WiMesh: Leveraging Mesh Networking For Disaster Communication in Poor Regions of the World

Usman Ashraf, Amir Khwaja, Junaid Qadir, Stefano Avallone, Chau Yuen

Abstract—This paper discusses the design, implementation and field trials of WiMesh - a resilient Wireless Mesh Network (WMN) based disaster communication system purpose-built for underdeveloped and rural parts of the world. Mesh networking is a mature area, and the focus of this paper is not on proposing novel models, protocols or other mesh solutions. Instead, the paper focuses on the identification of important design considerations and justifications for several design trade-offs in the context of mesh networking for disaster communication in developing countries with very limited resources. These trade-offs are discussed in the context of key desirable traits including security, low cost, low power, size, availability, customization, portability, ease of installation and deployment, and coverage area among others. We discuss at length the design, implementation, and field trial results of the WiMesh system which enables users to spread over large geographical regions, to communicate with each other despite the lack of cellular coverage, power, and other communication infrastructure by leveraging multi-hop mesh networking and Wi-Fi equipped handheld devices. Lessons learned along with real-world results are shared for WiMesh deployment in a remote rural mountainous village of Pakistan, and the source code is shared with the research community.

Index Terms—Wireless mesh networks, Disaster communication, Multi-hop network.

I. INTRODUCTION

While natural disasters have long been a scourge for humanity, our ability to respond to such emergencies has gradually been improving with the aid of technology. In recent times, a new powerful trend of “digital humanitarianism” [1] is emerging in which digital communication facilities are being used alongside big data and crowdsourcing techniques to scale up humanitarian response to natural disasters and other crisis situations [2]. The availability of communication services in crisis-hit situations is essential due to the facility it provides in coordination between the various stakeholders.

Wireless networking in particular has revolutionized communication by offering innovative, flexible, and cost-effective untethered solutions. Due to the significant benefits of the wireless technology, today we have a diverse range of wireless technologies at our disposal. Cellular technologies can be perhaps at the forefront of this wireless revolution and have penetrated even remote areas around the world. However, there are still some scenarios where these technologies fail. In particular, the cellular infrastructure as well as the existing Public Switched Telephone Network (PSTN) may be destroyed during disasters and alternate means may need to be investigated for disaster communication. Moreover, despite its deep penetration, there are still sizable areas of population at remote locations where the low return on investment (ROI) along with the difficulty of terrain have made deployment of cellular infrastructure infeasible. In addition, cellular technologies have also found it difficult to break into rural and low-income regions because common models of operations require a high Average Revenue Per User (ARPU), which rural areas cannot support since rural areas are sparsely populated [3].

South-East Asia, and Pakistan in particular, is a disaster-prone area that has suffered a number of calamities that have wreaked havoc and created misery. This research work is particularly motivated by the increasing frequency of disasters in Pakistan in the last decade or so. On October 8th, 2005, there was the Kashmir earthquake that resulted in 80,000 fatalities and 3.5 million homeless people [4] [5]. On October 29th, 2008, the Ziarat earthquake hit the Pakistan province of Balochistan resulting in 215 fatalities and 120,000 homeless people [6] [7]. During the earthquakes, the traditional telecommunication infrastructure, including towers and transmission lines, were destroyed rendering communication difficult and in many cases impossible. This region is also home to several underdeveloped remote villages and tiny communities where cellular penetration has not reached. Even the areas with coverage were badly damaged by the earthquakes and rehabilitation is still going on several years later.

Responding to emergency situations is not only difficult but can sometimes be life-threatening. One of the most difficult aspects of rescue is ensuring efficient coordination among rescue team members on hostile terrain in an unfamiliar territory, such as crumbled buildings during an earthquake and buildings on fire. Rescue workers, separated by even a couple of hundred of meters of hostile terrain, can face problem in communicating. Moreover, communication in such scenarios is not necessarily restricted to radio communication but may also involve text messages, live video feeds, geographical positioning information, and triggered warning messages among others. A dire need was felt for an appropriate wireless disaster communication system for these areas in responding to emergency situations for on-site coordination among the members of the rescue teams. In addition, such a system could also help in the post-disaster rehabilitation process spanning...
several months or even years.

An effective and successful disaster communication system should have the following desirable characteristics:

- Should be cheap, rugged, easy to build, and easy to deploy (especially for rural deployments in developing countries);
- Should have secure communication as well as protection against unauthorized access;
- Should have inbuilt fault tolerance and resiliency through redundancy at all levels including multiple communication paths;
- Should be flexible enough to effectively provide wireless coverage over diverse terrains; and
- Should continue to work even when the the power grid or the conventional communication infrastructure (PSTN and cellular) is disrupted.

There are existing disaster communication systems that have been around for decades, however, each have some limitations. For instance, satellite phones provide universal coverage, however, during the earthquake of 2005 in North-Pakistan, the high cost of satellite phones prohibited their regular use. Radio-based communication has the limitations of range, bandwidth, and functionality and only provides poor quality audio broadcast. Several wireless base stations and various communication cables were destroyed during Hurricane Katrina and the remaining network was not able to support communication services to first responders [8]. The legacy and tethered emergency response systems either are severely limited or completely destroyed in disaster struck regions due to their over dependence on the terrestrial infrastructure [9] and may not be able to support high bandwidth requirements for multimedia communication needed for disaster communication.

This paper presents the WiMesh system, a secure and resilient wireless mesh network based disaster communication system designed, developed, and deployed at a conglomeration of remote and underdeveloped villages and tiny cities in the mountainous South-West region of Pakistan. The WiMesh system comprises of a two-tier architecture: the first tier consists of a portable, battery-powered wireless mesh network which is auto-configuring, self-healing, and rapidly-deployable; whereas the second tier consists of mobile handheld phones carried by rescue workers which connect automatically with the nearest mesh access point through Wi-Fi enabling the workers to communicate and coordinate with other team members. A prominent feature of the WiMesh system is that it is a complete product, and provides several useful features including audio communication (one-to-one as well as broadcast mode), SMS messaging facility, ability to see the live geographical positions of other members at all times, and multimedia streaming facility. These features, and the cost-effectiveness of the proposed approach, make the WiMesh system appealing for use in disaster areas or an alternative to traditional communication in remote areas.

The major contributions of the paper are as follows:

1) WiMesh system details — a secure and resilient disaster communication system based on multi-radio, multi-channel 802.11n wireless mesh nodes — that provides audio, video, and text communication by exploiting the Wi-Fi capability of mobile phones in disaster-affected or remote and under-privileged areas where traditional communication and power infrastructure is non-existent or damaged;

2) The design and implementation details of the client and server applications, security considerations, routing protocols, and the resilient routing metric in light of the various design trade offs with respect to the constraints imposed by the infrastructure, equipment availability, and budget for developing countries. The feasibility and the performance of the WiMesh system is demonstrated using a thorough performance evaluation using both an indoor test bed and in a real-world outdoor setting; and

3) A discussion on common implementation challenges and pitfalls and sharing of general lessons that can guide researchers and practitioners in designing and developing future such systems. In addition, the source code, design specifications, and implementation guidelines for the WiMesh system are also shared publicly to facilitate community development and knowledge sharing.

The focus of this paper is not to propose yet another mesh modeling or routing protocol, but instead we believe that the state-of-the-art for mesh networks is quite mature and we focus instead on applying the research to solve a real-world problem while trying to balance several unique constraint imposed in the context of countries with very limited resources and semi-literate populace in remote villages who need a plug-and-play solution. We do share some experimental results, but those serve more as validation of the concept rather than focus on providing deep insights on the functioning of the system. We target remote villages and rural areas with little to no communication and power-grid infrastructure to start with. Additionally, in contrast to the disaster communication systems designed for developed countries, financial constraints and scarcity of resources were the primary design considerations for WiMesh. Moreover, the end-users of the WiMesh system comprise mostly of rural population with little to no knowledge or technical expertise. Thus, several of the design decisions were motivated by the cost constraints, availability of technology, and with a plug-and-play approach. The paper also provides rare insights into how modern wireless communication technology is deployed and adopted by the rural population in Ziarat, a disaster-struck and harsh remote mountainous region in South-West Pakistan with its unique centuries old traditions and values. The paper also shares actual field results from the experiments conducted in Ziarat after WiMesh was deployed following the earthquake of 2008 in Ziarat with its unique geographic diversity.

The rest of the paper is organized as follows. Section II summarizes a review of the literature and the state of the art. The key differences are highlighted from existing works and the motivation for this work is discussed. Section III identifies key disaster communication system features and discusses design choices for these features. Section IV provides the WiMesh system details including its overview, design con-
siderations, architecture, and various components. Section V provides results from field trials for the performance evaluation of the WiMesh system along with challenges and problems faced during the development and deployment. The major insights and lessons learned are summarized in Section VI. Section VII concludes the paper.

II. RELATED WORK

Rising fatalities across the world, from natural and men-made disasters, have attracted significant research attention for disaster communication systems. Commonly deployed wireless disaster communication systems may be classified into three categories: cellular, hybrid, and mesh/ad hoc disaster communication systems. The cellular infrastructure has been widely used in several disaster communication systems since it can provide significant coverage for areas with an existing and intact cellular infrastructure without the requirement of installing any additional hardware. However, these solutions are vulnerable as natural disasters often cause partial or complete damage to the telecommunication infrastructure. Moreover, the telecommunication infrastructure may not exist in remote areas of developing countries. Hybrid disaster communication systems leverage satellite connectivity in addition to the terrestrial infrastructure. As previously mentioned, the terrestrial infrastructure is vulnerable to disasters whereas the satellite subscription and equipment is costly, as was evidenced during the rescue operations in the 2005 Kashmir earthquake in the Northern Pakistan region in which satellite phones could only sparingly be used. The mesh or ad hoc disaster communication systems are based on leveraging wireless mesh and ad hoc networking to achieve rapid wireless multi-hop connectivity on disaster sites. Wireless mesh and ad hoc networks have several interesting features that make them an attractive option for disaster communication. Mesh networks offer wireless multihop connectivity, rapid deployment, minimal configuration, self-healing, and most importantly, these networks can often be built from off-the-shelf equipment. Due to these interesting features, mesh and ad hoc networking appears to be ideal candidates for disaster communication and, therefore, this section has focused on evaluating only mesh and ad hoc related network solutions.

**WIISARD** is an emergency response system for providing medical triage services by mobile first responders. The system initially used Ad hoc On-Demand Distance Vector (AODV), but later switched to a gossip-based communication protocol for delay tolerant networking. The main contribution of this research work is the characterization of wireless links and human mobility patterns during medical triage among mobile first responders. The research work presented in showed that network partition may occur and, therefore, proposed a gossip protocol for delay tolerant communication in the mesh IEEE SYSTEMS 3 backbone. One of the limitations of WIISARD system is that it uses a rudimentary routing metric of hop count that has been shown to perform sub-optimally.

**DistressNet** is an emergency response system for Urban Search and Rescue (USR) in disaster struck areas. Low-powered COTS (Commercial Off-the-Shelf) devices, such as smartphones and sensors, are used to assist in disaster communication. DistressNet uses IPv6 based routing protocol for low-power and lossy networks (RPL). The major contributions include file storage and social networking services offered over a delay tolerant network. DistressNet also supports a vibration sensing application that helps detect victims trapped under rubble. The main limitation of this solution is that the performance evaluation is performed on a piecemeal basis and lacks results for a holistic and comprehensive performance evaluation.

**OEMAN** (On-the-fly Establishment of Multihop wireless Access Networks) is a novel approach for disaster communications. This system leverages the availability of commodity mobile devices at disaster sites, turning them into virtual apps to extend Internet connectivity for disaster sites. There are several limitations including implementation and evaluation in a limited indoor building setting, support for only root top AP route discovery without any support for node-to-node communication, and scalability problems.

**CHORIST** (Communications for enHanced envirOnmental RISk management and citizens safeTy) system forms a WiMAX based mesh Vehicular Ad hoc NETwork (VANET) backbone through Wi-Fi. The system offers voice services and video surveillance for disaster sites. A layer 2.5 label switching approach is used for data communication. The main limitation of this solution is that it requires expensive and time consuming WiMAX base stations infrastructure setup for the wireless backbone and the availability of abundant vehicles. Moreover, the cost of the WiMAX licensed band may also limit the utility of this approach.

**WIDENS** (WIREless DEployable Network System) uses a reliable communication system for real-time applications in disaster situations. This system adopts a cross-layered, cluster-based approach to provide high speed communication through hotspots and exploits the high bandwidth and wireless range of the MIMO (Multiple Input Multiple Output) at the physical layer. This system introduces the concept of “Terminodes” in which each node in the network can perform the functions of a cluster-head, relay, router, or gateway, thereby, offering flexibility. The routing protocol used by the proposed solution is based on secure source routing, which addresses some security limitations in other similar systems. The main limitation of this solution, however, is that it uses licensed frequency bands, which may limit its use or may introduce excessive costs in some countries.

**SALICE** (Satellite-Assisted Localization and Communication system for Emergency services) system uses a hybrid disaster communication comprising of ground infrastructure as well as satellite connectivity. This work employs several different concepts including satellite-assisted localization, Software-Defined Radios (SDR), Cognitive Radios (CR), and satellite-assisted enhanced reachability for rescue workers. Due to these powerful and diverse technologies, the SALICE system offers comprehensive functionality with flexibility and heterogeneity for emergency situations. This research work highlights the benefits of the SDR technology for addressing the dynamic
| Features                      | WISARD                  | DistressNet        | OEMAN               | CHORIST              | WIDENS               | SALLICE             | VIEDOS               | WIMESH               |
|-------------------------------|-------------------------|--------------------|---------------------|----------------------|----------------------|---------------------|----------------------|----------------------|
| Services                      | First responder medical triage; Services for victims | Urban search and rescue; Internet connectivity for social networking between rescue workers | Search and rescue; Internet connectivity for social networking among rescue workers; Communication among victims via mobile devices | Voice services; Applications for public safety | Reliable communication for real-time applications in disaster situations; Security and video surveillance | Emergency communication, navigation and localization service for very large geographic areas | Security, communication, authentication and confidentiality | Secure and resilient services including multimedia (audio/video/SMS) and navigation and localization. |
| System Architecture           | Distributed, flexible and modular architecture | Distributed | Fully centralized | Mainly centralized with some distributed components | Mainly centralized with some distributed components | Mainly centralized with some distributed components | Completely distributed | Decentralized distributed architecture. |
| Wireless Network Structure, Technology, and Coverage | 802.11 for mesh networking; Medium range | 802.11 for mesh networking; Medium range | 802.11n for Wi-Fi; Medium range | 802.11n for edge mobile first responders; WiMax for vehicular backbone; Long range | OFDMA; MIMO; Long range | IEEE 802.11; DV-B-5; Very long range | Bluetooth; Very short range | IEEE 802.11n mesh networking technology; Long range. |
| Routing                       | Delay tolerant networking; Gossip based communication protocol (WCP); Hop count metric; AODV routing | RPL routing protocol for low power and lossy networks | Tree-based unidirectional forwarding from root to leaf nodes; No node-to-node routing | Layer 2-5 label switching approach | Secured sourced routing extension of OLSR protocol | Extension of ODMRP routing protocol | Delay tolerant networking device-to-device | OLSR routing protocol with Expected Link Performance (ELP) metric. |
| Security & Resilience         | Delay tolerant networking; Immune to comm. delays; No special security measures | Tolerant against delays and disruptions social networking ensured at all times; No special security measures | Limited fault tolerance due to centralized model; No special security measures | Resilient backbone based on WMAX; High security | Tolerance to link and node failures by using secure OLSR protocol; Highly secure protocol | Reliable centralized backbone; Highly secure | Resilient to Denial of Service attacks through epidemic dissemination; Specially design for security against DoS attacks | Secure and Resilient. Uses (i) authentication; (ii) encryption (using WPA2-AES); (iii) resilient routing using ELP metric. |
| Power Requirement             | Low                      | Low                | Low                 | High                | High                 | High               | High                 | Low                  |
| Software & Operating System   | Multiple OS support (Linux, WinXP, Max OS) | Multiple OS support (TuxOS, SGS, LinOS, Mantis, Linux, Windows) | Windows OS | Commercial proprietary software | Embedded OS (PHY/MAC) | Linux Ubuntu | Embedded OS (PHY/MAC) | OpenWRT (Mesh Nodes); Android OS (WiMesh clients). |
| Hardware                      | Triage devices (Nokia mobiles); Mid-tier (Tablet PC); Mesh boxes (Alix x86); RFID tags | Mobile phones; Sensors; Wireless APs | Mobile phones | Laptops; Mobile phones | Base station equipment | PCMCIA Adapters | Software-defined radios, laptops, mobile base stations | Ubiquiti PicoStation and the Nanostation devices. |
| Size & Portability            | Small size; Light; Highly portable | Small size; Light; Highly portable | Small size; Light; Highly portable | Large size; Heavy; Portability difficult - requires vehicles | Small size; Light; Highly portable | Not portable, requires infrastructure | Not portable, requires infrastructure | Highly Portable. |
| Installation Complexity       | Low                      | Low                | Low                 | Medium              | High                 | High               | Low                  | High                 |
| Cost                          | Low                      | Low                | Low                 | High                | Low                 | High               | Low                  | Low                  |
| Availability                  | Commercial off-the-shelf (COTS); Digital tags may be difficult to acquire in developing countries | COTS | COTS | Required hardware for base stations and associated network may be difficult to acquire at remote sites | Custom hardware and software | Custom hardware | Custom hardware | Relatively cheap and easily available equipment is used. |
| Major Contributions           | Mitigates network partitions due to mobility of rescue workers using DTN; Characterization of link quality and human mobility patterns during medical triage among mobile first responders | Offers file staging; Network networking despite network delays and disruptions; Mobile vibration sensing application detecting victims under rubble with improved accuracy | Converts commodity mobile devices into Virtual Access Points (VAPs); Supports extensions of Internet connectivity to disaster sites | Provides video surveillance and voice communication for disaster sites by providing a WiMAX VANETs for vehicles to which mobile first responders connect through Wi-Fi | Cross-layered approach for high data rate service; Terminode concept where each node performs the function of cluster head, relay, router, or gateway; Secure source routing | Platform based on SDR technology for emergency services; Integration of different components and technologies. | Platform based on SDR technology for emergency services; Integration of different components and technologies. | A resilient WMN based disaster communication system for under-developed/rural areas using low-cost, highly available, portable equipment. |
| Limitations                   | Sub optimal routing metric hop count | Limited experimental evaluations performed indoors only using a piece-wise approach e.g. mobility algorithm, sensing, etc. | Internet only for rescue staff; Limited indoor evaluation only; Does not support peer-to-peer communication, only root-to-node; Does not perform load balancing, uses a single AP | Expensive; Uses licensed frequency bands | Uses licensed frequency bands | Platform offers limited throughput due to the limitations of the USRP in NLOS conditions | Platform offers limited throughput due to the limitations of the USRP in NLOS conditions | Due to its specific design for disaster communication scenarios it may not work well for other high-performance requirements. |

**TABLE I: Disaster Communication Systems Comparison**
A. Services

Disaster communication systems are expected to provide following services:

1) User registration and authentication - User registration is required at the server whereas authentication is needed both at the server and the client level;
2) Audio call - Audio calls can be either one-to-one or one-to-many (broadcast). One-to-one audio calls may be between admin and users or among users;
3) SMS - SMS may also be either one-to-one or one-to-many (broadcast). Like audio call, one-to-one SMS may be between admin and users or among users;
4) File transfer - File transfer may also be either one-to-one or one-to-many. Like audio call, one-to-one file transfer may be between admin and users or among users;
5) Live video streaming - Live video streaming functionality may consist of following types: sending live video stream to client, sending live video stream to server, receiving live video stream by client, and receiving live video stream by server. All live video streams are usually one-to-one between two users;
6) Navigation - The navigation functionality allows users to view the geographical location of other peers/users on a map with some necessary information. User can also view his own position on the same map relative to the other peers/users; and
7) Network performance monitoring - Disaster communication systems need capability for monitoring, logging, and viewing user and network statistics to ensure QoS for the coordinated relief efforts among geographically dispersed clients in adverse situations.

B. System Architecture

The system architecture plays an important role in the overall performance of any disaster communication system. The specific choice of architecture may be influenced by several external factors including terrain, weather, existing infrastructure, and cost.

Broadly, there can be three categories for system architecture: centralized, distributed, and hybrid. In general, the loosely coupled distributed architecture is the best choice for disaster communication due to a number of reasons including flexibility, fault tolerance, autonomy, and scalability. Moreover, logistic reasons, such as difficult terrains, also dictate that autonomous distributed components are more likely to survive the stringent requirements of disaster communication. However, the distributed architecture do introduce some security concerns due to lack of control. Popular solutions in the distributed architecture category include WISARD [13] and DistressNet [14]. At the other end of the spectrum, centralized solutions, such as OEMAN [15], even with some limitations, offer certain appealing features such as ease of use, control, and security. Some solutions [16]–[18] offer a compromise by combining the strengths of both these approaches. In general, these solutions have a fixed backbone (e.g., WiMAX or cellular) and a mobile mesh based last-mile. These solutions do offer a trade-off, however, the centralized architecture imposes the same limitations of scalability, flexibility, and cost.

C. Wireless Network: Structure, Technology, and Coverage

The wireless network is perhaps the most fundamental component of the disaster communication system and consists of several key aspects including its structure, technology, and coverage. The structure of a wireless network can either be centralized, distributed or hybrid. Distributed structures, such as ad hoc and mesh networks [13]–[15], offer the inherent advantages of multi-hop networking, while a hybrid approach [16]–[18] combines some form of a centralized backbone network (e.g. satellite or cellular) along with a mesh component.

Another important decision in the design of the WiMesh system is the choice of the wireless technology. Technologies like WiMAX or 4G offer certain advantages, however, have the limitation of infrastructure and licensing costs. Similarly, different technologies offer a trade-off in terms of wireless coverage as well. For instance, installing WiMAX or cellular towers can significantly increase the wireless reach of the network, however, have cost and logistic difficulty offsets. Similarly, off-the-shelf technologies like 802.11 are free and require minimal setup, however, have coverage limitations. In Wi-Fi technologies, the IEEE 802.11n standard uses Multiple-Input and Multiple-Output (MIMO), and provides significantly better performance by exploiting spatial and temporal diversity.
through multi-path propagation and offers impressive data rates of up to 600 Mbps. Therefore, IEEE 802.11n based radios are a natural choice for the mesh backhaul as well as offering access to clients.

D. Routing

Routing forms the backbone of any disaster communication system and the routing protocol and metric used play a decisive role in the overall system performance. Towards this end, there are choices of several routing protocols and metrics including proactive, reactive, and hybrid routing solutions. Some completely centralized solutions employ static routing, but by and large, most disaster communication systems need a strong proactive or dynamic routing protocol. Popular choices in this category include OLSR and DSR routing protocols. Proactive protocols such as OLSR offer quick routes, however, create messaging overhead that can be burdensome if the equipment is battery powered. On the other hand, the reactive routing approaches do not provide quick routes, however, can offer overhead savings. Another important criteria is that of the routing metric as the metric needs to be dynamic and integrates fault-tolerance and adaptability to changing network conditions. This is especially important in disaster situations since the network may experience failures and faults due to users moving out or range, environmental factors, and power draining. State-of-the-art metrics such as can offer efficient routing metric solutions which cater to changing link qualities, bandwidth, and node failures.

E. Security and Resilience

Disaster communication system, by virtue of its application, may have several diverse clients connected to its network. These clients may consist of mobile devices such as smart phones, laptops, tablets, and other communication gadgets of either official rescue workers or personal users. These devices may be used for rescue communications and data sharing or people requesting for aid or other emergency services. Due to this wireless, cooperative, and decentralized nature, these systems are vulnerable to various security attacks such as relief communication disruption by terrorists by injecting false messages, denial-of-service (DoS), and unintentional clogging of the network by spamming from people in panic. Moreover, malicious attacks could damage strategic nodes or links in the network and, therefore, the underlying routing protocols and metrics should be designed to introduce resilience against these types of attacks. Hence, effective disaster systems must provide user authentication, message integrity, secure routing, confidential end-to-end communication and resilient routing.

F. Power Requirement

Since disaster communication systems are envisaged to be deployed in rural and disaster-struck areas, an implicit assumption is that electrical supply will be limited or unavailable. To solve this problem, renewable energy systems, such as solar power, may be considered to power the nodes. This implies that the designed system must be energy-efficient. Moreover, the software on the mobile phones must be designed to minimize the energy drain.

G. Software and Operating System

There are three main parts of disaster communication software system including the server, the clients, and the wireless back haul network OS/firmware.

1) OS for Wireless Network: Proprietary OS and firmware provide a user-friendly graphical user interface. However, there is little control over different network components such as the routing protocol, the routing metric, and dynamic rate adaptation. Therefore, disaster communication systems should consider the adaptability and flexibility of open-source software.

OpenEmbedded [22] is an open-source cross-compilation environment for building Linux distributions for embedded applications. It offers support for multiple hardware, customization, and rapid development. A set of rules and instructions, called a “recipe,” is used to generate architecture specific applications and packages. However, OpenEmbedded is difficult to use due to its complexity.

Linux From Scratch (LFS) [23] offers powerful paradigm for building custom Linux distributions from the source. The result is a compact, flexible, and secure system with a better understanding of the internal working of the OS. However, building LFS requires significant Linux proficiency to sort out the configuration dependencies and compilation errors.

DD-WRT [24] is a Linux based firmware for wireless routers and access points. DD-WRT provides a custom firmware option with several features and functionalities such as access control, bandwidth monitoring, quality of service, Dynamic DNS, universal plug-and-play, and many others.

OpenWRT [25] is an open-source Linux for embedded devices for routing network traffic. OpenWRT uses a smaller µClibc C library to reduce the memory footprint, which makes it possible to run on various types of devices, such as home routers, residential gateways, smartphones, pocket computers, and laptops. OpenWRT offers a built-in package manager that allows the installation of packages from a comprehensive software repository. OpenWRT can be used as an SSH server, a Bit-Torrent client, and VPN. OpenWRT offers significant flexibility and has an extensive support base.

2) Software for Clients: There are several mobile application development platforms available for the client OS. The Mobile Information Device Profile (MIDP) [26] is the toolkit from Sun for creating mobile applications and is considered quite easy to learn and use. The Symbian OS [27] is another popular platform for mobile phones and is used by Nokia in their mobile phones. Symbian OS is significantly more complex compared to MIDP. The iPhone OS is the operating system used in Apples iPhone and iPod touch. iOS is designed to be extremely efficient on mobile devices, using less than half a gigabyte of memory. The BlackBerry platform uses Java for application development and runs applications in protected run-time environment. It has several libraries with
sound inter-operability. BlackBerry has acceptable level of robustness and reliability, however, the BlackBerry platform does not allow programmers controlling the GPS and scanning of cell towers or Wi-Fi access points (APs) programmatically. Moreover, it is not very memory efficient. The Pocket PC/Windows Mobile OS is a mobile version of the desktop Windows OS. It has been designed mainly for Pocket PC and Smartphone devices and as such limits the third party development opportunities.

3) Software for Server: Every disaster communication system requires the usage of VoIP technology to support messaging and audio communication. While there are several dozen VoIP servers available, to limit the scope of discussion, this paper only considers the two major VoIP servers, Asterisk [30] and OpenSIPS [31]. Asterisk is a software implementation of the telephone Private Branch Exchange (PBX). Asterisk enables telephone calls between users connected to Asterisk as well as to other telephone services such as the PSTN and VoIP services. Asterisk is compatible with several operating systems including NetBSD, OpenBSD, FreeBSD, macOS, and Solaris. Asterisk supports several standard VoIP protocols, including the Media Gateway Control Protocol (MGCP), Session Initiation Protocol (SIP), and H.323. The OpenSIPS is basically a multifunctional, multi-purpose signaling SIP server for VoIP that can handle voice, text, and video communication. OpenSIPS can handle thousands of calls simultaneously. The fundamental difference between the two VoIP servers is that Asterisk is basically a media server, whereas OpenSIPS is essentially a SIP proxy server. While Asterisk has a better support for multimedia communication, the robustness, flexibility, adherence to SIP standards, makes OpenSIPS a suitable choice for the disaster communication systems.

H. Hardware

The major choice of hardware is in the realm of the infrastructure of the wireless network side including Access Points (APs) and antennas among others. The backbone for disaster communication can be created ranging from relatively simple off-the-shelf wireless access points to powerful network boards like Gateworks Cambria boards [32]. The higher end network boards are more powerful, rugged, and typically have multi-radio support, however, are more expensive and complex. Here, it is important to realize that there are important aspects including the cost, energy consumption, availability and robustness of the hardware which may play an important role in decision making. It is hard to deploy a solution which addresses problems of underdeveloped countries but requires importing expensive equipment that is not readily available. Moreover, the mesh nodes should also be easily portable and rugged since the equipment may need to be rapidly deployed over difficult geographical terrain. The mesh nodes should also be energy-efficient and would need to be solar-powered out in the field due to the likely destruction or non-availability of power grid in the disaster areas.

I. Size and Portability

Since disaster communication systems may be used over unfamiliar and uneven terrain, under hostile circumstances, and in inclement weather conditions, these systems should be easy to assemble, disassemble, carry, and move around. Often vehicles are not available or offer limited maneuverability. Hence, such systems need to be designed considering smaller size, lighter weight, and high portability.

J. Installation Complexity

A major factor which can dictate the eventual adoption and success of disaster communication systems is the installation complexity of the complete system. The installation complexity may include the physical deployment as well as configuring the server and client software. For disaster-struck, underdeveloped areas, it may be difficult to have highly skilled people. It would be pointless to add to the list of problems the complexity of deploying and configuring the disaster communication systems. A plug-and-play approach for all components of the system should be followed and extensive use of automated scripts for accomplishing repetitive tasks including bootstrapping common settings is recommended.

K. Cost

Since these disaster communication systems are expected to be used in developing and underdeveloped countries, the cost is an important factor to be considered in the design of such systems. Cost factor may consist of hardware equipment cost, software cost, maintenance cost, and skills cost. The choice of the hardware equipment should be as such that the cost is minimal and affordable in the developing countries. The Software cost may be minimized or eliminated by selecting open source and freeware software. The design of the system should be flexible and adaptable so that maintenance cost is minimal. The system should be easy to setup and use to eliminate need of highly skillful operators.

L. Availability

Availability of the equipment is an important consideration especially for the developing countries. Often there is a choice of employing state-of-the-art equipment which is highly efficient and effective, however, their availability may not be as widespread. For instance, the Avila Gateworks board [33] is considered a powerful solution for outdoor wireless networking, however, is not easily available in developing countries.

IV. The WiMesh System

This section presents architectural, component design, and implementation details of the WiMesh system. In addition, design choices for the WiMesh system are presented for the system features identified in section [I]
A. Design Choices for WiMesh

1) Services: Based on the key services identified in section III-A for disaster communication systems, following services were selected to be incorporated in the WiMesh system: audio calls in both unicast and broadcast modes; multimedia data exchanges such as images, video, data files, and SMS in both unicast and broadcast modes; sending and receiving live video streaming to/from clients and server; and geographical position viewing of the WiMesh users. Moreover, the WiMesh controller provides additional services such as remotely monitoring, configuring, and controlling the entire network even at the wireless interface level. Additionally, the server also provides user registration and authentication services.

2) System Architecture: For the WiMesh system, a decentralized architecture was selected to avoid the inherent limitations of centralized networks. The decentralized architecture was combined with the power of mesh networking, exploiting its traditional strength areas such as multiple available routes, adaptability of changing requirements, and flexibility. Thus, the WiMesh system has been designed to enable loose coupling among autonomously working components in order to avoid a purely centralized approach. The server does introduce some centralized components, however, the routing in the mesh backbone itself is highly flexible, independent and adaptable to changing environment and network conditions.

3) Wireless Technology and Coverage: For the wireless technology, the off-the-shelf license free IEEE 802.11n technology was selected for the WiMesh system. There is the trade-off of limited wireless coverage at hand, however, by using 802.11n technology and powerful outdoor multiple sector antennas, the wireless range issues can be somewhat mitigated.

4) Routing: The OLSR protocol was selected for routing in the WiMesh network since it is a proactive protocol allowing for real-time routing decisions. Moreover, there are several implementations available for this protocol. For the routing metric, the Expected Link Performance (ELP) routing metric was integrated since it provides state-of-the-art performance by incorporating aspects such as resilience, link qualities, interference, load, and link asymmetries.

5) Security and Resilience: The WiMesh server security and resilience was designed from three aspects. First and foremost, the WiMesh system registers and authenticates users (Authentication API module) in order to ensure that only authorized users access the services (there is a special emergency mode available which allows all users to register and connect without authentication). Second, the wireless communication is completely encrypted using WPA2-AES encryption for the wireless state-of-the-art encryption standards. Third, the ELP metric introduces resilience in the overall routing approach in the sense that nodes and link failures would be mitigated through the use of efficient and dynamic estimation of route qualities.

6) Power Requirement: The WiMesh system is designed to function in remote areas through the use of batteries and solar panels. This has been demonstrated through several experiments that none of the equipment is particularly power hungry including the server, controllers, and the mesh nodes and the system can function perfectly off-the-grid. Additionally, the WiMesh system conserves energy by minimizing the usage of the GPS module through periodic sleeping.

7) Software and Operating System: For the wireless back-haul network, the obvious choice for the WiMesh system was an open-source Linux distribution like OpenWRT or DD-WRT because these OS/firmware are easy to control, modify, and extend. Significant changes can be made to the protocol stack for designing efficient protocols and integrating customized code and scripts. Therefore, OpenWRT was used as the Linux distribution for the mesh nodes.

For the server software, Asterisk has a better support for multimedia communication, robustness, flexibility, and adherence to SIP standards. However, it seemed that the speed and call capacity of the OpenSIPS server, along with its tremendous scalability capacity, makes it an ideal choice for the WiMesh server.

The Android OS [34] open-source development platform was selected for the WiMesh mobile clients. Android OS offers component-based architecture, built-in APIs, automatic lifecycle management, and high portability.

8) Hardware: The principal hardware considerations were cost effectiveness, off-the-shelf availability, and open-source programmability for full control over device functionality. Hence, the hardware choice for this project was the Ubiquiti Picosat and the NanoStation devices [35].

9) Size and Portability: Highly portable equipment was selected, e.g., extensible poles that can be quickly extended or collapsed. The wireless APs and other devices were also lightweight and portable. No heavy installations were required, making the WiMesh system a highly portable solution.

10) Installation Complexity: Extensive use of bootstrapped automated scripts were used to ensure common settings and easy deployment. The OpenSIPS server and the server software was almost plug-and-play. The final product was a professionally assembled unit with LEDs and simply required turning it on. For the clients, an android application was developed which could be easily installed.

11) Cost: Instead of importing expensive wireless equipment, the project instead relied on cost-effective equipment locally available. Redundancy was introduced to offset some of the quality trade-offs in the solution. For example, instead of power Avila Gateworks boards with multiple Wi-Fi cards support [33], multiple Ubiquiti wireless APs [35] with multiple sector antennas were used per mesh node.

12) Availability: Avila Gateworks boards [33] were not available locally and had to be imported, which were costly and would pose problems for additional parts replacement problems down the road. Therefore, it was decided to use relatively cheap and easily available equipment, e.g., Ubiquiti Nanostations and Picosat [35] and antennas that were readily available. The trade-off was quality and performance, but redundancy (multiple APs per mesh node and more dense deployment) was used to overcome these limitations.

B. WiMesh Architecture and Overview

Figure 1 shows the WiMesh system architecture with its three main components: server module, client module, and the
backhaul mesh network. The WiMesh server module is responsible for the overall management of the WiMesh system. The server module provides mechanisms for all communications between clients and rest of the WiMesh network including audio, video, messaging, file exchange, and live geographical positioning of clients. The server also handles user registration and authentication. In addition, the server module monitors, collects, and logs user data and the performance statistics of the WiMesh backhaul network. The server module is typically hosted on a desktop machine. The WiMesh client module is installed on mobile devices carried by the users and enables multimedia communication and live geographical positioning with other clients and the WiMesh server. The WiMesh backhaul network is a multi-radio, multi-channel wireless (Wi-Fi) backhaul mesh network that forms the core of the WiMesh network. The backhaul mesh network creates a wireless multi-hop backbone infrastructure enabling mobile users, spread over large geographical regions, to communicate with each other. The WiMesh backhaul network uses the Optimized Link State Routing (OLSR) protocol, which is integrated with a custom WiMesh routing metric to provide efficient custom routing for performance improvement.

C. WiMesh Server

The WiMesh server is an important part of the WiMesh system, which handles communication among clients as well as the overall system including the mesh backhaul network. The core functionalities offered by the server include:

- Overall management of the system;
- Audio calls in broadcast mode with WiMesh clients;
- Multimedia exchanges (image, video, data files, SMS) with clients;
- Geographical position viewing of the WiMesh users;
- Registration and authentication of WiMesh clients; and
- User data and mesh network statistics monitoring and logging via WiMesh Controller.

Some of the functionalities such as audio calls, multimedia exchanges, and geographical positions viewing require interactions with the clients and, hence, are repeated in the WiMesh client modules as well. The client-side part of these functionalities will be discussed in the client module section.

Figure 1 shows the WiMesh architecture, and Figure 2 shows the main components of the WiMesh server module. The server module hosts an application server, an OpenSIPS VoIP server, and a WiMesh controller. Both the application and the OpenSIPS VoIP servers are hosted on the same machine with an underlying protocol stack consisting of various transport services. The server module is programmed in the Java programming language running on top of a Java Virtual Machine (JVM) in a Linux operating system environment. In addition to these main components, the server module also consists of a database module, graphical user interface (GUI) components, and various configurations files.

1) Application Server: The application server comprises of the different service APIs. These APIs consist of VoIP, text, file, video and the location/navigation API. In addition, the application server also contains a communication module that comprises of Mesh APIs for interfacing with the mesh network and managing the overall operation of the server. Finally, an authentication module provides client registration and authentication functionality.

Call APIs: The call APIs handle audio communication between the server and the OpenSIPS server. The audio communication supported by the application server is of two types: the real-time one-to-one VoIP call mode and the audio broadcast mode. The real-time one-to-one VoIP call is only supported on the WiMesh clients, such as mobile phones, whereas the audio broadcast mode is used on the WiMesh server by the administrator to send out a broadcast audio to all users through a radio-like functionality. For the broadcast audio messages, the administrator on the WiMesh server presses and holds a “record” button to record audio messages of up to two minutes. The audio message is encoded using Adaptive Multi-Rate audio codec and transmitted using HTTP Live Streaming (HLS) to clients.
The VoIP APIs are implemented on top of open-source Mobicents Application Server platform, which is a high-performance, scalable, and fault tolerant service level execution environment. The VoIP API implementation focuses on enabling the establishment and maintenance of audio communication between mobile users. From the implementation point of view, the VoIP API uses the Mobicents Application Server's SIP module and the SIP protocol for voice communication. A sub-module of the VoIP API allows the command and control station to communicate with the mobile users.

**Multimedia APIs:** The multimedia APIs allow sending a file or a message to users from the WiMesh server. Using these APIs, the administrator can send a message or a file to a specific user or broadcast to all users. In addition, the multimedia APIs allow streaming live video from WiMesh server to the clients. The purpose of this functionality is to facilitate monitoring of real-time ground situation.

For the file and text APIs, as soon as a text message is received from a user destined for another user, the server checks if the destination is available. If so, the message is sent and a copy is queued. In case no ACK is received within a certain time frame, the message is re-sent. Resending of message is attempted for a maximum of three times. If the destination is not available, then the message is routed to another queue for later delivery. Along with the implementation of new code, these APIs make use of the Netty open-source protocol framework. The implementation makes use of three packages of the Netty framework: HTTP and web sockets, sockets and datagram, and Large File Transfer package.

Similar to the VoIP APIs, the Video APIs are also implemented on top of the open-source Mobicents Application Server platform and uses the Mobicents Asterisk package for video communication. The Video API enables reception and transmission of video between command and control station and mobile users.

**Navigation APIs:** The navigation APIs collect location statistics of all the WiMesh clients and display these location specific statistics on a map on the server. The navigation APIs deal with locating specific users and, if needed, navigating to the required user(s) using Google maps API. The GPS module in the mobile phone gets the precise location of the user in terms of latitude and longitude and then passes this information to the Client Navigation API which passes this information to the Server Navigation API.

**SIP Server APIs:** The job of the SIP server APIs is to interface the WiMesh system with the OpenSIPS server. These APIs relay audio requests and data from the WiMesh clients to the OpenSIPS server. These APIs also relay SMS between WiMesh clients and the OpenSIPS server.

**Communication APIs:** The communication APIs provide the communication interface between the WiMesh server and the mesh backhaul, OpenSIPS, and the clients. Any communication from other components of the system to the WiMesh server takes place through these APIs. These APIs also handle the socket creation, maintenance, and destruction.

**Authentication APIs:** The authentication APIs provide user registration and authentication functionality. The WiMesh system supports two levels of security: the first level is “open access” in which no registration or authentication is required; whereas the second level requires user registration and login to the WiMesh system. The password can be saved at the client side for automatic authentication for subsequent logins.

2) **WiMesh Controller:** The WiMesh controller performs mesh network management, monitoring, and logging. It provides the following salient functionalities:

- Listing the current nodes status (up, down, restarting);
- Live statistics on routes, gateway, flags, and metric;
- Information on all the executing processes;
- Information logs such as router ID, router IP, received packets, transmitted packets, lost packets, interface ID, interface IP, and transmission queue length;
- Files and folders upload and download;
- Backup of the current state of all the network routers;
- Network restore to a previous state;
- Single screen remote shell access for fine-tuned control;
- Remote editing of the files on any router; and
- Remote modification of the radio interfaces of any router.

The WiMesh controller performs the management, monitoring, and logging of the mesh network. While there are some comprehensive and useful management tools available, the WiMesh system requirements were so diverse that no single tool met all the requirements. This motivated the development of a custom-built WiMesh controller using the Java programming language. The WiMesh controller has a user-friendly interface for monitoring the mesh network, collecting statistics, and logging network traffic. The tool uses JSch (Java Secure Channel), which is a Java implementation of the SSH2, for opening sockets to the mesh nodes. The JSch facilitates secure remote login, secure file transfer, and secure TCP/IP and X11 forwarding. The home page and the network configuration page of the WiMesh Controller can be seen in Figure 5.

3) **OpenSIPS VoIP Server:** The OpenSIPS VoIP server is an open-source, multi-functional, multi-purpose signaling SIP proxy server [31]. The application server interfaces with the OpenSIPS VoIP server to relay voice, text, files, and video among WiMesh clients and the OpenSIPS sever.

4) **The Protocol Stack:** The protocol stack is based on the Asynchronous Event-Driven Network Application Framework. The WiMesh server requires multi-threading approach to handle multiple client connections simultaneously. Research shows that the traditional multi-threading approaches perform poorly and usually incur significant overhead [36] especially for a performance driven system. Hence, the WiMesh system uses a more efficient approach, Staged Event-Driven Architecture (SEDA) [36], that avoids the high overhead associated with thread-based concurrency models and decouples event and thread scheduling from application logic. By performing admission control on each event queue, the service can be well conditioned to load, preventing resources from being overcommitted when demand exceeds service capacity.

**D. WiMesh Client**

The WiMesh client module is installed on mobile devices carried by the users and enables multimedia communication...
Fig. 3: WiMesh Controller Homepage and Network Configuration Page

The client module consists of the Mesh API, the client application, socket, and the Android stack, as shown in Figure 4.

Mesh APIs: The Mesh API interfaces with the mesh network and is used to transmit and receive client data and control packets to and from the mesh network. The Mesh API creates message headers for packets containing client information and the services required such as voice, video, or text. The client Mesh APIs are implemented to ensure the periodic sending of local information to the server including the local IP address and the GPS co-ordinates of the clients.

The client application module includes VoIP APIs, Text APIs, File APIs, Video APIs and the Navigation APIs.

VoIP APIs: The VoIP APIs provide audio communication and are implemented to allow mobile users to make voice calls between each other as well with the command and control station. The VoIP APIs use the Android SIP package.

Video APIs: The Video APIs provide video communication and are implemented to enable the transmission of captured or live video feeds to the command and control station as well as to the other users. The video APIs use the services of the Android media package for video communication.

Text and File APIs: The Text and the File APIs provide text communication and file upload/download capability between the mobile users as well as with the command and control station. Both APIs use UDP and TCP sockets along with Android HTTP package for communication. UDP socket is used for efficiency. However, to ensure reliability in the case a UDP packet is lost, code is implemented so that the destination sends an ACK for a text message received. In the case, the ACK is not received within a timer period, the packet is considered lost and the application layer asks the UDP to re-send the packet again. Reliability is also included for file transfer over the unreliable UDP. Since the packet will be relayed by the server to the destined mobile user, a custom packet is used in which the IP address of the eventual mobile user is included along with the IP of the server to which the packet will be forwarded as the next step.

Navigation APIs: The Navigation APIs periodically send GPS coordinates of the client to the server. The Navigation APIs are built on top of the Google Maps APIs.

The socket and the Android stack are already implemented modules and the aforementioned APIs will use their services through well-defined interfaces. The socket provides the interface for networking, enabling the client to communicate with other clients and the server over the mesh.

E. WiMesh Backhaul Network

The core of the WiMesh system is formed by the wireless backhaul network which creates the wireless multihop backbone, enabling mobile users to communicate with each other spread over a region. There were several aspects related to the design of the wireless mesh backhaul network. The next subsections describe the design choices made and the motivation for those choices.

1) Mesh Nodes and Associated Hardware: The project uses two different types of mesh nodes: PicoMesh node and NanoMesh node. The mesh hardware selected for the project was the Ubiquiti PicoStation M2HP and Nano-Station M2 models [35]. The PicoStation is a 600mW, 2.4 GHz, high power outdoor access point, equipped with an Atheros MIPS 24KC, 400 MHz processor, 32MB SDRAM, 8 MB Flash, one 10/100 Ethernet port, and a 6 dBi outdoor omni-antenna. Weighing a mere 100 g, with dimensions of 136x20x39 mm, and the ability to withstand extreme temperatures (-20 °C to 70 °C), the PicoStation is portable, rugged, and provides omni coverage of up to 1 km in open areas while consuming only 6 Watts. This specification makes the PicoStation a good choice
Fig. 5: A typical PicoMesh node

for use in meshing or as an access point to which mobile nodes can connect. The NanoStation is a more powerful device comprising of a 200mW 2x2 MIMO radio with an integrated dual polarity 13 dBi sector antenna, a horizontal beamwidth of 90, and providing outdoor coverage of up to 10 Km.

Every node comprises of an extensible stand (up to 25 feet), a 12 Volt, 7 AH dry battery, a small TP-Link (model TL-SF1005D) Switch, and modified Power over Ethernet (PoE) units (to enable direct connection to 12 V battery instead of 220 V DC supply) or 12 Volt PoEInjector units to power the mesh node. In addition, portable custom-made mesh-boxes are used at each node to house the various hardware components and for protection from weather conditions. The server mesh node additionally hosts the WiMesh Server in addition to the standard hardware.

The PicoMesh node comprises of two PicoStation devices connected through the switch. One PicoStation at each node operates in the access point mode and allows mobile devices to connect. The second PicoStation operates in the ad-hoc mode, runs the mesh OLSR protocol, and forms the wireless multihop backbone. The two devices are tuned to non-overlapping channels to avoid interference between the mesh backbone and the access network. The PicoStation has a relatively limited range but provides omnidirectional coverage. Hence, the PicoMesh node is used when a higher degree of meshing is required in a smaller area with problems of line-of-sight on the terrain, requiring multiple alternate routes to bypass obstacles such as dense foliage and small mountains.

The NanoMesh node comprises of two NanoStation devices and a single PicoStation device connected through the switch. Since the NanoStations provide sectorized wireless coverage, two devices are required at a minimum to enable the wireless reach to other nodes. Another noteworthy difference from the PicoMesh is that NanoMesh requires some alignment due to the sector coverage whereas the PicoMesh provides omni-coverage and does not require the deployment positions to be constrained if the mesh node is within the coverage of at least another node. As in the case of PicoMesh node, the PicoStation device is used to provide access to mobile devices since it offers omni-coverage. The NanoMesh node is the node of choice for covering large regions with as few nodes as possible. This is the typical usage pattern for rural areas or for disaster situations, in which a large area needs to be covered with a clear line-of-sight using a few nodes. Together, the two NanoStation devices can connect mesh nodes separated as far as 20 km. Figure 5 shows a typical PicoMesh node with batteries, switch, associated cabling, the WiMesh server/controller, and a mobile client. The server and the controller housed by the laptop was later replaced by Intel NUC device. The mesh-box (not shown in figure) is waterproof, rugged, and enables easy portability and deployment of the WiMesh System.

2) Mesh Performance Metric and Routing Protocol: Figure 6 shows the WiMesh backhaul network stack which consists of a Mesh Module on top of a Linux kernel running on an embedded hardware platform. The performance of the mesh module determines, to a large extent, the overall performance of the system. The primary job of the Mesh Module is to interconnect the clients and the server over a wireless multihop infrastructure. There are three main sub-components of the Mesh Module: Route Selector, Route Maintainer, and the QoS Module. While there are several areas where performance can be improved for a mesh network, two areas are particularly important from the perspective of providing QoS for multimedia traffic: route selection and route maintenance.

Resilient Route Selector: The route selection portion deals with selecting stable, high performing end-to-end communication routes in the mesh network between the source and destination pair to provide an acceptable level of quality of service (QoS) for multimedia flows. Route selection has been shown [19] to bring substantial performance improvements, particularly in the achievable end-to-end throughput. The traditional hop count metric is not a good approach [37] and a realistic metric needs to consider multiple factors pertaining link quality. In our previous contribution, the ELP metric [21] was proposed which addresses several limitations of existing
routing metrics including asymmetry of the links, radio interference, logical interference, as well as inter-flow and intra-flow interference. ELP solves the problems of asymmetry and inaccurate probe-approximation by introducing a corrective constant in the calculation to bias the estimation in favor of link loss ratios in the forward direction (data packets).

ELP explicitly incorporates the Channel Contention Interference (CCI) arising from the CSMA-CA based MAC by using the promiscuous listening mode [38]–[40] of IEEE 802.11. A node can observe the channel states using the NIC card. ELP also takes into consideration the link capacities as higher capacity links can transmit data at a higher rate and, therefore, occupy the medium for a shorter period. Low capacity links, however, take longer time and will create interference for nodes in vicinity and even for far away nodes that are outside carrier sensing range. In ELP, links with higher bandwidth (radio transmission rate) are given a lower link cost. Further, a few existing routing metrics cater to the problems which can be caused by frequently changing routing metrics which cause excessive routing overhead and frequent route shifting (or route oscillations). This is avoided in ELP in addition to offering quick recovery from failed nodes in ELP to offer a more resilient communication solution. Formally, the ELP metric for a link $e$ is defined as:

$$ELP(e) = \frac{\text{Link Loss Ratio}(e)}{\text{Link Capacity}(e)} \times \text{Link Interference}(e)$$

The ELP metric for a path $p$ is defined as:

$$ELP(p) = \sum_{e \in p} ELP(e)$$

Overall, ELP enables the selection of high performing end-to-end routes across the mesh network, enabling better performance for multimedia flows. More details about the proposed routing metric ELP can be found in [21].

**Route Maintainer:** The route maintenance component deals with maintaining end-to-end routes for the duration of the flow. In CSMA-CA based wireless multihop networks, such as IEEE 802.11, routes are fragile and often break down because of transitory interference problems on any of the intermediate wireless links [41]. These problems are particularly severe when several flows traverse the network, creating a high degree of interference. Route maintenance aims to provide stability to the end-to-end route despite the link instabilities of wireless links. Efficient route maintenance can provide substantial performance improvement in mesh networks in terms of reducing route breakages. In IEEE 802.11 based networks, the standard mandates a retry limit of seven [42]. If there are eight consecutive failed transmission attempts (one try and seven retries) to the next-hop node, then the link layer sends a failure notification to the network layer. Protocols, such as AODV, systematically translate a link breakage into a route breakage and the intermediate node notifies the source node to find new routes. The Route Maintainer mitigates this problem and considers the long-term link performance [41] to offer a resilient solution.

**QoS Module:** The main job of the QoS module is to keep an estimate of the available bandwidth in the wireless mesh network and to perform admission control for incoming flows based on the bandwidth availability on the wireless multihop path. This bandwidth management is necessary because although the system is designed to accommodate the maximum number of flows possible, too many flows can overwhelm the wireless mesh networks and cause performance degradation for existing flows. Bandwidth estimation is primarily based on a hybrid approach involving using a model as well as actual channel measurements using promiscuous listening at the data link layer to be as accurate as possible.

**F. Additional WiMesh System Components**

In addition to the main components, the WiMesh system consists of some auxiliary components. One such component is the performance evaluation tool for the multimedia traffic of the WiMesh network. Performance evaluation tools are important in designing, evaluating, and analyzing the performance of multimedia traffic of mesh-based wireless network due to the high bandwidth requirements of such traffic.

This performance tool is installed on end stations connected to the mesh network. The end stations are connected to cameras or microphones. This tool is used for streaming live multimedia traffic (audio and video) from connected clients to
the server and analyzing statistical data of these live streams in a wireless mesh network. The server aggregates data from all video/audio flows. The statistical data consist of throughput, delay, jitter, packet loss, video quality, and audio quality and can be visually viewed in terms of graphs. The server is aware of the connected clients and can separately send remote commands to each connected client to (1) select video device of the client; (2) select video codec of the feed; (3) select frames per second of the video feed; (4) select audio device; (5) select audio codec; and (6) start or stop the live feed.

The tool provides session management and data logging capabilities for scheduling experiments and storing the traffic metrics during the actual transfer of the data. The results obtained are used in the final evaluation of mesh routing protocols, metrics, and other communication related mechanisms. The tool uses UDP protocol at the transport layer. Figure 7 shows two snapshots of the performance tool session.

V. WiMesh System Deployment Experience

Several tests were carried out on both indoor and outdoor setup to evaluate the functionality and performance of the WiMesh system. More than a dozen of field trials were conducted at different times and places. Moreover, due to the diverse equipment involved, several outdoor visits were required to properly setup, power the devices, and to ensure proper interconnections between components.

A. Indoor Testbed

This section describes the experience gained during the deployment of the indoor testbed for the WiMesh system as well as the results of the experimental performance evaluation of the WiMesh system.

1) Testbed Environment: Before designing a testbed, it is essential to physically survey the site to determine its suitability to host the proposed testbed. There are several factors to consider:

- **Offer easy physical access:** The actual act of deployment involves several physical activities including installing wires, putting the mesh node in place, and securing the enclosures among others. Moreover, nodes in any testbed are known to be error-prone and it is sometimes required to remove a node or change its connections;

- **Offer easy wired access:** Despite being a wireless network, wired connection is eventually needed from a central switch to each node to facilitate controlling and manipulating the nodes remotely. This is an important constraint because installing wires require ducting, drilling, and administrative permissions. Moreover, the costs become higher as the length of deployed wires become longer. Thus, locations of nodes were selected for ease of wired installation and proximity to existing ducts where possible;

- **Offer protection from weather conditions:** Weather protection was important as the proposed testbed was not entirely in rooms, but rather in corridors. Therefore, it was necessary to select a location where devices are protected from rain and direct sunlight during summers;

- **Offer relative protection against theft:** It was important to select locations where normal students or visitors cannot easily access the device physically. This does not mean that nodes must be in a very difficult to reach place, however, the selected location should require the help of ladders and other tools to access the device. Some excellent locations had to be left out because the node would become easily approachable by visitors or students.

Based on the above factors, most of the nodes were deployed on the three-storey A-Block of the Computer Science Department of Air University, Islamabad, Pakistan.

2) Testbed Design: A 43 node testbed was deployed, which is a decent size for indoor wireless experimentation. The nodes were distributed across different departments of the Air University, however, the larger concentration of nodes (22 out of 43 nodes) was in A-Block, as shown in Figure 8 TP-Link WDR4300 routers, running OpenWRT embedded Linux firmware, based on IEEE 802.11n (MIMO) were used with data rates of 300 Mbps. The Power-Over-Ethernet (PoE) technology was used to power each of the routers since installing power sockets was impractical and dangerous.

3) Deployment of the Testbed: Deploying the indoor testbed was a long and grueling process that involved several aspects including purchase of diverse equipment, supervision of on-site labor and technicians, and administrative logistics. Table II shows the testbed specifications.

| Parameters | Description/Value |
|------------|-------------------|
| Number of Nodes | 22 |
| Covered Floors/Area | 3 Floors; Covered Area = 16000 sq. ft |
| Node Hardware | TP-Link WDR4300; Dual-Radio, Dual-Band (2.4 GHz and 5 GHz), routers with aggregate throughput of up to 750 Mbps, 2 USB ports |
| Frequency | 2.4 GHz and 5 GHz |
| Power Source | Power Over Ethernet |
| Operating System | OpenWRT |

Since the design of the complete system was well thought-out, the integration of the WiMesh system with the testbed was straightforward. A laptop or any other device was attached which was configured with the WiMesh server and the associated services and software. The attached device was used to monitor the multihop capability of the mesh testbed to see how it performed. There were some issues regarding the interoperability of the mesh testbed technologies and the WiMesh server, however, it was handled appropriately.

4) Lessons Learned: Despite comprehensive planning, closely overseeing the project, and a “hands-on” approach, a few problems arose:

- It was observed that using a monitoring software like the mesh controller was extremely useful since sometimes wireless interfaces on routers malfunctioned or had configuration settings problems, leading to routing and performance problems.
• It is important to have error-checking scripts which periodically checked active routers for a possibility of “bricking”; i.e., sometimes due to erroneous settings, the ports inside a router became internally connected in a loop and made access impossible. This was especially true when a reboot/shutdown remote call was given for routers. To mitigate this, scripts were installed which queried the routers periodically and removed those specific bridges.

• Wiring was a surprisingly difficult issue since using a star topology would have resulted in several kilometers of wires. Therefore, an efficient cabling plan was used. A related problem was that the Ethernet wire has a physical limit of carrying PoE signals for up to 100 meters. To solve this problem, switches were installed at required locations that acted as repeaters to regenerate the signals, especially between the buildings.

5) Testbed Results: Figure 9 shows the performance evaluation results of the WiMesh system on the indoor testbed. The graphs have been plotted for 95% confidence intervals to show the variation of results across different scenarios. Figure 9(a) shows the Packet Delivery Ratio (PDR) for 1-20 calls while varying the background load from 2-20 simultaneous calls. As seen in the figure, the performance of the WiMesh system stays acceptable, i.e., $PDR > 90\%$ even when a reasonable load of 40 concurrent calls was placed. There is a dense mesh deployment in the experiment leading to a significant number of alternate wireless links due to both, the large number of nodes, and the availability of multiple radios and channels. The high density and multi-channel availability result in high performance of the WiMesh system in terms of PDR. Similarly, as shown in Figure 9(b), the end-to-end delay is also within acceptable bounds, i.e., $< 100\text{ms}$. The delay becomes slightly bigger for higher number of calls since the traffic load increases compared to the lesser load scenario. Overall, the results show promising performance.

B. Outdoor Testbed

This section describes the experience during the deployment of the outdoor testbed as well as performance evaluation of the WiMesh system. This section presents the results obtained from the testbed deployed in the Fatima Jinnah park, located nearby the Air University, Islamabad, Pakistan. The park was selected as the site for most of the field trials due to geographic proximity and ease of equipment portability. However, insights obtained from field trials in other settings are also discussed.

1) Testbed Environment: For the outdoor testbed, the main concern was to ensure that performance evaluation of the system was carried out over a relatively large geographic region with varying terrain. Two nodes located on rooftops of Air University were also included in the testbed to extend the wireless range over a larger region.

2) Testbed Design: Due to logistic restrictions, the experiments were carried out for a seven node testbed. Based on varying needs of wireless antenna height, an extensible stand (up to 14 feet) was used. However, light-weight stands were used for portability. Wireless coverage was an important issue
and both PicoStations and NanoStations were used for outdoor mesh nodes depending on their utility. For mesh nodes with NanoStations, three nodes were deployed in a circular pattern to cover a wider area. A random topology was used with five nodes deployed randomly in the park while two nodes were deployed on Air University rooftops, as shown in Figure 10.

In order to ensure a relatively large distance between nodes, the average distance between nodes was around 0.75 km with some wireless links spanning 2 km. Similar to the indoor testbed, the OpenWRT embedded Linux OS was installed on the PicoStations and NanoStations. PoE was used to power the mesh nodes. This decision was even more pertinent for the outdoor testbed in order to avoid the dangers of live electricity, especially on wet terrains. Dry portable batteries were used to power outdoor mesh nodes in order to avoid battery refilling and spillage. Custom-built outdoor cabinets were used for the rooftop nodes since these installations were permanent and required protection from weather. The cabinets were installed with horizontal slits on sides for ventilation.

3) Lessons Learned: Several visits were performed to the outdoor Fatima Jinnah Park field site to test various features of the WiMesh system. The initial field trials resulted in several problems that prevented successful execution of the WiMesh system. Results were recorded only for the final field trials. Some of the problems observed during field trials and subsequent improvements to the WiMesh system were:

- When high antennas were used, they were sometimes unstable due to wind. This problem was mitigated by installing small hard pads on the end of pole legs and then embedding these legs in the ground.
- Ethernet switches sometimes malfunctioned, especially when there was even a slight difference in the applied voltage (applied ~12 volts vs. ~5–9 volts) normal range. The malfunctioning did not cause the mesh nodes to stop communicating, but rather started a never-ending cycle of disconnections and re-connections which resulted in intermittent connections to the server. This hardware fault was discovered after much difficulty as the root cause was not obvious;
- A communication problem was discovered that some mesh nodes were not aware of the existence of the server unless specifically advertised by the server mesh router.

This issue was handled by providing Host-Network Association (HNA) announcements in the routing protocol configuration. The scripts were appropriately updated;
- The range of some of the wireless hardware was extremely poor in outdoor scenarios. More notably, the Ubiquiti PicoStation performed quite poorly in outdoors for distances greater than half a kilometer;
- Sometimes, a user would be online, however, he would be shown offline. This user offline issue was due to the fact that the periodic update mechanism was flawed whereby each node was required to send data periodically to the server. The socket destruction technique was employed for detecting out of range users to solve this problem;
- There was previously no mechanism of acknowledging SMS texts, so the sender party had no information as to whether the text had been delivered or not. This problem was resolved by including notification as soon as message was successfully received at the other end; and
- When a user sent a video stream request, the sending user did not know whether the destination had accepted the request or not. This was fixed by the sender getting a confirmation from the receiver.

4) Testbed Results: The experiments were repeated under different timings and traffic load conditions. Figure 11 shows the performance evaluation graphs for 95% confidence intervals. It is pertinent to note for all metrics, the results are slightly worse than the indoor case and this is due to several reasons including long outdoor wireless links with fading, interference and even weather conditions. Figure 11(a) shows the Packet Delivery Ratio (PDR) for varying background loads vs. varying number of calls. As the results show, the performance degrades as the load increases, however, the performance still remains acceptable. Figure 11(b) shows the Packet Loss Ratio (PLR) which corresponds indirectly to the PLR and shows similar trend of increasing loss with increased load. Figure 11(c) shows the end-to-end delay which are elevated compared to the indoor scenario. Figure 11(d) shows the jitter observed for the calls, which is again considered acceptable. Overall, the results show that the performance degrades slightly compared to the indoor case, however, given the long wireless links and the outdoor environmental factors, such degradation is expected.

C. Ziarat Deployment

WiMesh system was deployed in the Ziarat mountainous region of Pakistan after the Ziarat earthquake disaster [6] [7]. This section shares WiMesh deployment experience, performance evaluation results, and the lessons learned during the Ziarat deployment.

Ziarat is a remote mountainous region in South-West Pakistan with several villages and small towns scattered over the Sulaiman mountain range stretching for hundreds of kilometers. It is sparsely populated and modern amenities such as electricity and telecommunication are sporadically available. Ziarat was hit by a catastrophic earthquake of magnitude 6.4 in 2008. The WiMesh system was deployed after the earthquake.
Apart from the problems of disaster and the subsequent rehabilitation, the major problem faced by these villagers was the fact that they had no mechanism of communication with each other and there was no cellular coverage in this region. This was particularly troublesome during an emergency situation where the men working in their gardens and fields could not be reached immediately. Before the WiMesh deployment, the mechanism that they used was that a person had to go physically to the fields and informed them of any emergency situation. However, since the mountain ran across its length and divided the residential areas from the fields, it made accessibility to villagers working in the fields even harder for any emergency situation. Another problem faced by the villages was the lack of communication between the water supply room (generator powered) and the villagers. Since the time for water for the fields was rationed among different villagers, so it would have been immensely helpful for coordination if people could somehow communicate.

These problems provided an ideal situation and a stringent test for the WiMesh deployment. In particular, two villages in Ziarat, Ahmadun and Gogi, were visited several times for practical testing and final WiMesh deployment.

1) **Deployment Logistics:** There were several aspects that were unique and challenging in deploying WiMesh to Ziarat. Perhaps the most interesting aspects were rather non-technical. The first mission was to take the villagers in confidence about the strange equipment and techies that would be roaming about. To put things into perspective, a 300 years old village named “Gogi” was selected for the deployment, which is entrenched deep inside the mountains in Ziarat with a population of around 2000-3000. It is shouldered on two sides by massive mountains and a semi-paved 20 km road is the only access off the main Ziarat road. The tribal people of this region are known through centuries for their bravery, pride, and hospitality. Fortunately, the lead author had personal contact with some people in the village to gain access for deployment. In addition, an agreement was signed with the Provincial Disaster Management Authority (PDMA), Balochistan Province of Pakistan, to become WiMesh deployment official partner and to help with logistics, regulation compliance, and other aspects of deployment. PDMA’s help in overcoming the regulatory compliance was important because the national telecommunication authority has strict regulations over the transmission power/range of outdoor Wi-Fi devices, and the WiMesh deployment plan was to use the maximum power possible in this remote region for optimal communication.

Information exchange sessions were held where the villagers shared their vows due to the earthquake and, in general, they were provided awareness about the WiMesh system (in layman terms) and how it could help ease communication problems. The village itself has a small mountain dividing the village into two lateral sections with one area covered with hectares of apples, cherry, and apricot farms while the other section contained the housing. Even though, a heavenly town (frequented yearly by the lead author), it is plagued by several problems which have worsened after the earthquake.

The deployment team was informed that one of the problems was that when men worked in the fields, the women back home could not contact them in case of emergencies or just for communication since there is no telecommunication...
coverage and electricity is rarely available especially after the earthquake. It was demonstrated to the villagers how WiMesh could help by starting with an initial deployment of two mesh nodes nearly 1.5 km apart on top of the small mountains and demonstrated to the utmost joy of the villagers how they could make calls to each other and how solar panels with batteries could be used to power the nodes indefinitely.

Other logistic concerns included portability of equipment since even the roads inside the village are tough to trek or drive on and it is not easy to carry heavy equipment.

2) Performance Results: Figure 13 shows the performance metrics for three calls in the Ziarat deployment. Figure 13(a) shows the Packet Delivery Ratio (PDR) for varying background loads. As the results show, the performance degrades as the load increases, however, the performance still remains acceptable. Figure 13(b) shows the end-to-end delay for three calls load.

3) Lessons Learned and Challenges: Some of the main lessons learned during the Ziarat deployment include:

- The WiMesh team was pleasantly surprised at the extreme hospitality of the villagers and the way they welcomed the team into their homes and their lives. Most of the deployment was straightforward and automated, however, there were still aspects of deployment that needed some manual labour such as hoisting the poles (as shown in Figure 12(b)) and fixing solar panels. Due to the prior agreements, connections, and villagers full support, the villagers were willing to provide any kind of cooperation including providing man-power for the manual labour that was very helpful in the success of the deployment.
- The WiMesh team faced several challenges with PDMA partnership mainly due to the fact that it is a pure governmental organization with its specific channels and standard operating procedures, some of which can be quite time consuming. Good deployment planning, however, helped overcome some of these limitations. The more serious problem was the fact that during the 19 months of this project, the PDMA Balochistan had seen the change of about five Director Generals. Each time, the Principal Investigator had to work from scratch on developing linkage with the new Director General, explaining to him...
the motivation and utility of the project and the progress made so far. Some were considerate and helpful, while some were not interested and this required significant effort on part of the Principal Investigator to keep them in loop and motivate them.

- The plug-and-play nature of WiMesh did make deployment quite easy, as it was realized that any kind of last minute “tweaks” and adjustments would have been impossible or extremely difficult without this feature. Moreover, sometimes the network needed to be restarted due to energy depletion (explained below) or some technical fault and it was a big advantage that simple villagers could simply press a few buttons and our automated scripts would render the system in a functioning state without any specialized intervention.

- It was realized that assumption of indefinite power supply through a combination of solar panels and batteries did not hold true since this region had mostly cloudy weather with sparse sunshine throughout the year except for the few months of summer. The region has long and harsh winters and few months of sunny summers. It was learned that solar panels still generate electricity even in the cloudy weather, albeit in significantly smaller quantities. Therefore, the project team decided to use better quality solar panels and bigger batteries, however, this came at the cost of compromising the cost-effectiveness. In addition, the team modified the routing protocol and metric to further restrict unnecessary routing messages to conserve energy. Despite all these changes, a 24/7 operation could not be guaranteed. Thus, the network was primarily used during the days. However, this limitation had minimal implication as the villagers had a natural lifestyle and slept early around 7pm in winters and 8pm in summers. The plug-and-play nature of WiMesh again helped in making sure that villagers were able to easily restart the network if it went down during the long winter nights.

- Since, the poles were of varying heights (and sometimes makeshift), the project team resorted to designing the Ethernet cables of varying lengths themselves and installing the RJ-45 connectors. Due to the custom design of the connectors and cables, sometimes these connectors would become loose due to rough handling and caused all kind of connectivity issues. This situation was exacerbated by the harsh deployment scenarios. To add to the complexity, the connectivity issues could be partial or time-varying as well, i.e., instead of a complete disconnection, we could get partial connectivity or high packet losses. Sometimes several hours would be wasted in diagnosing connectivity issues. This was particularly true for the mesh node hosting the server which could bring down the complete network. An effective and simple solution that eventually solved our problem was to use off-the-shelf Ethernet cables with professionally installed connectors, which do not come loose despite rough handling. This solved most of our connectivity problems.

- Due to the success of WiMesh deployment, the villagers requested to further extend the scope of the project and to connect the WiMesh network to a Base Transceiver Station (BTS) several kilometers away. The team investigated and did an initial feasibility analysis for this request. It turned out that there was an open-source software called OpenBTS which offered open-source BTS functionality for remote regions and has been deployed in impoverished areas of Africa [43]. The team investigated the complexities of integrating WiMesh with OpenBTS and interfacing with the commercial telecommunications provider. Due to the lack of time and sufficient resources, the team eventually decided to pursue it as a future project.

VI. GENERAL INSIGHTS AND LESSONS LEARNT

This section reports some of the salient insights, distilled from the project experience in developing and deploying the WiMesh system. A valuable contribution of this work is that different kinds of unexpected causes of failure were docu-
mented that may occur when deploying wireless technologies for emergency management, which are expected to benefit researchers undertaking such deployments in the future. In particular, the section describes how these insights add to the body of knowledge and also report where these findings are aligned with or discordant with existing wisdom. This section complements the earlier discussions in Sections V-A4 and V-B3 on the salient lessons learned during the experimentation with the indoor and the outdoor testbeds, respectively.

A. Sustainability of Wireless Based Emergency Networking

To develop a sustainable solution, it is important to factor in the unique requirements of the particular deployment area. It is well known in the ICT for development (ICTD) community that it is a real challenge to keep a rural wireless network running for long-term. Most projects fail to go beyond the pilot stage [44] and techniques such as resource pooling [43] can help. Since WiMesh is low-cost, rugged, and customized for the environment, therefore, the system is still operational in some villages, which supports its sustainability aspect.

B. Power Related Issues

Apart from the lack of a sustainable financial model, the most common operational cause underlying the lack of sustainability of most wireless networking projects in the rural areas is the high cost of dealing with the poor quality of power, which is a very common issue in rural areas of the developing world [44]. The poor power quality also leads to higher failure rates of components, which means that “the real cost of power is not the grid cost, but is the cost of overcoming poor power quality problems” [44]. This is because low-quality power necessitates an additional burden of deploying additional infrastructure such as the use of power controllers, batteries, and solar-powered backup power solutions.

C. Anticipating Malfunctions and Fault Diagnosis

For building a sustainable solution, it is important to have the ability to anticipate malfunctions and diagnose faults. A number of authors have written about simplifying and automating fault diagnosis, and interested readers are referred to [46] and [47]. Subramanian et al. [48] have shown that low-cost community wireless networks can malfunction in a myriad ways from complete failures (such as hardware board failures, corruption of the flash memory cards, and lightning strikes) as well gradual degradation over time (such as interference from external sources, effects of unreliable power, and rainfall).

As noted in Sections V-A4 and V-B3 it was also discovered through this project experience the great utility of having a featured monitoring software since it is very common for wireless interfaces on low-cost devices to malfunction (often due to imperfect power supply). This malfunction then leads to knock-on problems such as intermittent performance, cyclic loop of disconnections and reconnections, or routing issues that are hard to troubleshoot back to the offending cause.

D. Open-Source Community Software Contribution

We believe that it is necessary for the wireless networking for emergency community that we develop an ecosystem that encourages sharing of insights and the communal reuse of basic building blocks, which practitioners and researchers can use to develop customized solutions more efficiently, more quickly, and with lesser resources. In the same spirit, we have leveraged existing open-source solutions where available, and are releasing the custom-developed source code of the WiMesh system, as well as detailed instructions and usage guidelines. These resources can be accessed publicly at the website of the WiMesh project (https://sites.google.com/site/wikiwimesh/).

VII. ACKNOWLEDGEMENT

The authors would like to extend their thanks to the National ICT R & D Fund Pakistan for providing the financial support towards this project.

VIII. CONCLUSION

This paper presented the WiMesh disaster communication system including its conception, design, implementation and field trials. The WiMesh system leverages the wireless multi-hop capability of the mesh network and the Wi-Fi accessibility of mobile devices to connect to the nearest mesh node. Due to the nature of their application domain, disaster communication systems need to have specific traits of size, portability, power, cost, customization, ease of installation, wireless coverage, and availability of hardware and software. Some of these traits, such as cost and availability, are especially critical for deployment in developing and third world countries. These features were considered important during the design and development of the WiMesh system. Furthermore, as discussed in the paper, the WiMesh system meets all of these criteria. The paper provided architectural and design details of the WiMesh system including design trade-offs, pitfalls, and challenges faced during the design, development, and deployment. Extensive indoor and outdoor field trials were carried out to demonstrate the functionality and to collect performance metrics for the WiMesh system. Details of these field trials along with performance metrics results were also presented in the paper. These field trial results demonstrate that the WiMesh system can handle several dozen concurrent VoIP calls without noticeable performance degradation, even in the presence of significant background traffic. There are a number of areas that can be further improved. For instance, 5G technology is making inroads in the wireless ecosystem and it would be important to leverage that strength into the WiMesh solution.

REFERENCES

[1] P. Meier, Digital humanitarians: how big data is changing the face of humanitarian response. Routledge, 2015.
[2] J. Qadir, A. Ali, R. ur Rasool, A. Zwither, A. Sathiaseelan, and J. Crowcroft, “Crisis analytics: big data-driven crisis response,” Journal of International Humanitarian Action, vol. 1, no. 1, p. 12, 2016.
[3] O. Onireti, J. Qadir, M. A. Imran, and A. Sathiaseelan, “Will 5G see its blind side? evolving 5G for universal internet access,” in Workshop on Global Access to the Internet for All, pp. 1–6, ACM, 2016.
