Support Emergency Response in Automatic Identification System using an Opportunistic Resource Utilization Networks

Andrew Griesdorn, Quinn Hirt, Raed Salih, Michael R. Clark, Scott Brookes
Riverside Research, 2640 Hibiscus Way, Beavercreek, OH, USA 45431

ABSTRACT

The automatic identification system, or AIS, transmits a vessel's position, identity, and speed information so that other vessels are aware of the transmitting vessel and can avoid collisions. However, AIS has multiple limitations, including the quality, availability, and validity of data for non-cooperative vessels. We investigate methods of supporting the availability of AIS operation through an emergency response of cooperative vessels, thereby creating an opportunistic resource utilization network, or Oppnet. An Oppnet invites and integrates heterogeneous devices and systems available in the given environment, wherein the integrated devices/systems become the Oppnet's helpers. Helpers are resources that deliver support, communication, computation, and control to find missing cooperative vessels. We design and build this Oppnet scenario using the Common Open Research Emulator (CORE), an open-source network emulation tool funded by a U.S. Naval Research Laboratory research project. CORE provides a graphic user interface (GUI) tool for creating and tracking virtual network status and performance as well as a Python API to programmatically access, control, and monitor instances.

During the simulation, the Oppnet starts by inviting and integrating heterogeneous nodes (devices or systems) available within its reach, becoming helpers. The Oppnet then assigns tasks to all helpers to detect the missing cooperative vessel, "Nemo". In each iteration, the simulation runs for a specified time until Oppnet's helpers find "Nemo". Otherwise, "Nemo" is lost. Initial results of simulation show that the finding "Nemo" rate increase when Oppnet helpers increase; however, the timespan of finding "Nemo" remains nearly identical due to simulation conditions and parameters.

Keywords: Automatic Identification System (AIS), Mobile Ad Hoc NETworks (MANETs), Opportunistic Resource Utilization Networks (Oppnet)

1. INTRODUCTION

1.1 Background

The automatic identification system, or AIS, transmits a given ship's position so that other ships are aware of its position. More than 400,000 AIS devices broadcast vessel location, identity, course, and speed information every year [3]. Ground stations and satellites pick up this information, making vessels trackable even in the most remote areas of the ocean in order to broadcast their position with AIS and avoid collisions. However, AIS has its limitations, including the quality of data, the availability of data, and the validity of data for non-cooperative targets.

Traditional resource virtualization techniques have been developed mainly for relatively stable network infrastructures. As a result, they cannot adequately function over highly variable and dynamic Mobile Ad hoc NETworks (MANETs) of Unmanned Aerial Vehicles (UAVs), which play prominent roles in various fields—including defense, natural disaster recovery, real-time surveillance, and reconnaissance [1].

Oppnets behave like an emergency response team arriving at a disaster scene, growing opportunistically by taking control of resources provided by local helpers. An Oppnet can invite and integrate heterogeneous devices and systems available from the environment; the integrated devices/systems become the Oppnet's helpers [2] [1]. Helpers are resources that can deliver support, communication, and computation.

CORE, or the Common Open Research Emulator, is a network-based emulation tool developed for the Navy. It allows the creation of different computing machine and network connection types in both a graphical interface and Python API. These nodes are then given a Linux Namespace, making them able to interact with each other as if they were real computers on a real network, including the abilities for them to gain IP addresses and communicate over sockets.

1.2 Related Studies

Emmems et al. [3] present an empirical study of AIS data and discuss the promises and perils of AIS data by integrating the outcomes of most current research in this area. The authors also conduct a quantitative analysis of AIS data at the Amsterdam port. In response to the growing amount of data collected from a multitude of vessels, the authors discuss the limitations of AIS data, including the data quality, data redundancy, data complexity, and poor performance in high traffic areas. The study outcomes, however, show that AIS data provides additional information for navigation by radar, consequentially enhancing safety. Additionally, AIS data can improve route identification, efficiency monitoring, and environmental efforts.

Wang and Lilien [1] propose an application of Oppnets for UAV ad hoc networking by simulating a homeland defense use case for Oppnets that detects suspicious watercraft. The proposed simulation compares the performance of an Oppnet versus a baseline case in which no Oppnet is used. The goal is to determine and compare the time required to identify the speedboat location both with and without the help of Oppnet and its available helpers. In the Oppnet operation, resource or service discovery is limited to nodes only within the network-serving infrastructure, contrasting with a traditional network in which all nodes are deployed together. Supported by simulation results, the study condlues that Oppnet is a promising framework for high-performance ad hoc UAV networking. Furthermore, the authors claim that Oppnets achieve high performance via less invasive use of helpers combined with the denial of help by some of the candidate helpers.

Lilien et al. [2] use an Oppnet to simulate first responders finding rescue routes for evacuating victims following a static chemical spill on a single floor of a building. As a first responder, an Oppnet can assist in finding optimal rescue routes for evacuating victims as well as add more resources to cope with an emergency. The authors assume that the number of victims is known. The simulation tasks include both finding the location of all victims who cannot self-evacuate and that of all victims whom rescuers evacuate. In the simulation, the building floor is partially or fully covered by resources, including sensors or cameras. The victims carry these resources, such as cellphones and intelligent badges. Some of these resources are additionally located in individual rooms, including resources such as surveillance cameras, cellphones lying on desks or desktops, and laptops. The simulation results show how the number of victims rescued by the responders and the time required for their evacuation depend on (a) the number of an Oppnet's helpers that the Oppnet can integrate; (b) the number and distribution of victims within the incident area; and (c) the number of responders. The results demonstrate that the use of Oppnet can significantly improve the efficiency of emergency response operations, given that the evacuation duration depends on the communication of helpers via the Oppnet.

2. METHODOLOGY

2.1 Simulation Setup

Utilizing CORE, the Oppnet is simulated with a task of aiding in ships that lose Automatic Identification System (AIS) capabilities. In this scenario, a target ship “Nemo” drifts and loses AIS communications with its surrounding ships. The Oppnet is alerted of “Nemo” being unresponsive and is deployed to locate the missing ship, thereby finding “Nemo”.

We make the following assumptions for the simulation:

- “Nemo” experiences some critical failure and is unreachable from its original network.
- The Oppnet already has known nodes it can contact.
- The Oppnet is somehow able to communicate with “Nemo”, once found.
- The Oppnet has perfect communication within its own network, with infinite range.
- The Oppnet nodes will be potentially busy or unable to help the network as a whole.

The scenario is programmatically generated with Python code that interfaces with CORE itself as well as gRPC client/server API that interfaces with CORE using Python GUI. A small network that “Nemo” initially belongs to is created and is the same across all executions of the simulation. It is assumed that “Nemo” will lose all connections to this network, so the actual creation of this network is unimportant. From there, the Oppnet nodes are generated with some specified number of nodes in random locations.

2.2 Simulation Execution

Once the scenario is generated, it is executed. “Nemo” will have a random direction generated for it and will drift in said direction. Once it is out of range of its AIS network, the Oppnet will deploy. For every node in the Oppnet, there is some specified probability (P = 0.5 for this paper) that the node will become a helper and thus will be utilized in finding “Nemo”. This probability reflects how an individual node may be utilized for another task at the time or for some other condition that makes it unable to contribute resources. At this point, all nodes that have become helpers will spread out in pseudo-random directions.

The simulation then runs until one of two conditions are met:

- Some helper node in the Oppnet comes within some specified detection radius and signals to the rest of the Oppnet that “Nemo” has been found.
- A set amount of time passes without finding “Nemo”, signaling that the Oppnet was unable to intercept “Nemo”.

3. DATA

From the simulation, the following metrics are recorded per execution:

- The number of nodes that can potentially become helpers
- The probability of nodes becoming helpers
- The total number of nodes that became helper.
- The time taken for “Nemo” to be found
- Whether or not “Nemo” was found within the time constraint. *

*Note: In the case that “Nemo” were not found within some time frame, we record time-to-be-found as 0, an impossible value.

To run the simulation multiple times, a script is written that will generate system calls to the Oppnet. The number of potential nodes will range between 5 and 20 in increments of 5. Each increment will then be run 1000 times to find averages and standard deviations for the above-mentioned metrics of interest. The upper time limit is a static value that is empirically determined by noting the time it takes the failed attempts of finding “Nemo” to reach a state in which it is clear that “Nemo’s” trajectory will not intersect with an Oppnet helper node.

4. RESULTS

The data supports that as the number of helper nodes increases, as does the chance to find “Nemo” in a roughly linear manner, shown in Figure 1. Due to the probability of 0.5 being the chance that a node becomes a helper rather than does not, we have relatively few data points in which there were more than 15 helpers that join the Oppnet. Because of this, there are fewer data points to analyze and consequentially more noise in the results for those with higher helper counts. See Figure 2 for the frequency of helper counts.

The analysis of detection time per helper nodes, as shown in Figure 3, suggests that regardless of the number of helpers, “Nemo” is found in a consistently similar amount of time. Additionally, the minimum amount of time taken also remains nearly identical. Due to the way the simulation is generated, the starting positions of all the potential helper nodes are fixed to a certain radius around a fixed point for the Oppnet. This constraint helps ensure that the performance of a process instance is based on the number of nodes, not the position at which Oppnet starts. An initial start closer or farther away could skew results over many trials. This constraint, however, likely makes the time to detection tightly bounded regardless of the number of nodes. Only the likelihood of detection increases.

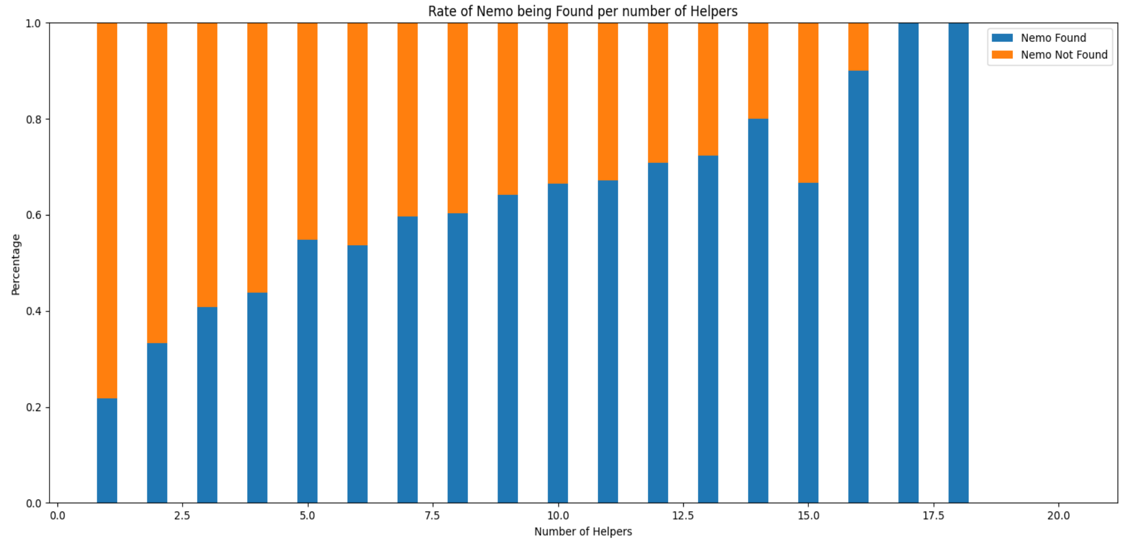

Figure 1. Detection rate of “Nemo”. Blue is successful, Orange is failure.

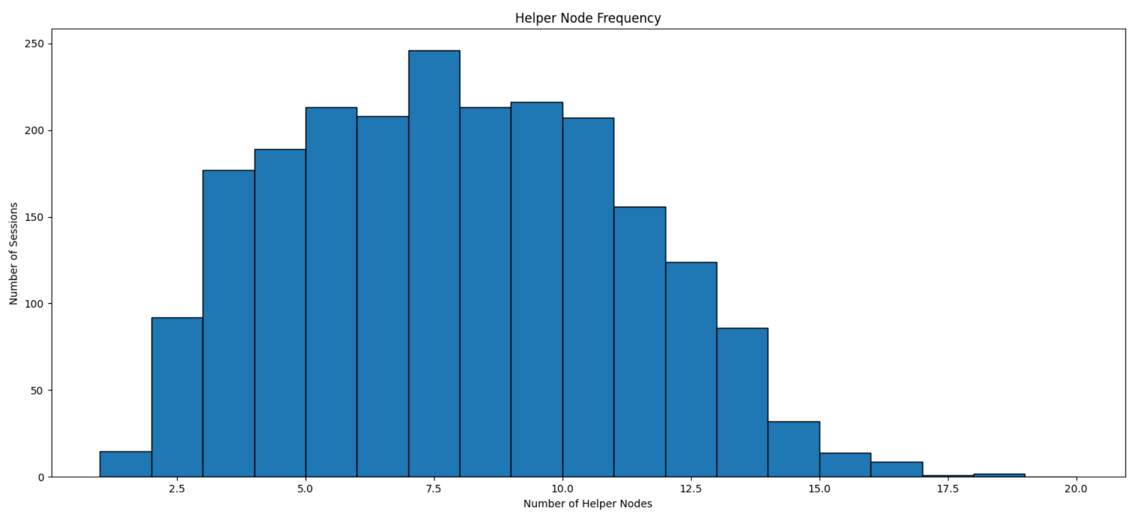

Figure 2. Distribution of helper nodes that joined the Oppnet across simulation runs.

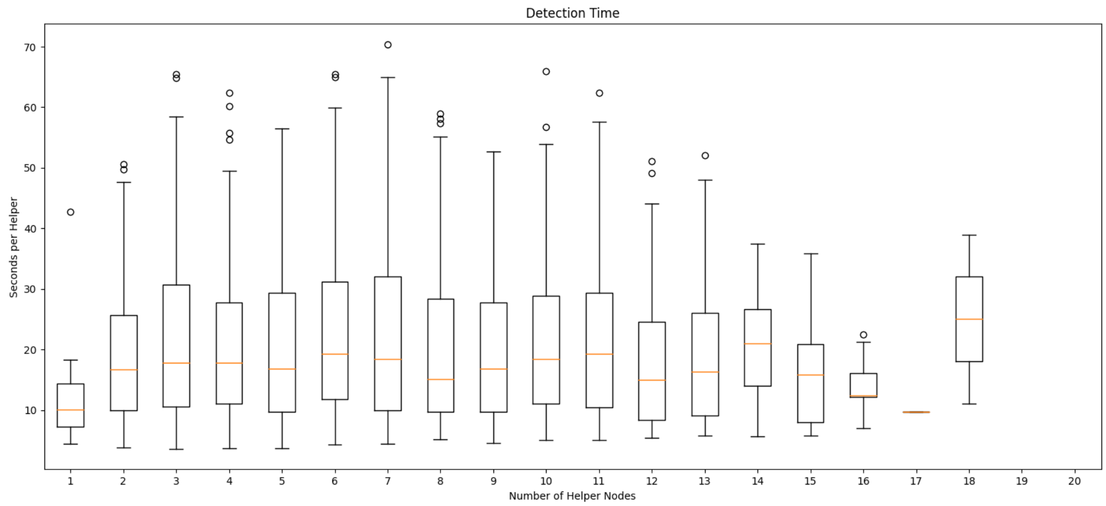

Figure 3. Box plots of detection time per number of helper nodes that join the Oppnet.

5. CONCLUSION

In this paper we highlight the use of CORE to simulate an Oppnet that is able to find a missing ship as it loses AIS capabilities. We believe that Oppnets are a good solution for emergency response and defense solutions. Simulations of these networks are vital to understanding the latent behaviors that emerge, such as our results with non-variable time to detect regardless of increasing number of helpers. In future works, exploring the use of heterogenous node capabilities that could further expand the simulation potential for Oppnets.

REFERENCES

[1] L. Lilien, L. b. Othmane, P. Angin, A. DeCarlo, R. Salih and B. Bhargava, "A simulation study of ad hoc networking of UAVs with opportunistic resource utilization," *Journal of Network and Computer Applications,* pp. 3-15, 2014.

[2] L. Lilien, M. Elbes, L. b. Othmane and R. Salih, "Simulation of emergency response operations for a static chemical spill within a building using an opportunistic resource utilization network," in *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, Waltham, MA, 2013.

[3] T. Emmens, C. Amrit, A. Abdi and M. Ghosh, "The promises and perils of Automatic Identification System data," *Expert Systems with Applications*, vol. 178, 2021.

[5] DARPA, "Creating Cross-Domain Kill Webs in Real Time," 18 September 2020. https://www.darpa.mil/nes-events/2020-09-18a.