E-DARWIN: Energy Aware Disaster Recovery Network using WiFi Tethering

Mayank Raj*, Krishna Kant†, Sajal K. Das*
*Missouri University of Science and Technology †Temple University
E-mail: {mayank.raj,sdas}@mst.edu kkant@temple.edu

***Abstract*—In this paper, we propose a novel architecture called Energy Aware Disaster Recovery Network using WiFi Tethering (E-DARWIN). The underlying idea is to make use of WiFi Tethering technology ubiquitously available on wireless devices, like smartphones and tablets, to set up an ad hoc network for data collection in disaster scenarios. To this end, we design novel mechanisms, which aid in autonomous creation of the ad hoc network, distribution of data capturing task among the devices, and collection of data with minimum delay. Specifically, we design and implement a distributed coalition formation game for distributing the data capturing task among wireless devices based on their capabilities, available energy, and network participation for higher network lifetime. Finally, we evaluate the performance of the proposed architecture using a prototype application implemented on Android platform and large-scale simulations.**

***Keywords*—energy adaptive computing, wifi tethering, disaster recovery network, ad hoc network**

I. Introduction

In the event of a disaster, immediate influx of health, environmental and infrastructure data is essential for assessing the conditions in the affected region, coordinating rescue efforts, and addressing the needs of the affected community. With the proliferation of wireless devices, like smartphones and tablets, they can be assumed to be abundantly available among the people in the affected region and can act as valuable resources in rapid establishment of disaster recovery network for data collection and analysis followed by important decision making. For example, after the Haiti earthquake in 2010, there were approximately 2.8 million active mobile subscribers out of 10 million inhabitants contributing data for tracking the movement of population in the affected region [1]. A network infrastructure based on such a large number of pre-existing devices could address the coverage and connectivity needs of the disaster recovery network under consideration.

To this end, this paper proposes a novel architecture called *Energy Aware Disaster Recovery Network Using WiFi Tethering* (E-DARWIN) for creation of the desired network infrastructure using wireless devices. When using WiFi Tethering, a wireless device can either act as a WiFi hotspot or client. Only when a device acts as hotspot, other devices in its communication range, i.e., its neighbors, can communicate with one another through it. Thus, to facilitate communication in the network, each device randomly takes up the role of WiFi hotspot. Specifically, we let the devices behave autonomously, wherein they discover and synchronize with one another's schedule so that they can act as WiFi client only when one of their neighbors acts as hotspot. When acting as a hotspot, the device and its clients select from among themselves a relay device and offload data to it. The relay device stores the data until one of its neighbors acts as a hotspot, wherein the data is offloaded to another device selected as the relay device. Data is stored and forwarded by the relay devices, until it can be delivered to a remote emergency command center (ECC). Furthermore, at each step, the devices select the relay device such that data is delivered to the ECC with minimum delay. Additionally, wireless devices are increasingly being equipped with multi-modal sensors, such as temperature, accelerometer, pressure, GPS, microphone and camera and hence, can act as rich sources of sensory information in disaster scenarios [2]. To utilize these rich capabilities, we design and implement a distributed coalition formation game, which aims at distributing the data requirements of the network among wireless devices based on their capabilities, available energy and network participation for higher energy efficiency. We implemented the proposed architecture on Android platform and evaluated the performance of the proposed solution using the implemented prototype and large-scale simulations.

The rest of the paper is organized as follows. We summarize the related work in Section II and introduce the proposed architecture in Section III. In Section IV, we discuss the network initialization and data forwarding mechanisms followed by the energy aware distributed coalition formation game in Section V. Section VI presents an experimental analysis of the proposed solution. Finally, we conclude the paper in Section VII and discuss directions for future work.

II. Related Work

Using sensors for constructing disaster recovery network and capturing data has been extensively explored in literature. However, making use of wireless devices, like smartphones and tablets, widely available among people in the affected region is a more practical approach [3]. Even though the solutions discussed in literature utilize the ad hoc mode of operation in wireless devices to construct disaster recovery network, implementing it on current wireless devices requires modifications of the transceiver driver firmware, root access to kernel or jailbroken devices [4]. Evidently, this is undesirable, as these solutions are not seamlessly supported across all devices and are illegal in many countries. This motivated us to explore the use of technologies available on the wireless devices, i.e. WiFi Direct and WiFi Tethering, for constructing disaster recovery network.

WiFi Direct or WiFi Peer-to-Peer (P2P) is a standard, which allows devices to communicate with one another without the need of a wireless access point. Wi-Fi Direct works by embedding a limited wireless access point in the devices, and using the Wi-Fi Protected Setup system to negotiate a link. The feasibility of using WiFi Direct to create an ad hoc network in disaster scenarios has been shown in [5]; however, the technology has its limitations. When using WiFi Direct, a device is connected to another device acting as a limited AP. Moreover, not all devices have the capability to connect to multiple hotspots or APs at the same time, i.e., cannot pair with multiple devices simultaneously. This limits the usability of

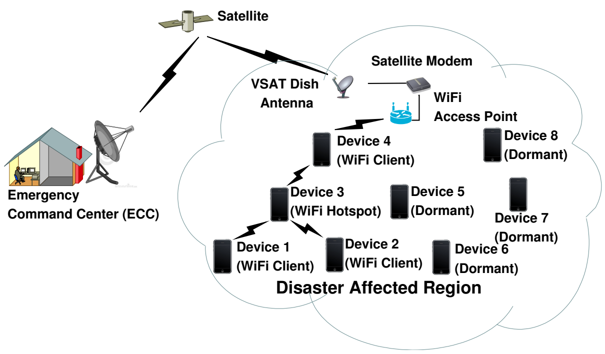


Fig. 1: Network Architecture of E-DARWIN

the technology in supporting a robust and connected network. Furthermore, the technology is not available on all devices and is more energy-exhaustive than WiFi Tethering [6].

WiFi Tethering technology is ubiquitously supported by all major device manufacturers and is available across all major OS platforms, such as iOS 4.3+, Android 2.2+ and Windows 7.5+. It allows a device to act as a WiFi hotspot, which can be used to facilitate communication between its neighbors. A characterization of the energy consumption of a device using WiFi Tethering has been presented in [7]. Based on the observed behavior, the authors proposed an energy efficient mechanism to further improve the usability of WiFi Tethering technology on mobile devices. Furthermore, WiFi Tethering has been used in the context of opportunistic networks [8] and creation of a computing cluster grid [9], wherein the devices utilize it to establish communication with their neighbors. Additionally, in [6] and [10], the authors propose the use of WiFi Tethering to construct an ad hoc and mobile ad hoc network respectively. In [10], the authors use network virtualzation to let devices simultaneously assume the roles of hotspot and client. However, virtualization of the network interfaces is not supported in current mobile devices. Furthermore, the authors in [6] and [10], do not address issues related to network formation, routing and maintenance. To the best of our knowledge, no other work in literature investigates the construction of an ad hoc network for disaster scenarios using WiFi Tethering and routing of data in it.

III.E-DARWIN Network Architecture

We assume the network infrastructure has been destroyed in the affected region and needs to be set up primarily using local resources, i.e., wireless devices. The proposed architecture seamlessly integrates wireless devices into a robust ad hoc network irrespective of the fact whether they can or cannot connect to multiple hotspots. In this section, we state our assumptions, introduce the various components in the proposed architecture and discuss their functioning.

A. Network Components

As depicted in Figure 1, the central component of the disaster recovery network is a remote Emergency Command Center (ECC), which receives the captured data from the disaster affected region, analyzes them and coordinates rescue efforts accordingly. We assume that the ECC becomes operational soon after the occurrence of the disaster. The disaster recovery network can be described as being composed of various connected components primarily formed by wireless devices available with the people in the affected region. The devices capture, store and forward the data to the ECC. We assume the devices having high storage capacity, ranging from a few gigabytes to 512 GB, are capable of temporarily storing the captured data. In each component, connectivity to the ECC is provided by deploying WiFi access points (APs) with satellite connectivity, as shown in Figure 1. We assume that satellite connectivity is provided by using the APs in conjunction with satellite gateways. Each satellite gateway is composed of a very-small-aperture terminal (VSAT) dish antenna and a satellite modem, which can be easily assembled and disassembled for portability [11]. We assume the satellite gateways with support for data rates up to 50 Mbps [12] can satisfy the backhaul capacity requirement of the network. Wireless devices forward data to the nearest WiFi AP with satellite connectivity in the component, which in turn relays them to the ECC over the satellite link. We assume when road connectivity is available, the WiFi APs are deployed in the affected region by emergency vehicles such that each device in a component can communicate with at least one WiFi AP using the proposed solution. If no road connectivity is available to the affected region, these equipments can be carried or airdropped in the affected region by emergency crews [11] and deployed similarly. Furthermore, we assume that all devices have the prototype of the proposed solution installed, which comes in to play when the initial exodus of people has already occurred and there is no significant mobility of the devices.

B. Functioning of E-DARWIN

In the proposed architecture, the devices behave autonomously, wherein they discover and synchronize with one another and forward data to the WiFi APs with minimum delay. To enable this, each device in the network can be in only one of the three states described below at any given instant of time.

Dormant - A device by default stays in the dormant state to conserve energy, until it is scheduled to act either as a WiFi hotspot or client. In this state, all the network interfaces of the device are disabled and the entire device is in a low power mode with the CPU sleeping.

WiFi Hotspot - Each device in the network becomes a WiFi hotspot consecutively after a random time interval. In this state, the device is responsible for selecting the relay node and facilitates offloading of data to it. Each device acts as a hotspot for a predetermined time interval and then enters the dormant or client state. If the time when a device is scheduled to be in hotspot state overlaps with another state, the device will always enter or continue to be in hotspot state, as its role as hotspot is essential for facilitating communication between its neighbors and offloading the data.

WiFi Client - Using the mechanism discussed in Section IV, each device in the network synchronizes with its neighbors and schedules itself to act as WiFi client whenever one of its neighbors acts as hotspot. In this state, the device periodically scans the wireless channel for advertisements from the hotspot and associates with it. On successful association, the client devices connected to the hotspot participate in selection of the relay device and offload data to it. A device stays in client state as long as the hotspot is active or until it is scheduled to act as hotspot, whichever is earlier.

A device can capture data while it is in hotspot or client state. If a device is scheduled to capture data in dormant state, it wakes up to do so and becomes dormant again.

Figure 2 depicts the functioning of the E-DARWIN architecture corresponding to the network deployment shown in Figure 1. We denote each Device i, $\forall i \in \{1, 2, \ldots, 8\}$ in

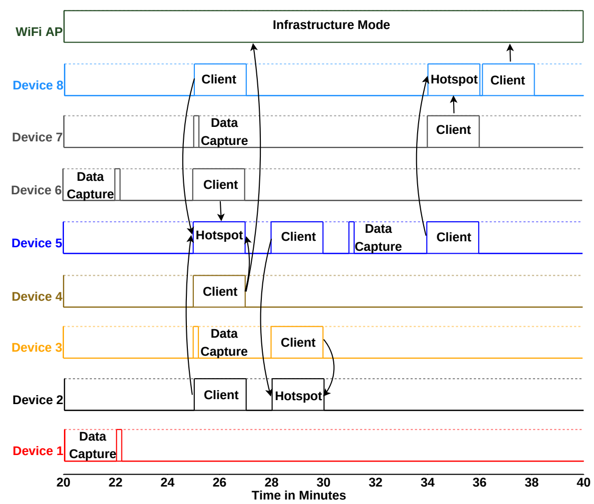


Fig. 2: Functioning of E-DARWIN

Figure 2 as D_i. We assume the devices have synchronized with their neighbors and have scheduled themselves to act as WiFi client whenever one of their neighbors acts as hotspot. Let us assume D_5 is scheduled to act as a WiFi hotspot at $t = 25$ mins with devices D_2, D_4, D_6 and D_8, which are in its communication range, waking up as WiFi clients and associating with it. We assume D_4, which is also in the communication range of WiFi AP, is connected to both D_5 and WiFi AP simultaneously as it can connect to multiple hotspots at the same time. Since, D_4 can forward data from D_2, D_4, D_6 and D_8 to the WiFi AP as soon as it receives them, it is selected as relay node. Now D_8 is also in the communication range of the WiFi AP and D_5. However, assuming the capabilities of D_8 is limited, i.e., it can only connect to one hotspot at a time, it will only associate with D_5 at $t = 25$ mins. D_8 can collect data from other devices and when D_5 ceases to be a hotspot at $t = 27mins$, it can continue as WiFi client and associate with the WiFi AP to deliver the data. Thus, D_4, which can deliver data to the WiFi AP first is chosen as the relay node. If D_2, D_5, D_6 and D_8 have any stored data, they forward the data to D_4. Assuming, the devices were previously configured to act as WiFi hotspot for 2 mins, the devices stay in their respective state till $t = 27$ mins and then become dormant. At the next time instant, i.e., $t = 28$ mins, D_2, D_3 and D_5 wake up with D_2 acting as a WiFi hotspot. Assuming D_5, which is also in the communication range of D_8, knows the schedule of D_8. Hence, it can advertise that since D_8 is supposed to act as WiFi hotspot after 6 mins, i.e. at $t = 34$ mins, data can be forwarded to the WiFi AP through D_8 after 8 mins, when it ceases to be hotspot. Since, no other device has the opportunity to deliver data to the WiFi AP before D_5, the latter is selected as the relay device. If devices D_2 and D_3 have any captured data, they forward them to D_5. At $t = 34$ mins, when D_5, D_7 and D_8 wake up with D_8 acting as a WiFi hotspot. D_5 which had previously received data from other devices, forwards the stored data to D_8, which is now the relay node. At the end of the hotspot state, D_8 delivers the data to the WiFi AP by becoming WiFi client and associating with the WiFi AP. D_1, D_3, D_5 and D_6 wake up from dormant state whenever they are scheduled to capture data and then become dormant again.

IV. Network Initialization and Data Forwarding

On activation of the E-DARWIN application, the devices discover their neighbors and synchronize with one another's schedule. After initialization, the devices forward data to the WiFi AP using relay devices such that they are delivered with minimum delay. To this end, we describe the process of autonomous neighbor discovery and synchronization as well as forwarding data in this section.

A. Network Initialization

Existing solutions in literature for neighbor discovery and synchronization rely on coordination between the devices to ensure their radios are simultaneously turned on for discovering one another. However, to allow two devices using WiFi Tethering to discover one another, they must not only have their radio turned on but be in different states, i.e., they should act as WiFi hotspot and client respectively. To address this problem, we restrict the devices to stay in client state during initialization, i.e., for $T_{initial}$ time, and successively take up the role of hotspot after a random time interval, that is uniformly distributed between $(0, T_{discover})$. During initialization, when a device is in hotspot state, it stays in that state for a predetermined time interval ($I_{discover}$) and periodically broadcasts beacon frames to advertise its presence, capabilities and configuration. All devices in client state scan the wireless channel periodically every T_{scan} time interval for advertisements from the hotspot. On discovering and associating with a hotspot, the devices inform one another when they will act as hotspot next. On receiving the information from its neighbor, each device schedules itself to act as WiFi client after the specified time interval. After $I_{discover}$ time interval, the hotspot device returns to client state. By keeping $T_{initial} \gg T_{discover}$, we let the devices become hotspot multiple times in the initialization phase and reduce the probability that they will not discover one another if they become hotspot at the same time.

Let us assume after initialization, the devices become hotspot successively after a random time interval, that is uniformly distributed between $(0, T_{max})$. After initialization, the devices stay as hotspot for I_H time to allow neighboring devices to associate with it, select the relay node, and offload data as discussed in Section IV-B. In order to allow a device to discover its neighbor, which is already initialized, we must keep $T_{initial} > T_{max}$ to ensure that such a neighbor becomes hotspot at least once within the initialization period and the devices discover one another. However, the device may still not discover the neighbor as the two may become hotspot at the same time. We assume that $I_H \gg I_{discover}$, as a device, which is already initialized, must also perform additional functions as mentioned earlier. Thus, even though the two devices may become hotspot at the same time, the device in the initialization phase exits the hotspot state first and becomes client after $I_{discover}$ time to discover and synchronize with its neighbor. Thereby, the proposed mechanism allows devices to discover one another and synchronize their schedules autonomously, irrespective of the time they join the network.

B. Forwarding Data in E-DARWIN

The devices start capturing and forwarding data to the WiFi AP after the initialization phase. The problem of routing data in ad hoc network has been extensively studied in literature, wherein the devices select a neighbor to forward data based on a given route selection algorithm. In E-DARWIN, devices offload data to another device selected as the relay node until they are delivered to the WiFi AP. However, the relay device may not be a neighbor of the device offloading the data as they

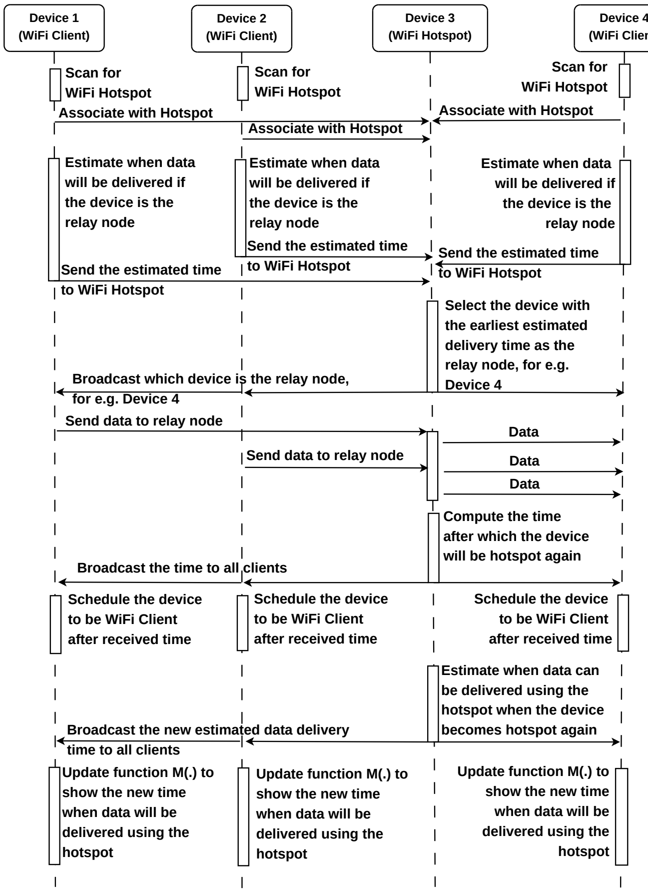


Fig. 3: Data Forwarding in E-DARWIN

can be connected through the hotspot and not be aware of one another. Additionally, the relay device must be selected such that the data can be delivered to the WiFi AP earliest based on the schedule of the devices. To enable this, each device maintains a function M, which stores the information on how soon the data can be delivered to the WiFi AP using a relay device connected to one of its neighbors acting as hotspot. Figure 3 describes how the function M is maintained and used to forward data to WiFi AP.

On being configured as WiFi client, each device scans the wireless channel for advertisements from its neighbors scheduled to act as hotspot then. After discovering and associating with the hotspot, each client device estimates when data can be delivered to the WiFi AP using it as the relay device and informs the hotspot of its estimated delivery time. If the client device has the WiFi AP as its neighbor and is capable of connecting to multiple hotspots at the same time, it can connect to both hotspot and WiFi AP simultaneously and forward data to the WiFi AP as soon as it receives them. Hence, the client device estimates its delivery time as 0. On the other hand, if the client device has WiFi AP as its neighbor but can only associate with one hotspot at a time, it can deliver data to the WiFi AP after I_H time when it disconnects from the hotspot and associates with WiFi AP. Hence, it estimates its delivery time as I_H. Additionally, if the WiFi AP is not a neighbor of the client device, it implies that the device will forward data to the WiFi AP using another device as relay device when either it or one of its neighbor acts as hotspot. To this end, the client device chooses its estimated delivery time as the minimum of the estimated delivery time of its neighbors using the function M. The hotspot on receiving all the delivery estimates from its clients, selects the relay node as the device using which data can be delivered to the WiFi AP first and announces it. The client devices on receiving the information, offload all the stored data to the relay device.

Meanwhile, the hotspot determines when it will act as hotspot again (T_{next}) and schedules itself to act as hotspot then. Additionally, the mapping function of the clients needs to be updated to reflect the estimated delivery time when the device acts as hotspot again. If one of the clients is connected to WiFi AP, then data can again be forwarded to the WiFi AP as soon as the device acts as hotspot again. Hence, the delivery time is estimated as the time when the device will become hotspot again, i.e. T_{next}. If the client cannot connect to more than one hotspot at a time, it can deliver the data to the WiFi AP after it disassociates from the hotspot next. Hence, the delivery time is estimates as $(T_{next} + I_H)$. If none of its clients are in the communication range of the WiFi AP, then the hotspot must forward data to the ECC through one of its clients acting as a relay device. Therefore, the hotspot determines after T_{next} time, which of its clients can deliver data first based on their previously communicated delivery estimates and announces it as its estimated delivery time to them. The clients on receiving the information, update their mapping function accordingly and set themselves to be client again after T_{next} time. The clients in turn propagate the delivery estimates in the network through other neighbors acting as hotspot.

V. Energy Aware Distributed Coalition Formation

In this paper, we aim at exploiting the multi-modal sensing capabilities of the wireless devices and use them for capturing data too. The problem of data capturing and gathering has been extensively studied in literature, especially in the context of wireless sensor networks. The solutions discussed in literature primarily perform in-network data aggregation or compression to reduce the communication cost in the network or use model driven approaches to reduce the amount of data being forwarded in the network by suppressing redundant information [13]. However, these approaches require frequent interaction between the nodes and flow of information in the network, like parameters for predicting the behavior of the random process and error estimates. In the E-DARWIN architecture, the devices must wait for one of their neighbors to become hotspot to forward data. Thus, the flow of information in the network is restricted by the frequency with which the devices become hotspot, which makes these traditional approaches infeasible when the frequency is kept low to conserve energy. In this paper, we propose a distributed coalition formation game, which aims at suppressing the data capturing task at source by exploiting the spatial correlation between the devices and discuss its implementation.

The data requirements of disaster recovery network can vary from collecting spatial data, like GPS, for tracking the affected population [1], temperature and humidity readings from bluetooth based acquisition systems for environmental assessment, text based inputs for nutritional and health assessment [2,14], audio samples captured from the environment for detecting trapped victims [15] as well as capturing video and images to build a spatial view of the affected region for infrastructure assessment [2]. Each data requirement represents a real time process that the ECC wants to monitor.

Let us represent the data requirements of ECC by the set $X = \{x_1, x_2, \ldots, x_n\}$. For each process $x_i \in X$, the ECC defines points of interest in the affected region and wants devices in their vicinity to collect samples of data and send them with frequency f_i. The ECC collects samples of each data over a decision interval (τ) and then analyzes them to detect events and make decisions accordingly. Let us assume, on initialization of each device, the ECC informs it about the set of physical processes X it wants to monitor and the frequency f_i with which it wants each data source in the network to capture samples of data for each process $x_i \in X$. The ECC broadcasts the data requirements to all WiFi APs through the satellite link, which in turn broadcast them to all the devices connected to them. The devices in turn propagate the information in the network by broadcasting it to their clients when they act as hotspot or through a neighboring device acting as hotspot. On learning about the data requirements, each device determines the subset of data requirements it can capture and starts capturing and sending samples of them to the ECC with the desired frequency. However, not all devices are required to participate in the data capturing process, as samples from sources with high spatial correlation are redundant. This provides a set of highly correlated devices with the incentive to collaborate with one another and act as a single data source, i.e., collectively contribute data with frequency f_i, for reduced energy consumption. In [16], the authors define a distortion function $D_{x_i}(S)$ to measure the difficulty in estimation of events in a physical process x_i from samples captured by spatially correlated sources S and is used in this paper to represent the redundancy among the samples. Thus, for a given process $x_i \in X$, the ECC can accurately detect events from samples as long as the level of distortion $D_{x_i}(S)$ or redundancy in the samples from the set of devices S is below a threshold $D^*_{x_i}$. Hence, a set of highly correlated devices whose samples are distorted above the threshold $D^*_{x_i}$, can collaborate and form coalitions to share the data capturing task among them for reduced energy consumption.

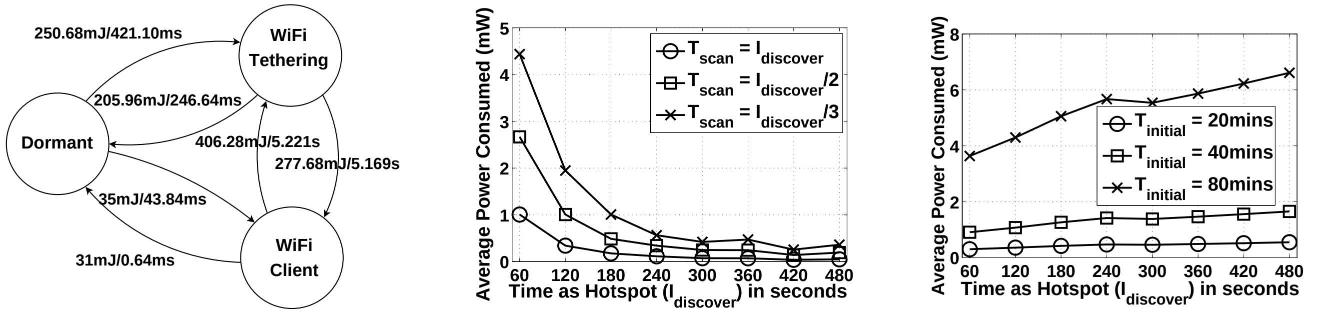
However, for the devices to form coalitions or join them, we must provide them with a method to evaluate and compare different coalitions and decide which coalition to join. To this end, we define the utility of a coalition. In coalition game theory, the utility of a coalition represents the worth of a coalition and is distributed among the devices in the coalition based on a solution concept. By relating the utility to the frequency with which a device captures data, we provide the devices with a way to evaluate, compare and decide which coalition to join. In this paper, we define utility based on the following premise: the allocation of data capturing task should prefer devices with high available energy and low network participation. Let us assume, a device j allocates a fraction $E^{x_i}_j$ of its available energy (E_j) for capturing samples of the data $x_i \in X$, such that $\sum_{x_i \in X} E^{x_i}_j = E_j$. However, the available energy of the device will also be used for network operations. When acting as hotspot, the amount of energy a device consumes increases with the number of neighbors because more neighbors mean a higher volume of traffic flowing through it to the relay node. Additionally, the amount of time a device spends as client depends on the number of neighbors it has and the frequency with which its neighbors become hotspot. Given that all devices become hotspot with the same frequency, a device with more neighbors will switch to client state more frequently and spend more energy in network operations. Therefore, devices with more neighbors should be allocated a lesser burden of capturing data for higher network lifetime. Hence, when determining the contributions of a device j in a coalition, we must divide the total energy allocated by the device for capturing samples of a process by the number of neighbors N_j it has. The total utility of a coalition represents the total weighted available energy of the devices in a coalition for capturing data and is given by Equation 1. Therefore, by distributing the total utility among the devices in the coalition, we are essentially determining how much data capturing task should be allocated to each device in the coalition. To distribute the utility among the devices in the coalition, we define a payoff vector $\phi^v = \{\phi^v_1, \ldots, \phi^v_{|S|}\}$ that represents the distribution of the utility $v(S)$ among the set of devices S. We propose a simple and strict method of distributing the total utility among the devices, i.e. Proportional fairness, wherein the total utility is distributed among the devices in the ratio of their contributions, as shown in Equation 2, where $\sum_{j \in S} w_j = 1$ and within the coalition $\frac{w_i}{w_j} = \frac{v(\{i\})}{v(\{j\})}$. Hence, the ratio of utility that each device $j \in S$ in a coalition receives is representative of the ratio of data it should capture, i.e. $w_j f_i$. Furthermore, by keeping the allocated energy for capturing samples of each data independent of one another, the algorithm can be executed independently for each data requirement $x_i \in X$.

$$v(S) = \begin{cases} \sum_{j \in S} \frac{E^{x_i}_j}{N_j} & \text{if } D_{x_i}(S) > D^{\star}_{x_i} \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

$$\phi^v_j = w_j v(S), \forall j \in S \tag{2}$$

A. Formulation of Distributed Coalition Formation Game

In this section, we define a transferable utility (TU) coalition formation game using which devices in the network can form coalitions. Let us define a coalition game $G = (D, v)$, where D is the set of players representing each device and v is the utility function given by Equation 1. Coalitions in the network can be formed with a centralized approach; however determining the optimal coalitions is NP-hard [17], as it requires iterating over all possible coalitions of D devices. Thus, a distributed coalition formation approach is more desirable. In [17], the authors provide a generic framework for forming coalition games among players using two simple merge and split rules, which can be applied in a distributed manner. However, before we adapt and apply the framework to our problem, we must define a comparison relation to provide players with a way to compare coalitions for joining or splitting from them. Among several well known comparison relations discussed in literature [17], the Pareto order correctly captures the properties of the game. The Pareto order states that devices in coalitions have incentive to leave, join or form new coalitions if at least one device can get a smaller allocation of frequency of capturing data without increasing that of others, resulting in lower energy consumption for the device. Based on the above formulations, we define rules for merging and splitting of the coalitions as follows. The merge rule specifies that if at least one device in the coalitions $\{C_1, \ldots, C_l\}$ can achieve a lower frequency of capturing data by merging them, the devices will merge and form a single coalition $\{\cup^l_{z=1} C_z\}$. Similarly, the split rule specifies that if



(a) Energy and Time Consumed in State Transitions (b) Average Power Consumed in Client State (c) Average Power Consumed in Hotspot State

Fig. 4: Characterization of Power/Energy Consumed in Initialization Phase

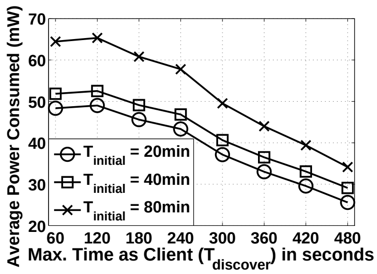


Fig. 5: Average Power Consumed in Initialization Phase

one of the devices in the coalition $\{\cup_{z=1}^{l} C_z\}$ can achieve a lower frequency of capturing data by splitting the coalitions into smaller coalitions, the devices will split and form smaller coalitions $\{C_1, \ldots, C_l\}$. In [17], the authors show that from any arbitrary starting point, random iterations of the merge and split rule will always terminate in finite steps to stable coalitions.

B. Implementation of the Coalition Game

On receiving the data requirements, each device creates a coalition with only itself in the coalition. Whenever a device acts as hotspot, all its neighbors send their coalition information including payoff of each device in the coalition. On receiving the information, the hotspot determines if it should split its coalition or merge it with its neighbors, i.e., if it would get a smaller allocation of frequency by merging or splitting its coalition. If it decides to merge or split the coalition, it first determines the validity of the new coalitions, then computes the new data capturing frequency allocations and informs its neighbors affected by its decision. The client devices in turn propagate the information to their neighbors in the coalition when they act as hotspot or client again. The proposed algorithm is repeated periodically, enabling devices to autonomously adapt to changes in available energy and mobility, if any, of the devices.

VI. Performance Evaluation

To evaluate the proposed architecture, we first implemented it on Android platform and deployed it on a Google Nexus One device running Android OS 2.3.6. Using the device, we characterize the energy consumed by the application in the initialization phase and provide guidelines for selection of various algorithm parameters based on it. Based on the provided guidelines, we select the algorithm parameters in data forwarding phase and conduct large-scale simulations to evaluate it. Each data point represents the average value observed over 25 iterations of each experiment.

A. Prototype-based Evaluation

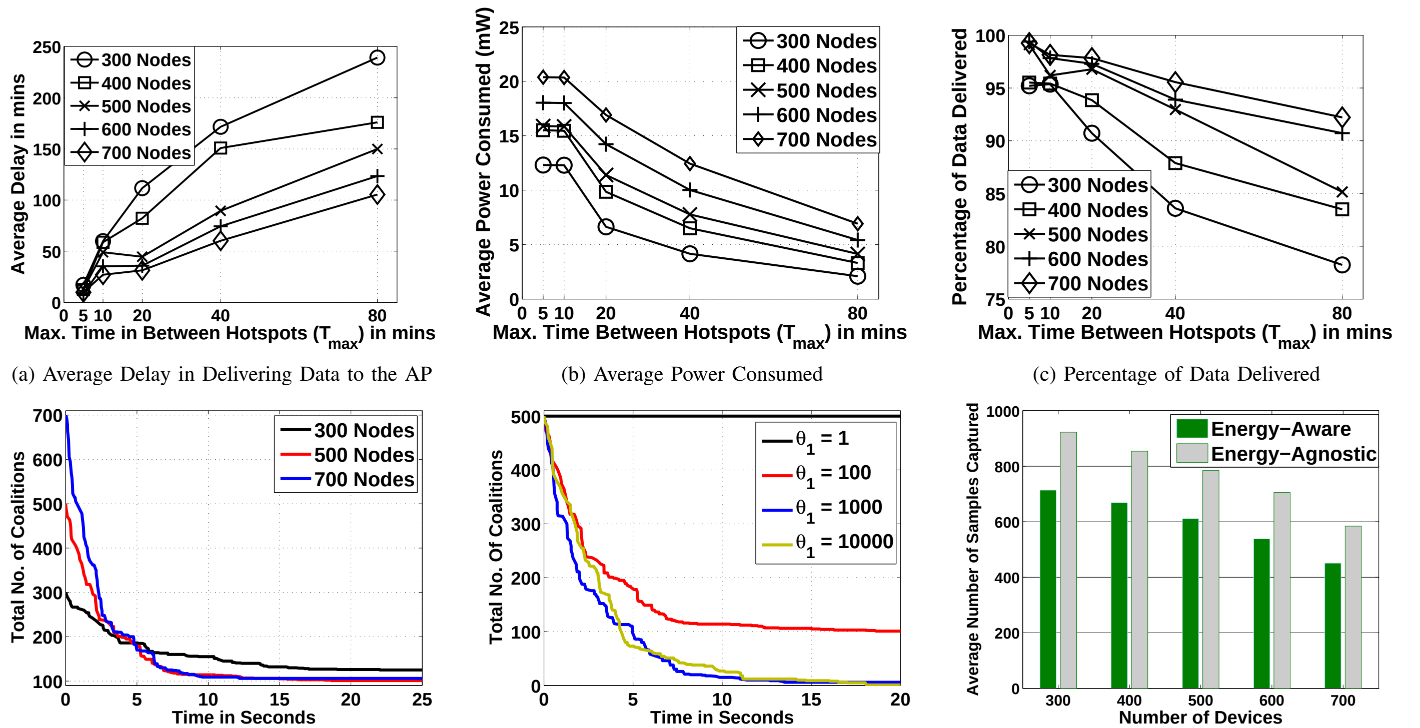
To measure the energy consumed by the E-DARWIN application, we used the PowerTutor energy profiler [18]. In the first set of experiments, we measured the energy and time consumed in transitioning between different states, as shown in Figure 4a. More energy and time is consumed in transitioning between dormant and WiFi Tethering state than WiFi client state as not only do we need to enable or disable the wireless interface but also store or retrieve the hotspot configuration and create or destroy the wireless network with the given configuration. Maximum energy and time is consumed transitioning from the WiFi client state to the WiFi Tethering state as the application needs to store the client state, disassociate with the hotspot, disable the wireless interface, retrieve the hotspot configuration and then create the hotspot with the given configuration. In the initialization phase, the frequency with which a device becomes hotspot depends on $T_{discover}$. A large value of $T_{discover}$ will ensure less energy is consumed due to transitions to or from the WiFi Tethering state. However, this implies that the devices will spend more time in client state. In client state, the devices scan the network for advertisements from hotspot. Since, the devices become hotspot for $I_{discover}$ time, the client devices must scan the network at least once to discover the hotspot, i.e $T_{scan} \leq I_{discover}$. To this end, we kept $T_{initial}$ = 40 mins and varied $I_{discover}$ and T_{scan}, to study their impact on the power consumed by the device in client state during initialization, as shown in Figure 4b. As the frequency with which the devices scan the network increases, the sleep cycle of the WiFi interface is interrupted more frequently, which results in less sleep and higher power consumption. Thus, keeping $T_{scan} = I_{discover}$ ensures that the client device scans the network at least once to receive advertisements from the hotspot and minimum power is consumed. However, the advertisements may be lost due to variations in wireless channel or interference. Hence, we keep $T_{scan} = \frac{I_{discover}}{3}$ to ensure that the client devices have more opportunities to discover a hotspot. Furthermore, the power consumed by the device decreases with increasing $I_{discover}$ as the scanning frequency comparatively decreases. Thus, keeping $I_{discover}$ high, results in lower rate of energy consumption for the client devices. But it implies that the devices stay longer in hotspot state. To this end, we kept $T_{initial}$ = 40 min and varied $I_{discover}$, to study its impact on the battery consumption of the device in hotspot state, as shown in Figure 4c. The power consumed by the application in hotspot state increases with increase in $I_{discover}$ as the wireless interface needs to be active for the complete duration. Thus, while higher $I_{discover}$ implies lesser rate of energy consumption for the devices in client state, it comes at a cost of higher rate of energy consumption for the devices in hotspot state. Based on these observations, we conducted additional experiments to study the impact of $T_{discover}$ and $T_{initial}$ on the energy consumption of the application by selecting $I_{discover}$ = 1 min, as shown in Figure 5. As $T_{initial}$ increases, the total activity time of the application increases, which results in more battery consumption. For a given $T_{initial}$, as $T_{discover}$ increases, the devices enter WiFi Tethering state less frequently and results in lower power consumption as discussed earlier.

B. Simulations

We implemented the proposed solution using MiXiM-INET framework in OMNET++ simulator. We assumed a deployment area of 1 sq km and varied the number of devices from $300 - 700$ in steps of 100 to represent varying populations densities. The devices were distributed uniformly at random in the deployment area based on the observed distribution of population count in real-world data sets [19]. The time a device stays in initialization phase was selected as $T_{initial} = T_{max} + 1$, with $T_{discover}$ = 6 mins, $I_{discover}$ = 1 min, I_H = 2 min, $T_{scan} = \frac{I_{discover}}{3}$ during initialization and $T_{scan} = \frac{I_H}{3}$ thereafter. We assumed that there is only one WiFi AP with satellite connectivity located at the center of the network and the ECC requires devices to capture sensory data at the rate of 1 sample/min. The battery capacity and maximum transmission power of each device was selected as $11.78Wh$ and 110.11mW respectively, which is representative of current smartphones. The available energy of each device was chosen uniformly at random between 0 and maximum capacity. The data observed from experiments in Section VI-A on the time and energy consumed in transitioning between various states was input into the battery model of the simulator to make the results as close to reality as possible. The wireless channel was modelled as two ray path loss model with the devices operating at carrier frequency of 2.4 GHz. In the first set of experiments, we studied the impact of T_{max} on the delay in delivering data to the WiFi AP, as shown in Figure 6a. As we increase T_{max}, the time interval between a device acting as hotspot increases, resulting in longer delays in delivering data to the WiFi AP as devices must wait for one of their neighbors to become hotspot to offload data. Furthermore, for a given T_{max}, as the density of the devices in network increases, the devices spend more time in client state and less in dormant state as they have more neighbors. This results in higher rate of energy consumption for the devices as shown in Figure 6b. However, the devices have more opportunities to offload data to the relay device, which reduces the delay in delivering data to the WiFi AP. When T_{max} = 5 mins and T_{max} = 10 mins, the devices enter the hotspot state very frequently as a result of which they are always in client or hotspot state but never in dormant state. Thus, the power consumption is highest in both the cases. As T_{max} increases, the devices enter hotspot and consecutively client state less frequently and stay dormant longer, resulting in lower power consumption, as shown in Figure 6b. However, with increasing T_{max} more data gets stored at each device, which results in higher volume of traffic being offloaded at each step. Thus, there is more contention among the devices in forwarding data to the relay device through the hotspot at each step. The contention results in higher data loss due to congestion and interference at the hotspot, as shown in Figure 6c. Hence, even though higher T_{max} results in lower rate of energy consumption for the devices, it comes at the cost of higher delay and lower data delivery. In the second set of experiments we evaluated the coalition formation game. Among various models used in literature to measure spatial correlation between the devices, the power exponential model is the most popular and is used here [16]. The power exponential model is given by $e^{\left(\frac{-d}{\theta_1}\right)^{\theta_2}}$, with $\theta_1 > 0, \theta_2 \in (0, 2]$, where d is the distance between the devices, θ_1 is the range parameter, which controls how fast the spatial correlation between the devices decays with the distance between them and θ_2 is the smoothness parameter, which controls the geometrical properties of the random field. The model is substituted in the distortion function given by [16] as covariance function to measure the distortion in samples of the devices. First, we establish our claim that the proposed coalition formation game converges to a stable solution in small finite steps. To this end, we keep T_{max} = 40 mins and vary the number of devices in the network as 300, 500 and 700 with $\theta_1 = 100$ to depict different population densities as well as θ_1 as 10, 100, 1000, 10000 with 500 nodes to represent varying degree of correlation between the devices. Figure 6d and Figure 6e show that the devices converge to a constant number of stable coalitions in the network. With increasing device density, the devices find more devices to collaborate with from their set of neighbors. Hence, the devices form coalitions faster and converge faster. Furthermore, increase in θ_1 implies increased spatial correlation between the devices. Thus, the devices form larger coalitions, which increases the convergence time as the devices need to wait for one of their neighboring devices to become a hotspot before they can propagate their coalition information. At θ_1 = 10, the spatial correlation between the devices is low, as a result of which no valid coalition involving two or more devices can be formed and thus, we see no coalition formation activity. To evaluate the benefits of the proposed energy aware coalition formation game, we assume $\theta_1 = 100$, $\theta_2 = 1$ and $D^{\star}_{x_i} = 2$ and compare it with an energy agnostic approach, where the coalitions are formed as per the algorithm discussed in Section V but without considering the available energy of the device in Equation 1. The energy consumed in capturing samples of data, depends on the type of data being captured and varies with it. Therefore, to perform a comparison independent of the cost of the capturing data samples, we compare the average number of samples captured per device using the two approaches, as shown in Figure 6f. In the proposed mechanism, the devices get allocated the frequency of capturing data based on their available energy and network participation. The process results in higher energy efficiency for the devices, which results in higher degree of cooperation among the devices than energy-agnostic approach.

VII. Conclusion

In this paper, we proposed a novel architecture called E-DARWIN for creating the network infrastructure in disaster affected areas using WiFi Tethering. Specifically, we discuss mechanisms using which the devices can autonomously discover their neighbors and forward data to the WiFi AP with minimum delay. Additionally, to utilize the rich multi-modal sensory capability of wireless devices, we designed a distributed coalition formation game that allows spatially correlated devices to share the data capturing tasks among them based on their available energy and network participation for higher network lifetime, which is crucial in disaster scenarios.



(a) Average Delay in Delivering Data to the AP

(b) Average Power Consumed

(c) Percentage of Data Delivered

(d) Convergence of the Coalition Formation Game with Varying Number of Devices and $\theta_1 = 100$

(e) Convergence of the Coalition Formation Game with Varying $\theta_1 = 10, 100, 1000, 10000$

(f) Comparison of Energy Aware and Agnostic Approach

Fig. 6: Performance Evaluation of Data Forwarding Phase

We analyzed the proposed solutions through an exhaustive set of experiments using a prototype implemented on Android platform and large-scale simulations. In the future, we plan to address many of the assumptions and limitations of the current scheme. In particular we plan to consider mobile devices with a much wider set of characteristics in terms of available sensors, networking capabilities and storage availability as well as the topology of the network to ensure devices crucial to connectivity do not exhaust their battery quickly. We will also design mechanisms for automated activation of E-DARWIN application and optimal deployment of WiFi APs. We will also address the security and trust issues that we have currently ignored. In particular, we would devise mechanisms to verify the veracity of the data supplied and build trust in the system.

VIII. Acknowledgement

This work is partially supported by NSF grants under award numbers CNS-1355505, IIS-1404673, CNS-1404677.

References

[1] L. Bengtsson *et al.*, "Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: A post-earthquake geospatial study in haiti," *PLoS medicine*, vol. 8, 2011.

[2] W. Shan *et al.*, "Collecting earthquake disaster area information using smart phone," in *IEEE Int'l Conf. on System Science and Engineering (ICSSE)*, 2012, pp. 310–314.

[3] P. W. Gething and A. J. Tatem, "Can mobile phone data improve emergency response to natural disasters?" *PLoS Medicine*, vol. 8, 2011.

[4] "B.a.t.m.a.n. - better approach to mobile ad-hoc networking." [Online]. Available: http://www.open-mesh.org/projects/open-mesh/wiki

[5] M. Gielen, "Ad hoc networking using wi-fi during natural disasters: Overview and improvements," in *17th Twente Student Conference on IT*, vol. 17, June 2012.

[6] S. Trifunovic, A. Picu, T. Hossmann, and K. A. Hummel, "Slicing the battery pie: Fair and efficient energy usage in device-to-device communication via role switching," in *Proc. of the 8th ACM MobiCom Workshop on Challenged Networks*, 2013, pp. 31–36.

[7] H. Han *et al.*, "Dozyap: Power-efficient wi-fi tethering," in *Proc. of the MobiSys*, 2012, pp. 421–434.

[8] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "Wifi-opp: Ad-hoc-less opportunistic networking," in *Proc. of the 6th ACM Workshop on Challenged Networks*, 2011, pp. 37–42.

[9] F. Busching, S. Schildt, and L. Wolf, "Droidcluster: Towards smartphone cluster computing – the streets are paved with potential computer clusters," in *Proc. of Int'l Conf. on Distributed Computing Systems Workshops*, 2012, pp. 114–117.

[10] H. Wirtz, T. Heer, R. Backhaus, and K. Wehrle, "Establishing mobile ad-hoc networks in 802.11 infrastructure mode," in *Proc. of the 6th ACM Workshop on Challenged Networks*, 2011, pp. 49–52.

[11] H. Kazerooni, "The berkeley lower extremity exoskeleton project," in *Experimental Robotics IX*, ser. Springer Tracts in Advanced Robotics, 2006, vol. 21, pp. 291–301.

[12] "Surfbeam 2 broadband system," ViaSat Inc. [Online]. Available: http://www.viasat.com/files/assets/surfbeam2_Overview_018_web.pdf

[13] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537 – 568, 2009.

[14] C. Yang, J. Yang, X. Luo, and P. Gong, "Use of mobile phones in an emergency reporting system for infectious disease surveillance after the Sichuan earthquake in China," *Bulletin of the World Health Organization*, vol. 87, pp. 619 – 623, 2009.

[15] S. M. George *et al.*, "Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 128–136, 2010.

[16] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Comput. Netw.*, vol. 45, no. 3, pp. 245–259, June 2004.

[17] K. R. Apt and A. Witzel, "A generic approach to coalition formation," *Int'l Game Theory Review*, vol. 11, no. 03, pp. 347–367, 2009.

[18] "Powertutor." [Online]. Available: http://powertutor.org/index.html

[19] M. Salvatore et al., "Mapping global urban and rural population distributions," *Food and Agriculture Organization of the United Nations*, 2005.