

ReDiCom: Resilient Communication for First Responders in Disaster Management

Jiachen Chen*, Yuxuan Xing[†], K.K. Ramakrishnan[‡], Mohammad Jahanian[‡], Hulya Seferoglu[†], and Murat Yuksel[§]

*WINLAB, Rutgers University [†]University of Illinois at Chicago

[‡]University of California, Riverside [§]University of Central Florida

I. INTRODUCTION

Effective communication among first responders during and in the aftermath of a disaster can affect outcomes dramatically. We seek to build a resilient architecture that allows first responders to communicate even with: 1) damage to infrastructure — civilian and / or specialized communication facilities may be damaged by the disaster, and 2) dynamically formed groups — first responder teams may be formed dynamically in response to a disaster and team member addresses (*e.g.*, phone numbers, network addresses) may not be known to one another. We propose a resilient network architecture that allows efficient communication among first responders during and after a disaster [1]. We seek to support dynamically formed groups for incident response, allowing first responders to securely and conveniently communicate based on roles (names). The architecture supports communication in disasters by 1) building resilience into the framework across all the layers, 2) creating a framework that allows communication by role and identity, rather than addresses, 3) supporting multiple modalities (data, voice) for communication among dynamically formed first responder teams, and 4) providing robust and resilient communication and computing even when facilities are error- and disruption-prone.

In our demo, we will show an emulated disaster management situation with a remote command center, and a data mule that shuttles between the command center and a shelter. We will demonstrate the resilient network architecture, including: 1) propagation of the namespace across fragmented networks, 2) device-to-device (D2D) communication software that can utilize heterogeneous wireless links (*i.e.*, Bluetooth and WiFi Direct), and 3) coded cooperative computing mechanisms in heterogeneous and time-varying environments.

II. DEMO ARCHITECTURE

To achieve the above functions, we designed a layered architecture to allow for flexibility in adapting each layer as needed to support different functionality at the lower layers (*e.g.*, different network layers and different link layers). From bottom to top, the architecture has the following layers (see Fig. 1):

The **link layer** exploits D2D communication to complement infrastructure-based communications. It provides a generic “link” abstraction to the upper layer, providing connectivity no matter what the underlying technology is: WiFi, WiFi-Direct, Bluetooth or infrastructure-based cellular or wired network connections. The identity in this layer is `NeighborID`, a variable-length device ID.

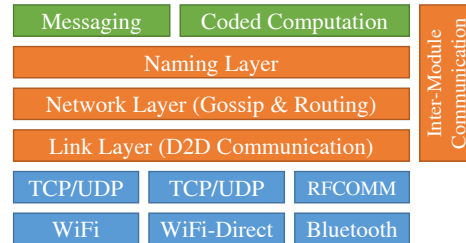


Fig. 1: Architectural view of ReDiCom.

The **network layer** supports data dissemination using flat identifiers over both connected and delay-tolerant links. It provides a “connected network” abstraction to the upper layer. The identity in this layer is `Name` (a fixed-length flat identifier independent of the location, similar to MobilityFirst [2]).

The **naming layer** maintains a graph-based namespace and performs name expansion for handling publications (similar to the protocol in [3]). It provides an abstraction of “connected names” to the applications. The identity in this layer is also a `Name`, but with graph-based relationships among them.

Applications are supported by the **application layer**. A messaging app allows first responders to communicate using the graph-based namespace. A coded computation app allows first responders to perform compute-intensive tasks with the assist from nearby *helpers*.

To assist the communication between layers, we implemented an **inter-module communication** component which mimics the `Context`-based broadcast in Android, but with lower overhead and no limitation on the object size.

III. DEMO TOPOLOGY AND SCENARIO

We assume a disaster happens requiring first responders to assemble teams for search and rescue. We have two places: a coordination center and a shelter which does not have communication infrastructure support (see Fig. 2). At the coordination center, we have an `Incident Commander` that takes control of managing the disaster and a `Dispatcher` that dispatches units upon instructions from the commander. In the shelter, we have several first responders. All the connections in the shelter are based on D2D communication via either WiFi Direct (orange connections) or Bluetooth (blue connections). A `Patrol Car` travels between the coordination center and the shelter to carry messages around, acting as a data mule. We emulate the patrol car movement by manually connecting (and disconnecting) the device with (and from) the `Coordination Center` or `Rescue 1` in the shelter.

A. Messaging

To setup a search and rescue team, the `Incident Commander` can choose to create a set of names based on predefined

