Demo Abstract: DOME – IoT-Based Monitoring Emergent Events for Wildland Fire Resilience

Fangqi Liu, Tung-Chun Chang, Kevin Liu, Cyrus Li, Sachin Krishan Thyaharajan, Sideshwar Jappeswaran Balasubramanian, Nalini Venkatasubramanian
University of California, Irvine
{fangqil2,tungchuc,kevil21,yuanpeil,sthyahar,sjappesw,nalini}@uci.edu

ABSTRACT

We present DOME, an IoT monitoring system that employs mobile drones and in-situ IoT devices to gather real-time data for situational awareness during emergent and evolving events, with a focus on wildland fires. DOME integrates, and processes all collected sensing data and presents a dashboard that displays the dynamic status of various features, including fire, weather, and air quality. Based on the perceived fire status and wind conditions, DOME leverages physics-based fire models to predict the future evolution of the fire. Moreover, DOME integrates algorithms that plan the flight of multiple drones and control their motions to support automatic drone-based mobile sensing. This feature enables efficient data collection and enhances the system's overall monitoring capabilities.

KEYWORDS

Internet of Things, Sensors, Drone, Mobile sensing, Wildland fires

ACM Reference Format:
Fangqi Liu, Tung-Chun Chang, Kevin Liu, Cyrus Li, Sachin Krishan Thyaharajan, Sideshwar Jappeswaran Balasubramanian, Nalini Venkatasubramanian. 2023. Demo Abstract: DOME – IoT-Based Monitoring Emergent Events for Wildland Fire Resilience. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3576842.3589180

1 INTRODUCTION

This paper explores techniques for monitoring emergent events in community settings, such as large-scale sporting events, military activities, and natural disasters like earthquakes, fires, and floods. By tracking these evolving scenarios over time, the goal is to promote human safety, minimize property damage, and mitigate ecosystem impacts in mission-critical situations. As a driving use case for emergent events, this work focuses on wildland fires, an ongoing threat that has caused significant damage to natural ecosystems, property, infrastructure, and human safety. From 2012 to 2021, the US has experienced an average of 61,289 wildfires, impacting approximately 7.4 million acres annually [5]. However, gaining situational awareness is difficult in wildland communities with limited infrastructure. Existing technologies for monitoring wildland fires include remote sensing/satellite imagery and in-situ wireless sensor networks, but they have practical limitations, such as delays, coarse data resolution, and maintenance difficulties due to fire damage. Thanks to advances in sensing, mobility, and computing, low-cost aerial sensing technologies such as drones are now feasible and suitable for gathering data on wildland fires. This work presents a prototype IoT system called DOME [4], which utilizes drones and in-situ sensors to enable continuous monitoring of wildland fires. DOME offers an interface to integrate multiple types of sensing data and supports automatic planning and control of drone-based mobile sensing. Furthermore, it provides a graphical user interface for users to visualize the dynamic status of various features during wildland fires.

2 SYSTEM OVERVIEW

Our DOME system, depicted in Figure 1, utilizes drones equipped with RGB cameras and various in-situ sensors to collect real-time data for monitoring critical factors during wildland fires, such as fire status, air quality, and wind conditions. All collected data is transmitted via WiFi networks to an edge server, which is responsible for controlling the drone-based sensing process and managing data integration, analysis, and storage. The DOME code and detailed installation and implementation instructions are publicly available in [1]. In this section, we will describe DOME's four primary functions.

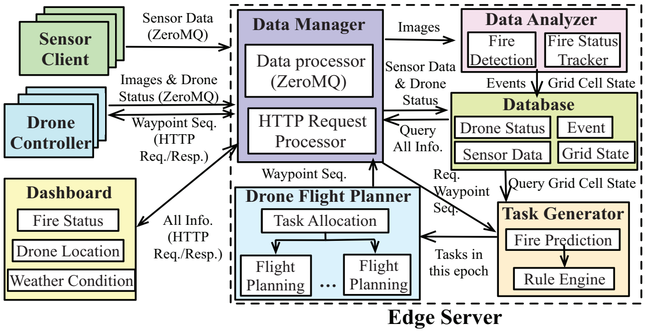


Figure 1: System Architecture of DOME

In-situ sensor data collection. In DOME, sensors are located near the fire setting and periodically collect and transmit sensing data to the edge server. All sensors are attached to Raspberry Pis (RPi) running Scale-Client [2], a modular software stack that collects raw data from sensors via I^2C and extracts sensing data by parsing them. The extracted data are then transmitted to the edge server for further processing. We leverage the Zero Message Queue (ZeroMQ), an open-source publish and subscribe communication

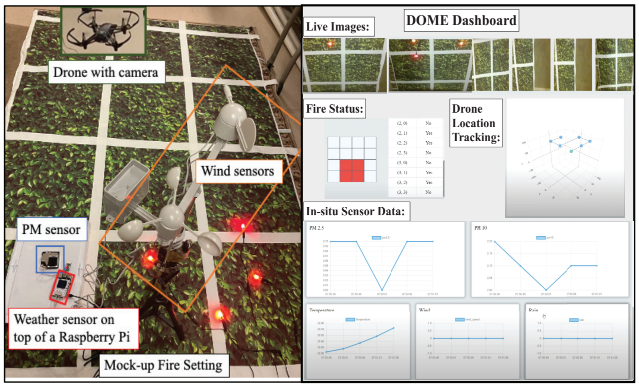
IoTDI '23, May 09–12, 2023, San Antonio, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0037-8/23/05.
https://doi.org/10.1145/3576842.3589180

model, to send/receive various types of sensing data in parallel. In each RPi, we create a publisher for each sensor and let it publish data with a specific topic corresponding to its sensing type and ID following a pre-defined period. The edge server has a **data manager** component that subscribes to all sensing data and stores them in a MongoDB-based database.

Drone-based mobile sensing. We utilize a predefined waypoint sequence to guide each drone's flight path and capture a top-down view image at each waypoint. The images are stamped by time and location and transmitted to the edge server through a WiFi network. This process is overseen by a **drone controller** that sends commands to the drone and relays the captured images using ZeroMQ. The data manager receives the images and passes them to a **data analyzer**, which employs image processing techniques such as FireNet [3] to detect and locate fires. The fire setting is divided into grid cells to represent the fire status, such as 'Fire' or 'No fire.' Each fire detection result combined with fire location (grid cell) and detection time would be cast as an event to update the state of grid cells. Rules for state transitions are described in [4].

Multi-drone flight planning. The DOME system schedules drone flights periodically for a predefined time period called an epoch. At the start of each epoch, each drone controller sends a waypoint query HTTP request to the data manager, which handles the request using Flask. The data manager will then trigger the **task generator** to create tasks that specify the spatial and temporal drone data collection requirements based on the fire status, predicted fire evolution, and user-defined task generation rules. The **drone flight planner** then assigns these tasks to specific drones and generates a waypoint sequence for each drone to complete its tasks. The generated waypoint sequences are sent to the data manager, which forwards them to the corresponding drone controller. Please refer to [4] for task generation and flight planning details.

Visualizing fire setting status. DOME provides a comprehensive **dashboard** that enables users to visualize critical aspects of wildland fires, including current fire locations, predicted fire spread, drone positions, wind conditions, and air quality. We developed an HTML-based website that retrieves this information from the edge server through HTTP requests and updates the data periodically. The data manager responds to these requests by querying data from the database and sending it to the dashboard for display.



(2, 0)	No
(2, 1)	Yes
(2, 2)	Yes
(2, 3)	No
(3, 0)	No
(3, 1)	Yes
(3, 2)	Yes
(3, 3)	No

Figure 2: Sensors, Testbed and Dashboard in DOME

3 DEMO OVERVIEW

Our demo will showcase three key operations in the DOME system that facilitate continuous monitoring of wildland fires, which will be demonstrated in a mock-up fire setting as follows.

In-situ sensor data collection: As illustrated in Figure 2, our demo scenario utilizes three types of sensors for monitoring: wind (speed and direction), weather (temperature and relative humidity), and PM (particulate matter for smoke monitoring). Each sensor is connected to a Raspberry Pi (RPi) through various hardware communication interfaces. The weather sensor is connected to the RPi via GPIO, the wind sensor is connected through RJ11 cables, and the PM sensor communicates with the RPi through a USB interface. All in-situ sensors capture sensing data every 30 seconds, which is transmitted and stored in our edge server (a laptop) through a local WiFi network. To illustrate the continuous data collection and transmission process, we will display incoming data in the database and figures in the dashboard, updated every 10 seconds.

Drone-assisted fire status monitoring: We will present how the DOME system leverages drone-based sensing to track fire status updates in a small-scale mock-up fire setting measuring 6×8 ft and partitioned into a 3×4 grid, as shown in Figure 2. To simulate wildland fires, we utilize multiple red LED lights. For data collection and flight control, we employ a DJI Tello drone equipped with a top-down RGB camera and the Tello SDK interface. A laptop drone controller will send commands to control the drone to fly above the fire setting following a waypoint sequence and capture an image at each waypoint. During this demo, we will showcase the received images and fire detection results in our dashboard. The dashboard will display recently received photos with detected bounded fires. Using the drone location and data capture time, we will create fire detection events that provide the location of the fires with granular detail down to the grid cells. The detected events will trigger updates of the fire status in the dashboard.

Drone flight planning process: We will demo our periodic drone flight planning process, which occurs every 3 minutes. To provide a clear understanding of the workflow of the flight planning procedure at each planning time, we will first display the predicted fire evolution (estimated fire arrival time of all grid cells) based on the current perceived fire status and the wind condition. Then, we will illustrate all generated tasks for the drone given some user-specified rules. Finally, we will demonstrate the developed waypoint sequence for the drone during this epoch. The dashboard will continuously track the drone's flight and update its location, as shown in Figure 2. Furthermore, we will demonstrate the adaptability of our flight planning procedure to diverse monitoring requirements, fire statuses, and weather conditions. This will be achieved by modifying the task generation rules, turning on/off mockup fires, and adjusting the wind condition.

ACKNOWLEDGMENTS

This work is supported by the University of California Office of the President (No. LFR-20-653572), the National Institute of Standards and Technology (No. 70NANB17H285), NSF (No. 2008993) and DARPA (No. FA8750-16-2-0021).

REFERENCES

[1] 2023. DOME. https://github.com/Fangqierin/DOME-Monitoring-System
[2] Kyle Benson et al. 2015. Scale: Safe community awareness and alerting leveraging the internet of things. *IEEE Communications Magazine* 53, 12 (2015).

[3] Arpit Jadon et al. 2019. FireNet: a specialized lightweight fire & smoke detection model for real-time IoT applications. *arXiv preprint arXiv:1905.11922* (2019).

[4] F. Liu, J. Baijnath-Rodino, et al. 2023. DOME: Drone-assisted Monitoring of Emergent Events For Wildland Fire Resilience. In *ACM/IEEE ICCPS'23*. ACM.

[5] Congressional Research Service. 2023. Wildfire Statistics. Retrieved March 16, 2023 from https://sgp.fas.org/crs/misc/IF10244.pdf

SPECIAL REQUIREMENT

In order to demonstrate our DOME system effectively, we will require a space of approximately 7 by 8 feet for our fire setting, a table and three power strips to power all sensors and mock-up fires.