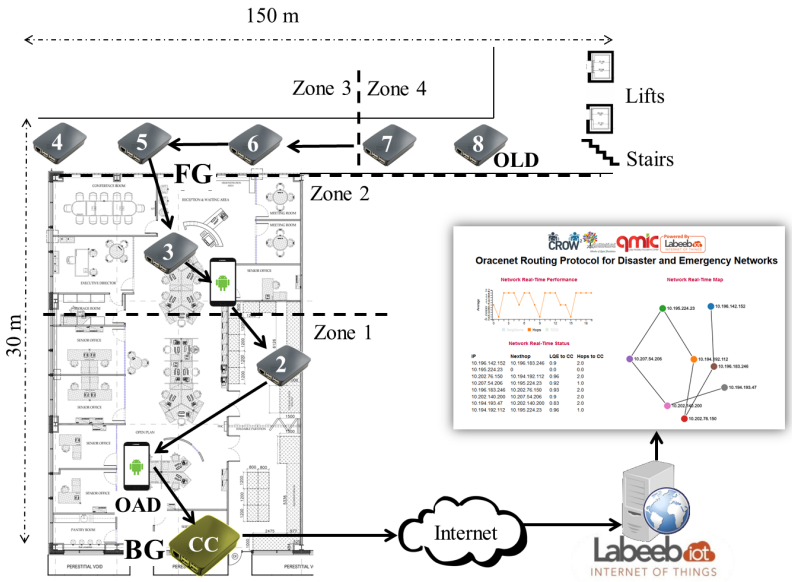


Fig. 5. Experimentation Scenario and Data Flow from Deployed Nodes to Labeeb-IoT Platform. Command Center (CC Node) is placed at the Back Gate (BG), ORACE-Net Android Devices (OAD) are mobile devices carried by the rescuers. The tactical ORACE-Net network is made by the ORACE-Net Linux Devices (OLD). Various experimentation area is divided into 4 zones. All collected data does through CC Node to Labeeb-IoT platform. A real-time dynamic topology website displays instantly the network topology.

TABLE I
EXPERIMENTATION PARAMETERS AND CONFIGURATION SETTINGS

General Settings	
Parameter	**Setting**
Tactical base station nodes	8 (raspberry pi 2) OS: Raspbian v8.0
Mobile nodes	2 (Samsung Galaxy S3-I9300 - rooted) OS: Android 4.2 CyanogenMod 10.0
Wireless mode	Ad hoc
ESSID	CROW2
Wireless standard	IEEE 802.11n / 2.412 GHz
Transmission power	0 dBm
Experiment area	$30m \times 150m$
CC-node connection	Ethernet to Internet WiFi to ORACE-Net network
Number of iterations	3
Experimentation duration	1 hour / iteration
Power batteries	- Smart phone: 2100 mAh Li-Ion (3.7v) - Raspberry-Pi: 10000 mAh Li-Ion (12v) - CC-node : 12v power supply
ORACE-Net Protocol Settings	
Application layer	MQTT client used for pushing data into Labeeb-IoT platform
MQTT message size	30 KB
MQTT message interval	1s
Hello/ADV packet size	20 / 25 Bytes
Hello and ADV intervals	3s
Multicast address/port	224.0.0.1 / 10000

metrics: *Average Packet Delivery Rate*, $E2E_{LQE}$, *Average Round Trip Time Delay*, and *Average Disconnections*. The *Average Packet Delivery Rate* is an application layer metric that provides the average of the received data packets on *Labeeb-IoT* platform against the data packets sent by each node of the network. The $E2E_{LQE}$ is calculated by ORACE-Net protocol as detailed in Section III. The *Average Round Trip Delay* is the time for an ICMP (ping) protocol to send a packet and get the acknowledgment between a node and the *CC node*. Also the ICMP protocol provides whether a node is connected

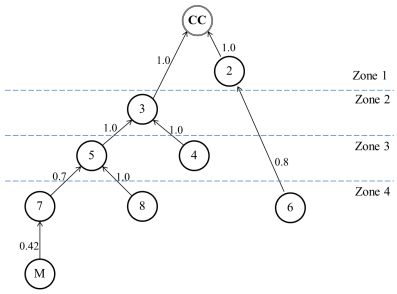


Fig. 6. Snapshot of the network topology after 24 minutes and 13 seconds of the experimentation.

to the network or not. *Average Disconnections* metric gives the average connection of a specific node. It is important to note that the following results were recorded inside QMIC's office. This office already includes many wireless base stations and devices. Thus, the resulting interference has a significant impact on the obtained results of ORACE-Net implementation.

1) Mobile node behavior over the time: Figure 7 shows the behavior of a smart phone mobile node during the entire experimentation. According to Table II, it can be seen that the Hop_{count} and the zone are not correlated. In fact, when a mobile node is in zone 2, this does not mean that it is always 2 hops far from the *CC node*. It could select a route through 2 nodes and then will have 3 hops to reach the *CC node*. Whereas, the $E2E_{LQE}$ and the Hop_{count} are correlated. In fact, the $E2E_{LQE}$ decreases when the Hop_{count} increases (as shown in Table II and Figure 7). On the other hand, the delay provided as the Round Trip Time from the mobile node to the *CC node*, shows three main peaks when the mobile node is in zone 4. Indeed, this caused significant delay and even disconnection, which is explained in details below in subsection V-B5.

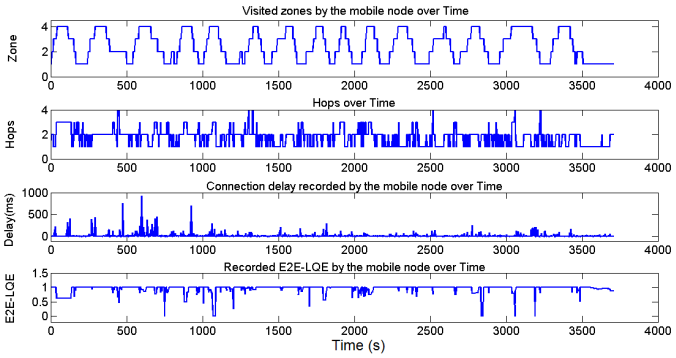


Fig. 7. Zone, Hop count, Delay and E2E-LQE variation during the experimentation (1 hour) for a smart phone mobile node.

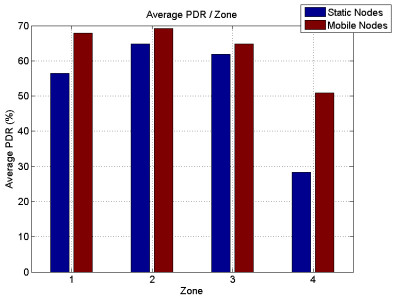


Fig. 8. Average Packet Delivery Rate per zone for Mobile and Tactical static nodes.

2) Average Packet Delivery Rate: The results of the Average Packet Delivery Rate are shown in Figure 8. Average PDR is given by zone and based on the tactical and mobile nodes separately. Overall, mobile nodes are more performant than the tactical nodes. But similarly, mobile and tactical nodes achieve the best PDR average in zone 2 with 69.23% and 64.75% respectively. Mobile nodes behave similarly in zone 1 and 2 in terms of packet delivery rate. However, tactical nodes have lower average PDR in zone 1 than in zone 2. Indeed, it is true that distance from node 3 in zone 2 to the *CC node* is higher than distance from node 2 in zone 1 to the *CC node*. But, there is an open space between the *CC node* and node 3 in zone 2. Whereas, many obstacles between node 2 (zone 1) and the *CC node* causes the SNR degradation. Additionally, zone 1 is the congestion zone in the network, since all nodes are sending their data towards one destination (*CC node*) which is in zone 1. The lowest average PDR for mobile and tactical deployed nodes is recorded in zone 4, achieving 28.37% and 50.87% respectively. According to our scenario, zone 4 is the safe zone as explained above, where victims and rescuers are considered out of danger. Thus, in this safe area, there is no urgent need to be connected to the network. So, mobile nodes may get disconnected from the network, which can affect the obtained average PDR recorded in that zone. Moreover, the tactical nodes have a low average PDR value, because zone 4 contains the most important number of persons (medical teams, civilians, media, etc.) compared to the other zones in the whole disaster area. Therefore, the reason for low PDR is mainly the radio signal blockage by the human

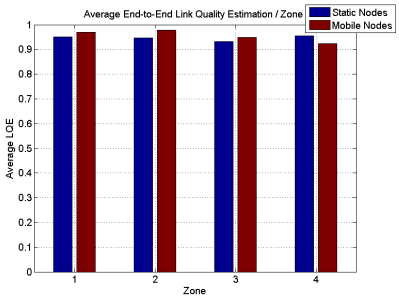


Fig. 9. Average End-to-End Link Quality Estimation per zone for Mobile and Tactical nodes.

TABLE II
CORRELATION BETWEEN THE METRICS

	E2E-LQE	**Zone**	**Hops**	**Delay**
E2E-LQE	-	-0.1780	-0.4225	-0.0176
Zone	-0.1780	-	0.1069	0.0933
Hops	-0.4225	0.1069	-	6.0642e-04
Delay	-0.0176	0.0933	6.0642e-04	-

bodies rather than being out-of-range. For the same reason, the number of re-transmissions increases and affects consequently the *Average Round Trip Time Delay*.

3) Average End-to-End Link Quality Estimation per zones: The $E2E_{LQE}$ results for mobile and tactical nodes are depicted in Figure 9. The best performance is recorded in zone 1 and 2 similarly for mobile and tactical nodes. The average $E2E_{LQE}$ is used for routing decisions in ORACE-Net, when the node is disconnected, routing table is empty and $E2E_{LQE}$ is not considered. For that reason, the average $E2E_{LQE}$ in zone 4 is higher than 0.9, because the average $E2E_{LQE}$ is only calculated when a node is connected with available route to the *CC node*. This observation leads to the fact that the $E2E_{LQE}$ is a relevant metric to consider in tactical disaster relief routing.

4) Average Round Trip Time Delay for mobile nodes per zones and hops: Figure 10 shows the average round trip time delay for the mobile nodes by zones and hops in milliseconds. As can be seen, the best performance is recorded in zone 1. The delay increases slightly from zone 1 to 3 and it reaches 29 ms. Results of the RTT delay are coherent with the average PDR explained above. Indeed, the delay increases significantly in zone 4 and with 4 hops, reaching up to 72.5 ms over 4 hops and 55ms in zone 4. Definitely, the significant delay recorded is not caused only by radio signal mitigation by the human bodies and obstacles as explained in V-B2, but also due to the low TX-power configured on the devices, which is reduced to 0 dbm.

5) Average Disconnections for mobiles nodes per zones and hops: The average disconnections for the mobile nodes is shown in Figure 11. The lowest disconnection rate is recorded in zone 1 and it is equal to 2%. Whereas, the best performance by hops is within 1 hop and equal to 12.01%. The network coverage in zone 1,2 and 3 is better than zone 4, that's why the highest average disconnections is recorded in zone 4. Additionally, the average disconnections within 3 hops is also relatively high (i.e, 37.15%). Figure 7 shows that 65% of the

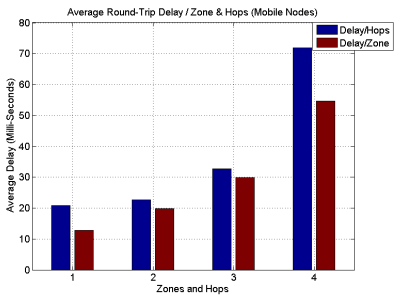


Fig. 10. Average Round Trip Time Delay per zone for Mobile nodes.

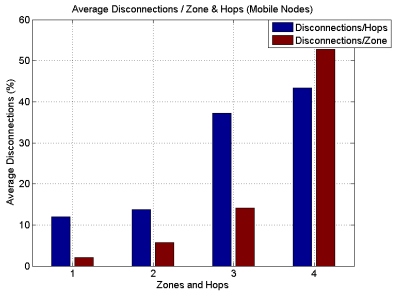


Fig. 11. Average Disconnections per zone for Mobile nodes.

3 hops are recorded in zone 4. This explains, that the average disconnections within 3 hops is high because the mobile nodes were in zone 4. Therefore, in the considered scenario, the mobility of the nodes decreases the average disconnections. Finally, please note that according to our scenario, zone 1, 2 and 3 are the most critical zones in the rescuing operations. Hence, in zone 1, 2, and 3 the protocol is more efficient than in zone 4.

VI. Conclusion

In this work, we considered the issue of tactical mobile ad-hoc communications inside a disaster area. We proposed a new routing approach which relies on end-to-end link quality estimation and shortest path for routing decisions. We conducted extensive experiments using eight tactically deployed wireless devices and two smart phone devices on which we implemented ORACE-Net protocol. Then, we evaluated the protocol in a real indoor scenario. Despite the important interference inside and outside the office, the experiments show that the proposed approach provides a good average connectivity and average packet delivery rate. Further, we observe a decrease in round trip time delay for the mobile nodes compared to the tactical deployed nodes. As future works, we will investigate the performance of ORACE-Net by connecting on-body sensors to the smart phone devices in order to send the vital data of the rescuers and victims to the command center.

Acknowledgment

This publication was made possible by NPRP grant $\#[6 - 1508 - 2 - 616]$ from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] "Natural disasters in the middle east and north africa: a regional overview," Tech. Rep. 1, 3 2014.

[2] M. M. Alam, D. Ben Arbia, and E. Ben Hamida, *Wireless Public Safety Networks 2, 1st Edition - A Systematic Approachs. Chapter 3: Wearable Wireless Sensor Networks for Emergency Response in Public Safety Networks.*, 1st ed. ISTE Press - Elsevier, 6 2016, vol. 2.

[3] J. P. P. Varun G. Menon and J. Priya, "Ensuring reliable communication in disaster recovery operations with reliable routing technique," *Mobile Information Systems*, p. 10, 2016.

[4] T. Zhuang, P. Baskett, and Y. Shang, "Managing ad hoc networks of smartphones," *International Journal of Information and Education Technology*, vol. 3, no. 5, pp. 540–546, 2013.

[5] M. M. Alam and E. B. Hamida, "Surveying wearable human assistive technology for life and safety critical applications: Standards, challenges and opportunities," *Sensors*, vol. 14, no. 5, pp. 9153–9209, 2014.

[6] D. B. Arbia, M. M. Alam, R. Attia, and E. B. Hamida, "Behavior of wireless body-to-body networks routing strategies for public protection and disaster relief," in *proceedings of the Workshop on Emergency Networks for Public Protection and Disaster Relief (EN4PPDR 2015), in 11th IEEE WiMob Conference*, Oct. 2015.

[7] C. Tata and M. Kadoch, "Multipath routing algorithm for device-to-device communications for public safety over lte heterogeneous networks," in *Information and Communication Technologies for Disaster Management (ICT-DM), 2014 1st International Conference on*, March 2014, pp. 1–7.

[8] W. S. Jung, H. Ahn, and Y. B. Ko, "Designing content-centric multi-hop networking over wi-fi direct on smartphones," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 2934–2939.

[9] A. Laha, X. Cao, W. Shen, X. Tian, and Y. Cheng, "An energy efficient routing protocol for device-to-device based multihop smartphone networks," in *Communications (ICC), 2015 IEEE International Conference on*, June 2015, pp. 5448–5453.

[10] H. Shuai and R. Jiamin, "The research on olsr routing protocol based on android system and its implement," *International Journal of Information Engineering*, vol. 2, no. 4, pp. 147–151, 2012.

[11] P. Baskett, Y. Shang, W. Zeng, and B. Guttersohn, "Sdnan: Software-defined networking in ad hoc networks of smartphones," in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Jan 2013, pp. 861–862.

[12] D. T. N. R. Group. (2015) Delay tolerant network 2. [Online]. Available: https://dtnrg.org/

[13] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 128–136, March 2010.

[14] G. C. Zongqing Lu and T. L. Porta, "Networking smartphones for disaster recovery," *Proceedings of IEEE PerCom, 2016*, 2016.

[15] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254. [Online]. Available: http://doi.acm.org/10.1145/345910.345953

[16] D. Kang, C. Joo, S. Kang, and S. Bahk, "Virtual back-pressure routing in mobile ad-hoc networks for disaster scenarios," UNIST, Tech. Rep., 2013, available at http://netlab. unist. ac. kr/cjoo/infocom14tr. pdf, Tech. Rep.

[17] C. Clausen, T. Dearlove and J. Dean. (2011) Mobile ad hoc network (manet) neighborhood discovery protocol (nhdp),. [Online]. Available: https://tools.ietf.org/html/rfc6130

[18] E. Ben Hamida, M. M. Alam, M. Maman, and B. Denis, "Short-term link quality estimation for opportunistic and mobility aware routing in wearable body sensors networks," in *proceedings of the IEEE 10th International Conference on Wireless and Mobile Computing Networking and Communications (WiMob)*, Oct 2014, pp. 519–526.

[19] E. Ben Hamida and G. Chelius, "Investigating the impact of human activity on the performance of wireless networks: An experimental approach," in *WoWMoM 2010 Conference*, June 2010, pp. 1–8.

[20] Q. M. I. Center, "Labeeb iot platform and solutions,," 2016. [Online]. Available: www.labeeb-iot.com

[21] "Information technology message queuing telemetry transport (mqtt) v3.1.1 - iso/iec 20922:2016," 7 2016, geneva,CH.