



Contents lists available at SciVerse ScienceDirect

## Journal of Network and Computer Applications

journal homepage: [www.elsevier.com/locate/jnca](http://www.elsevier.com/locate/jnca)

## A simulation study of ad hoc networking of UAVs with opportunistic resource utilization networks

Leszek T. Lilien<sup>a</sup>, Lotfi ben Othmane<sup>b</sup>, Pelin Angin<sup>c,\*</sup>, Andrew DeCarlo<sup>d</sup>, Raed M. Salih<sup>a</sup>, Bharat Bhargava<sup>c</sup>

<sup>a</sup> Department of Computer Science, Western Michigan University, 1903 West Michigan Avenue, Kalamazoo, MI 49008, USA

<sup>b</sup> Department of Mathematics and Computer Science, Eindhoven University of Technology, Den Dolech 2, 5612 AZ, Eindhoven, The Netherlands

<sup>c</sup> Department of Computer Science, Purdue University, 305 N. University Street, West Lafayette, IN 47907, USA

<sup>d</sup> Infoscitex Corporation, 303 Bear Hill Road, Waltham, MA 02451, USA

### ARTICLE INFO

#### Article history:

Received 3 November 2012

Received in revised form

4 March 2013

Accepted 8 May 2013

#### Keywords:

Ad hoc networks

MANETs

Opportunistic networks

Opportunistic resource utilization networks

Simulation

UAV

### ABSTRACT

Specialized ad hoc networks of unmanned aerial vehicles (UAVs) have been playing increasingly important roles in applications for homeland defense and security. Common resource virtualization techniques are mainly designed for stable networks; they fall short in providing optimal performance in more dynamic networks—such as mobile ad hoc networks (MANETs)—due to their highly dynamic and unstable nature. We propose application of Opportunistic Resource Utilization Networks (Oppnets), a novel type of MANETs, for UAV ad hoc networking. Oppnets provide middleware to facilitate building flexible and adaptive distributed systems that provide all kinds of resources or services to the requesting application via a helper mechanism. We simulated a homeland defense use case for Oppnets that involves detecting a suspicious watercraft. Our simulation compares performance of an Oppnet with a baseline case in which no Oppnet is used. The simulation results show that Oppnets are a promising framework for high-performance ad hoc UAV networking. They provide excellent performance even under imperfect (and realistic) conditions, such as a less invasive use of helpers, denial of help by some of the candidate helpers, and imperfect detection capabilities of Oppnet components.

© 2013 Elsevier Ltd. All rights reserved.

### 1. Introduction

Mobile Ad hoc NETWORKS (MANETs) of Unmanned Aerial Vehicles (UAVs) are specialized ad hoc networks with increasingly widespread use. Their popularity is due to the cost-effective wireless communication and surveillance capabilities they provide. UAVs have had increasingly prominent roles in a variety of fields—including homeland defense and security, natural disaster recovery, real-time surveillance and reconnaissance among others. All these applications involve demanding applications in terms of rapid response to events.

Today's homeland defense operations rely heavily on MANETs of UAVs, as these offer actions free from possible human losses and have the advantage of being autonomous. UAVs have been great assets for homeland defense in the past few decades. Recently, the United States Congress has expressed a great interest in using UAVs for homeland security as well. In particular, they can improve

surveillance coverage along remote sections of the U.S. borders (Haddal and Gertler, 2010; Homeland Security Unmanned Aerial Vehicles (UAVs), 2013). UAVs are becoming indispensable tools for many other homeland security missions including emergency preparedness and response, intelligence activities, infrastructure and perimeter surveillance, detection of dangerous materials, etc. (cf. Homeland security, 2013).

Effective resource virtualization is the key to the success of MANETs, as it facilitates communication between network nodes and speeds up operations in the whole network involved in the target mission. Traditional resource virtualization techniques have been developed mainly for fixed and stable network infrastructures. These techniques cannot adequately function over MANETs of UAVs, because MANETs are highly variable, highly dynamic, unstable, lacking in infrastructure and pose additional challenges for resource virtualization. These challenges include frequent link breakage, inconsistencies in data rates, incompatibility of resources, and temporary unavailability of needed resources and communication links.

In this paper we propose *Opportunistic Resource Utilization Networks*—in short, *Oppnets*—for UAV ad hoc networking. Oppnet is a paradigm for virtualizing resources and acquiring necessary resources (e.g., sensors, computers, lightweight clients, and other

\* Corresponding author. Tel.: +1 765 430 2140.

E-mail addresses: [leszek.lilien@wmich.edu](mailto:leszek.lilien@wmich.edu) (L.T. Lilien),

[lben.othmane@tue.nl](mailto:lben.othmane@tue.nl) (L. ben Othmane), [pangin@cs.purdue.edu](mailto:pangin@cs.purdue.edu) (P. Angin),

[adecarlo@infoscitex.com](mailto:adecarlo@infoscitex.com) (A. DeCarlo), [raedmahdi.salih@wmich.edu](mailto:raedmahdi.salih@wmich.edu) (R.M. Salih),

[bb@cs.purdue.edu](mailto:bb@cs.purdue.edu) (B. Bhargava).

networks) and capabilities via a network (Lilien et al., 2006, 2010). The Oppnet technology provides middleware to facilitate building flexible and adaptive distributed systems that provide all kinds of resources or services to the requesting application via a helper mechanism.

In order to test the effectiveness of Oppnets in ad hoc networking, we developed a use case scenario and simulated the developed scenario for the cases where Oppnets are *not* used for achieving the target mission vs. cases where Oppnets are used for the mission. Results of the simulation experiments show that the use of Oppnet capabilities results in a significantly higher mission success rate and takes significantly shorter time to complete, which is promising for the use of the proposed approach in real-life operations.

The rest of this paper is organized as follows: Section 2 provides a brief overview of related work; Section 3 describes the general characteristics and operation of Oppnets; Section 4 describes our use case scenario involving UAVs and Oppnets; Section 5 provides the details of the simulation setup; Section 6 provides the results of the simulation experiments; and Section 7 concludes the paper, and provides future work directions.

## 2. Related work

MANET-based communication systems have been studied by many researchers for various emergency response applications. Jang et al. (2009) proposed a rescue information system for earthquakes based on MANETs of notebook PCs. Fujiwara and Watanabe (2005) proposed a hybrid wireless network, combining ad hoc networks and a cellular network, for maintaining connectivity between a base station and nodes in a disaster. Berio et al. (2007) introduced Wireless Infrastructure over Satellite for Emergency Communication (WISECOM), which aims to develop a complete telecommunication solution that can be rapidly deployed after a disaster, replacing the traditional use of satellite phones or heavy devices. Oh et al. (2010) proposed content-centric networking as a communication architecture providing retrieval of content by name in emergency MANETs.

MANETs specifically involving UAVs have also been studied mainly in the context of military operations. Reidt and Wolthusen (2008) proposed an authentication and negotiation protocol for MANETs to request services from UAVs in order to optimize the use of available network resources. They also investigated the effects of moving patterns of UAVs on network connectivity. Han et al. (2009) proposed an optimization model for the locations and movements of UAVs to improve MANET connectivity.

Most of the previous approaches for emergency response assumed a homogeneous network of resources, which limits the functionality and the performance of the systems proposed. The approach for UAV ad hoc networking proposed in this paper differs from previous work; it provides a framework capable of integrating heterogeneous resources into the MANET, thereby, maximizing performance with lowered response time.

## 3. Oppnets—Opportunistic Resource Utilization Networks

This section provides an overview of Opportunistic Resource Utilization Networks. Other publications provide implementation details of Oppnets (Lilien et al., 2007, 2010) and discuss their security and privacy challenges (Lilien et al., 2006).

The basic working principle of Oppnets is as follows (Lilien et al., 2010). To gain needed resources or services, an initially deployed Oppnet—known as the *Seed Oppnet*—starts discovering foreign nodes or application networks that are in the vicinity of the Oppnet. These foreign nodes and networks are not originally

components of the Oppnet that discovers them. The Oppnet invites or orders some of them to join its efforts as *helpers*. The Oppnet grows into an *Extended Oppnet* by taking control over helpers that join it (without compromising their privacy), and incorporates their resources/capabilities.

This ad hoc mode of Oppnet operation contrasts with traditional networks (including application networks, or distributed systems) where all nodes are deployed together, and resource or service discovery is limited to nodes only within the infrastructure serving the networks. In addition, Oppnets—being Specialized Ad Hoc Networks and Systems (SAHNS) (Lilien, 2007) for well-matching applications—might provide more application support than other ad hoc networks.

Behavior of Oppnets is analogous to the operation of an emergency response team arriving at a scene of a natural disaster. As the emergency response team grows opportunistically by taking control over local resources provided by numerous helpers, helpers in the Oppnet significantly expand communication, computing, storage, sensing, actuating, and other capabilities of the Extended Oppnet. This gives Oppnets an unparalleled potential for leveraging resources recruited from their environments. An avalanche of capabilities can thus be acquired at a minimal cost or even for free when nodes join an Oppnet.

The set of potential helpers for Oppnets is very broad, including communication, computing and sensor systems, wired and wireless, free-standing and embedded. Before a Seed Oppnet can grow, it must discover its own set of potential helpers available to it. This is not limited to a lookup of previously prepared information (e.g., a directory), which is often referred to as “discovery”, but also includes the more challenging true discovery. True discovery could involve an Oppnet node scanning the spectrum for signals or beacons, and collecting enough information to contact their senders.

Ordering candidate helpers to join an Oppnet may be controversial. To ensure expansion, there is a category of systems always available on an order of an emergency or other critical-application Oppnet. Each such system is called an *Oppnet reservist* (Lilien et al., 2010). (To complicate things, some reservists could be listed in the directory, while others have to be discovered; the latter will reveal their reservist status only upon being contacted by an appropriate Oppnet category.)

Oppnets as discussed in this paper are different from the opportunistic networks proposed by other researchers including Pelusi et al. (2006) and Sistla et al. (2005), which basically provide specialized facilities for opportunistic data forwarding and dissemination. They are also different from Oppnets that have limited opportunism such as those restricted to opportunistic communication when devices are within each other's range (Chau, 2011).

## 4. Use case for ad hoc networking of UAVs

In this section we describe the use case scenario we simulated to demonstrate the efficacy of Oppnets in UAV ad hoc networking for defense operations. A visualization of the scenario can be seen in Fig. 1. The basic setting of the scenario is as follows. A Carrier Strike Group (CSG)—with one carrier and four Littoral Combat Ships (LCSs)—is deployed on Maritime Interdiction Operations (MIO) and mine clearing missions in a dangerous area. The LCS/CSG force conducts surveillance of likely transit routes, using surface and aerial assets, as well as off board surveillance systems (such as acoustic arrays).

A Northrop Grumman X-47B (2012), an Unmanned Combat Air System (UCAS), is deployed from the carrier for a maritime intelligence, surveillance and reconnaissance mission over the

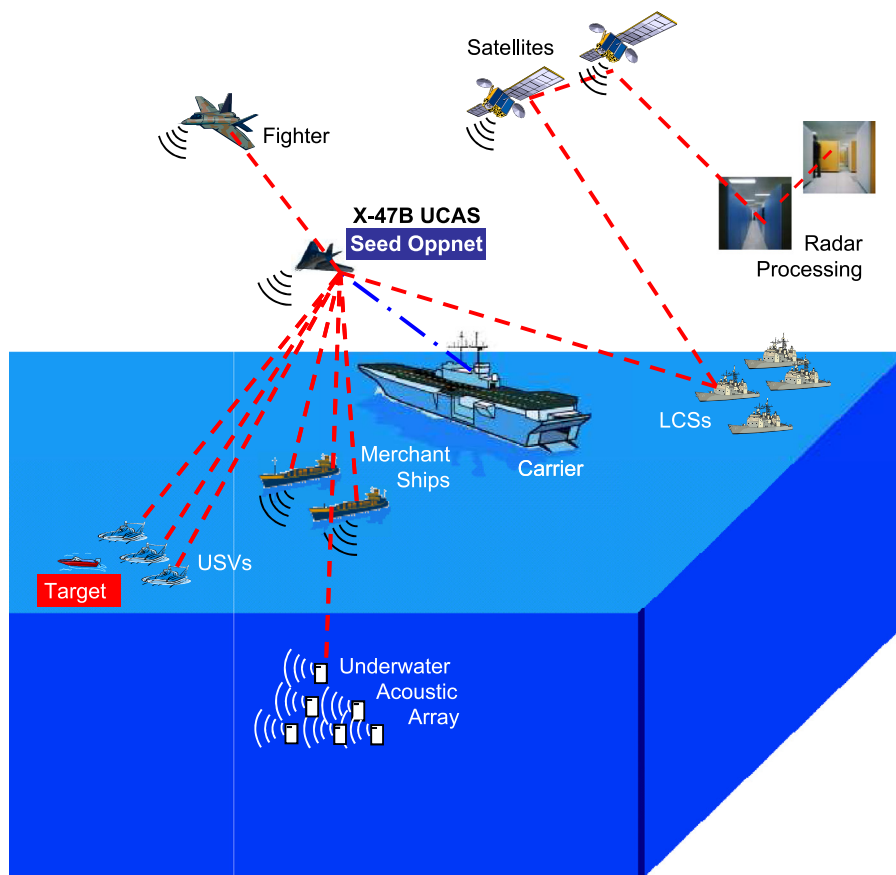


Fig. 1. Visualization of the use case.

whole Area of Responsibility (AOR) of the LCS/CSG force. The X-47B is a single-node Seed Oppnet. At the beginning of its mission, the UCAS contacts (either directly or via unnamed communication helpers) the gateway of the Acoustic Array; the gateway is an Oppnet reservist listed in the Oppnet Directory, OD. X-47B orders the gateway to become its Helper 1, and receives a report from it. The report indicates the location of a small suspicious watercraft but does not identify it unequivocally as a threat<sup>1</sup>. The X-47B is too far from the location of the watercraft to see it with its own sensors in order to evaluate if and what threat they present.

Scanning the merchant marine radio frequencies, the X-47B discovers two merchant ships within its communications range and close to the area of the sightings. It asks them to become its Helper 2 and Helper 3, respectively, and to provide their radar plots produced (primarily for collision avoidance) by their Automatic Identification Systems (AISs). The individual radar plots received by the UCAS need now be combined into an integrated radar plot, and analyzed to find radar signatures of the suspicious watercraft, and find its precise location.

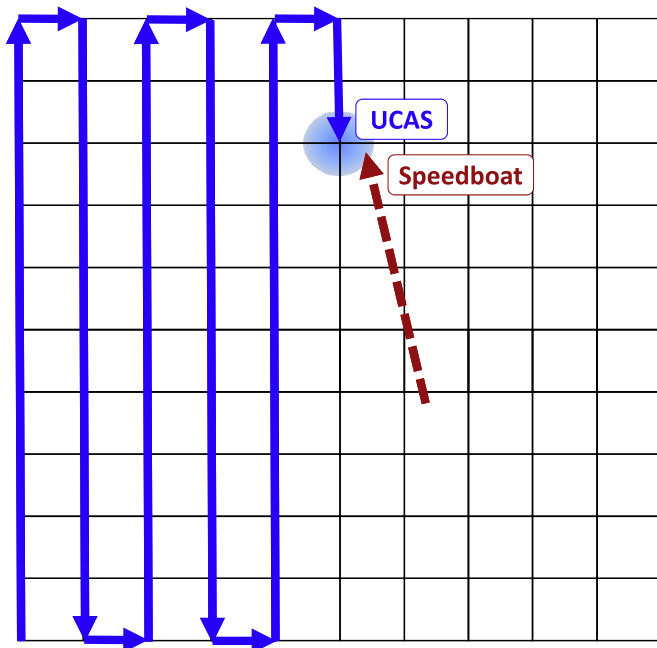
The UCAS could do this computation-intensive job by itself, but it would be relatively slow. Therefore, it checks its Oppnet Directory for a helper with a powerful computing system able to integrate the plots. The directory indicates that a needed helper, Helper 4, is available on board of LCS1, the closest LCS. Helper 4 produces the integrated radar plot. However, Helper 4 does not have the specialized software to analyze the integrated radar plot

itself. Helper 4 checks its Oppnet Directory, and finds out that a system with needed skills, Helper 6, can be contacted via another listed Oppnet reservist, the Advanced Extremely High Frequency (AEHF) satellite communication system. The satellite becomes Helper 5 and is told to pass the integration-and-analysis task to Helper 6 (which requires transmitting the task over a few infrastructure-based hops, involving other AEHF satellites and ground stations). Helper 6 analyzes the integrated radar plot searching for watercraft signatures, and returns the results to the UCAS via Helpers 5 and 4.

The results received by the UCAS are negative: no suspicious watercraft was found in the areas covered by the AISs of the merchant ships, which overlap only a portion of the area monitored by the underwater Acoustic Array. All the steps above take just seconds, so the UCAS is still too far from the area where the suspicious watercraft was spotted to use its own sensors for searching for it.

An F/A-18E Hornet strike fighter (2012), returning from its mission, happens to be passing by at this very moment, nearly over the location at which the Acoustic Array reported the suspicious watercraft. The fighter, a non-listed non-reservist, is discovered by the UCAS. The UCAS asks it to become its next helper, Helper 7, and to use its sea surface search capabilities to look for the suspicious watercraft. Helper 7 finds the suspicious object, determines that it is a speedboat, and reports its precise location to the UCAS. The UCAS now searches its Oppnet Directory for Unmanned Surface Vehicles (USVs) that can become its helpers, and intercept the speedboat. Three Spartan Scout USVs patrolling nearby become Helpers 8, 9 and 10 of the UCAS. They move at the full speed toward the suspicious speedboat to apprehend it. USVs use both radio communications and loudspeakers to order the speedboat to stop. Because the watercraft does not slow down, they fire warning shots

<sup>1</sup> The Acoustic Array might be too far from the watercraft or might be malfunctioning due to either internal factors (e.g., nearly exhausted batteries) or external factors (incl. enemy actions).



**Fig. 2.** The UCAS localizing a speedboat in the *No Oppnet* scenario (with the circle illustrating the range of the UCAS' sensors at the moment when it detects the speedboat).

from their remotely-operated .50-caliber deck guns. The speedboat stops and is apprehended. Helper 9 is the first to notify the UCAS of the success. The UCAS releases Helpers 9 and 10, and orders Helper 8 to stay with the speedboat until a manned Navy ship arrives.

## 5. Simulation setup

### 5.1. Two simulation scenarios

We simulated the use case scenario described in Section 4 in Java using the simulation package SimJava 2 (SimJava, 2012).

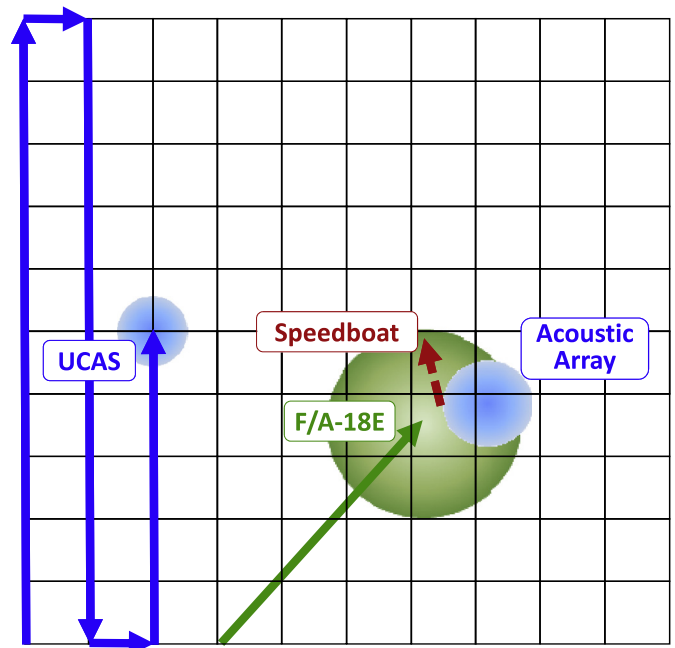
The goal of the simulation is to compare the time required to identify the location of a speedboat without the help of Oppnet helpers and the time required with the assistance of Oppnet helpers. To this end, we developed and compared two main scenarios through which the UCAS could accomplish its mission: In the *No Oppnet scenario*, UCAS patrols AOR and does not use any helpers.<sup>2</sup> In the *Oppnet Use scenario*, UCAS patrols AOR and uses all available helpers (also including more than one F/A-18E Super Hornet helper shown in the scenario description above).

### 5.2. The *No Oppnet* scenario

In the *No Oppnet* scenario, the UCAS patrols the entire AOR and uses exclusively its own sensors to detect the speedboat as illustrated in Fig. 2. This means that, in particular, none of the Helpers 1–10 is used by the UCAS. The AOR is represented as a rectangular area (though a special case of a square is shown in Fig. 2).

The UCAS follows a predefined snake-like flying pattern within AOR, as shown with a solid line in Fig. 2, starting at the bottom-left corner of the AOR. The UCAS has a surveillance range indicated with a circle. To simplify the simulation of the UCAS surveillance/

<sup>2</sup> A careful reader will notice that the *No Oppnet* scenario is equivalent to using only Seed Oppnet capabilities (in our scenario, using only UCAS' capabilities).



**Fig. 3.** The UCAS (the Oppnet Seed) and F/A-18E (a helper) localizing a speedboat in the *Oppnet Use* scenario (with the arrowhead tip for the broken speedboat line indicating the speedboat position at the moment of its detection, the arrowhead tip for the solid F/A-18E line indicating the F/A-18E position at the moment when it detects the speedboat, the larger circle illustrating the range of the F/A-18E at the moment when it detects the speedboat, and the smaller circle illustrating the range of the Acoustic Array). Note that the ranges (circles) are not in the same scale.

sensing range, we assume that the UCAS present at the edge of a square with side length  $s$  can sense anything within two bands of width  $s/2$  on its both sides (we allow for 90-degree on-the spot turns in our simulation).<sup>3</sup>

The moving pattern of the speedboat is as follows: It begins at a starting point,<sup>4</sup> *InitialOrReportedSpeedboatPosition*, selected within AOR at random, and proceeds on a straight line toward a random point, *Target Final Speedboat Position*, at the top edge of the AOR rectangle. The speedboat reaches its end point at the top edge only if it is *not* detected by the UCAS; otherwise, the end of its route is at the point of its detection by the UCAS. The dashed line in Fig. 2 shows an example speedboat route in case of its detection by the UCAS.

### 5.3. The *Oppnet Use* scenario

In the *Oppnet Use* scenario, illustrated in Fig. 3, the UCAS patrolling the AOR is an Oppnet Seed, and the Oppnet uses all helpers it is able to muster. Immediately after starting its mission, the UCAS obtains information from Helpers 1–3, that is, obtains the location of a suspicious watercraft from Helper 1 (the Acoustic Array) and the radar plots from Helpers 2 and 3 (the merchant ships). From now on, the simulation is limited to the activities of Helpers 4–7 (the system integrating radar plots on board of the

<sup>3</sup> To be precise, the sensor ranges are circular, with diameter  $s$ . But, when moving, the area swiped by them can be visualized as two bands of width  $s/2$ .

<sup>4</sup> To assure fairness of our performance comparisons, the starting point for the speedboat is the same as the starting point in the next, *Oppnet Use* scenario. The difference is that in this *No Oppnet* scenario the UCAS does *not* know about this point, and we may call this point the *InitialSpeedboatPosition*. In the *Oppnet Use* case in the next section, the UCAS is the seed Oppnet and is notified by the Acoustic Array helper about the location of a suspicious watercraft; there we may call this point the *ReportedSpeedboatPosition*. Since *InitialSpeedboatPosition* and *ReportedSpeedboatPosition* are identical, we use the single name, *InitialOrReportedSpeedboatPosition*, for both.



LCS1, the AEHF satellite, the system that analyzes the integrated radar plot, and the fighter, respectively). Only after Helpers 4–7 complete their tasks, as simulated (and described below in detail), Helpers 8–10 are interacted with.

The UCAS follows the same predefined snake-like moving pattern as described for the *No Oppnet* scenario, shown with a solid line in Fig. 3. UCAS surveillance/sensing range is also defined as in the *No Oppnet* scenario. However, this time by using helpers it can detect objects in areas not covered by its own sensors (but covered by sensors of a helper).

The speedboat follows the same linear route as in the *No Oppnet* scenario. However this time the starting point for the speedboat, *ReportedSpeedboatPosition = InitialOrReportedSpeedboatPosition*, is known to the UCAS (the Oppnet), so—after it musters helpers—it can direct them toward this point.

One more comment is needed. The starting point is selected as a random point within the AOR. In the *Oppnet Use* case this represents a situation when the Acoustic Array reports this point as a position of a suspicious watercraft detected by its sensors; this means that the location of the Acoustic Array changes for each simulation run (since the starting point must be within the sensor range for the Array). This matters only in conceptual terms (as explained here) but does not affect the simulation code since the location per se of the Acoustic Array is not simulated.

Let us assume that in the *Oppnet Use* scenario UCAS tries to find a helper to check the location of a possible enemy object (as reported by Helper 1, the gateway of an Underwater Acoustic Array). UCAS finds an ad hoc helper, an F/A-18E fighter, and requests that the fighter plane uses its sensors while flying over the area in which a possible enemy object can be located.

We assume that the fighter plane moves at a constant speed on a straight line from the initial fighter position at the bottom of the AOR toward the location of the “suspicious object” as detected by the Underwater Acoustic Array (which is the initial speedboat location for the simulation). The larger of the two overlapping circles in Fig. 3 is the area covered by the fighter's sensors, and the

smaller one is the area covered by the Underwater Acoustic Array. If the fighter detects the speedboat, the fighter location at which detection occurred is the final fighter position (and the final simulation point). Otherwise, the final fighter position is the intersection of its straight-line course and an edge (left, right, or top) of the AOR.

#### 5.4. Parameters and random variables for the simulation

Table 1 lists and describes the parameters used in the simulation. Table 2 lists and describes the random variables and their statistical distributions.

## 6. Experiment results and discussion

There are two possible mission outcomes for the UCAS:

1. It succeeds in localizing the speedboat. This occurs if it detects the speedboat (by itself or with a fighter helper's assistance) while patrolling the AOR.
2. It fails in localizing the speedboat. This occurs if the speedboat gets outside the AOR without being localized by the UCAS or its helpers.

The effectiveness of Oppnets for the simulated task was evaluated based on two main measures: (i) the *success rate* for detecting the speedboat (the number of times the speedboat is detected by the Oppnet divided by the number of total simulation runs); and (ii) the *average speedboat detection time* (the average of the time periods between the start of the simulation and the detection of the speedboat by the Oppnet, only for runs with successful speedboat detection).

In the experiment result graphs, *No Oppnet* represents the case where the UCAS does not use Oppnets and *Oppnet Use* represents the case where all Extended Oppnet capabilities (all available

**Table 1**  
Simulation parameters.

Parameter	Value	Description
<i>N</i>	1000	The number of simulation runs averaged for each result point
<i>AreaMaxX</i>	100	Maximum value for the <i>x</i> coordinate defining AOR (miles)
<i>AreaMaxY</i>	100	Maximum value for the <i>y</i> coordinate defining AOR (miles)
<i>UCASSpeed</i>	300	Speed of the UCAS (mph)
<i>UCASSensorRange</i>	10	The radius for the circular range of the UCAS sensors (miles)
<i>SpeedboatSpeed</i>	90	Cruising speed of the speedboat in calm waters (mph)
<i>FighterSpeed</i>	777	Cruising speed of the F/A-18E (mph)
<i>FighterSensorRange</i>	20	The radius for the circular range of the F/A-18E sensors (miles)
<i>OppnetDelayMin</i> <sup>a</sup>	3	Minimum value for the delay in discovering and integrating the F/A-18E helper by UCAS (min)
<i>OppnetDelayMax</i>	66, 33, 22, 16, 13	Set of simulated maximum values for the delay in discovering and integrating the F/A-18E helper by UCAS (min)

<sup>a</sup> The helper integration delay range is defined by *OppnetDelayMin* and *OppnetDelayMax*; these minimum and maximum values were selected to assure that we have a sufficient range of possible values for helper integration delay. It is safe to make this range broader than necessary, while making it too narrow would reduce usability of simulation results.

**Table 2**  
Simulation random variables.

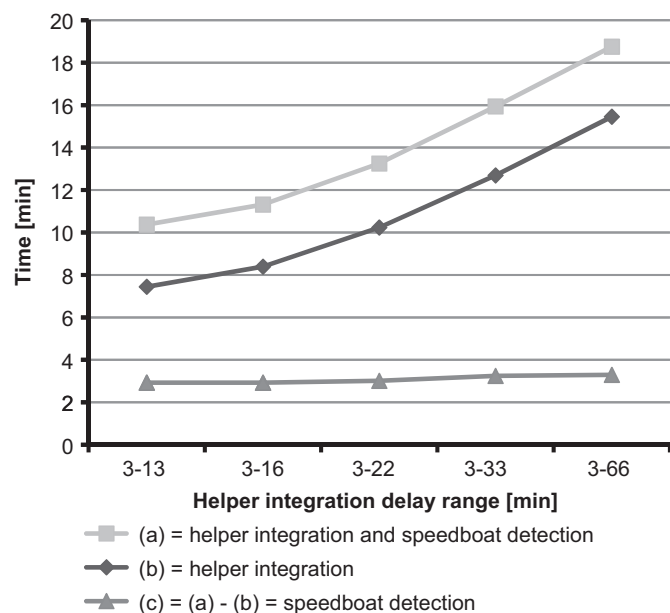
Random variable	Value range	Distribution	Description
<i>InitialOrReportedSpeedboatPosition</i>	<i>xval2</i> : 0–AreaMaxX <i>yval2</i> : 0–AreaMaxY	Uniform	The initial position for the <i>No Oppnet</i> scenario or the reported (by the Acoustic Array) suspicious watercraft position for the <i>Oppnet Use</i> scenario: ( <i>xval2</i> , <i>yval2</i> )
<i>TargetFinalSpeedboatPosition</i>	<i>xval3</i> : 0–AreaMaxX	Uniform	The target final (may be never reached) position for the speedboat (at the top of the AOR) is ( <i>xval3</i> , 100)
<i>InitialHelperPosition</i>	<i>xval1</i> : 0–AreaMaxX	Uniform	The point (at the bottom of AOR) at which the F/A-18E enters the AOR is ( <i>xval1</i> , 0)
<i>TimeToIntegrateHelperByUCAS</i>	<i>t</i> : 3–OppnetDelayMax	Uniform	This is the time before UCAS can start using F/A-18E as a helper

helpers) are used for the detection task. For each type of experiment in the *Oppnet Use* cases, we also demonstrate the effect of having multiple fighter helpers (vs. a single fighter helper). Note that unless it is explicitly stated, the discussed results consider a single fighter helper.

### 6.1. Invasive use of fighter helpers

In all subsections but the next one, we consider the cases in which the Oppnet can order all its helpers (including a fighter helper or fighter helpers) to help with detecting the speedboat. For example, consider a fighter; upon the Oppnet's command, the fighter must immediately change its original course and start flying toward the point of suspicious object detection. This is an *invasive* use of the fighter helper since the actions cannot be autonomously performed by the fighter's subsystems without a fighter's pilot involvement. Instead, the actions (incl. change of the fighter's route) must be communicated at a high priority to the fighter's pilot, and are highly visible to him.

The two higher-lying curves in Fig. 4 compare the following average times to accomplish tasks (for the runs with successful speedboat detection and a single helper): (a) the average total time needed for Oppnet's helper integration and speedboat detection; with (b) the average time needed for the Oppnet's helper integration alone. The figure shows that the latter is proportional to the former. In other words, the faster the Oppnet completes the helper integration phase, the shorter is the total time till the speedboat is detected. The reason for this dependency is as follows: If a helper is integrated sooner, the speedboat has less time to get away from the position  $(xval2, yval2)$  where a "suspicious object" was detected by the Acoustic Array. A fighter helper, once integrated, takes course from its initial position  $(xval1, 0)$  toward the  $(xval2, yval2)$  position. For a sufficiently long time needed for completing helper integration alone, the position of the speedboat becomes outside of the detection range of the helper flying over the location  $(xval2, yval2)$ . Therefore, the probability of detecting a speedboat by the helper decreases if



**Fig. 4.** Comparison of time periods (only for the runs with successful speedboat detection): (a) the average total time needed for helper integration and speedboat detection; (b) the average time needed alone for helper integration; and (c) the average time needed alone for speedboat detection (after completing integration); this time is equal to the difference between the two preceding average times (all results for a single helper).

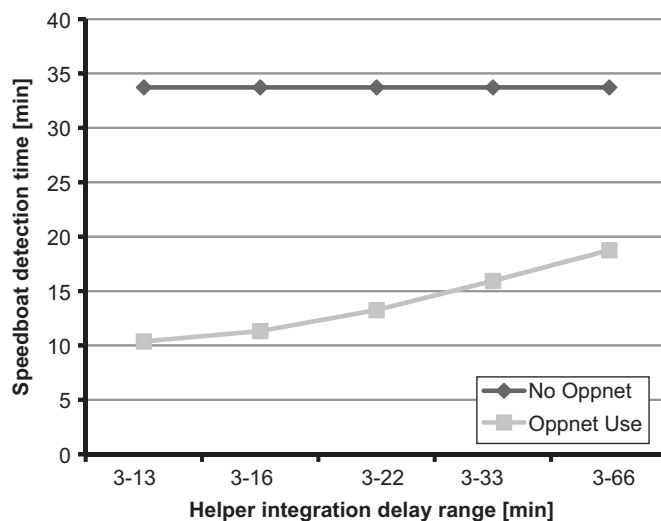
the speedboat has more time to move away from the  $(xval2, yval2)$  position.

The lowest curve in Fig. 4 shows the average time needed alone for speedboat detection (after its helper integration); it is the difference between the two other curves from Fig. 4. Interestingly, this time varies very little (from 2.93 min for the integration delay ranges 3–13 and 3–16 min, increasing monotonously to 3.29 min for the range 3–66 min). The reason for a slow increase of this time with the increasing upper bound of the range is as follows: An Oppnet's detection phase using a helper can start only after completion of the integration of this helper into the Oppnet. A longer helper integration phase gives the speedboat more time to move further away from the point of detection of the suspicious object by the Acoustic Array. Hence, the fighter (the helper) has to fly further to detect the speedboat, which extends its speedboat detection time.

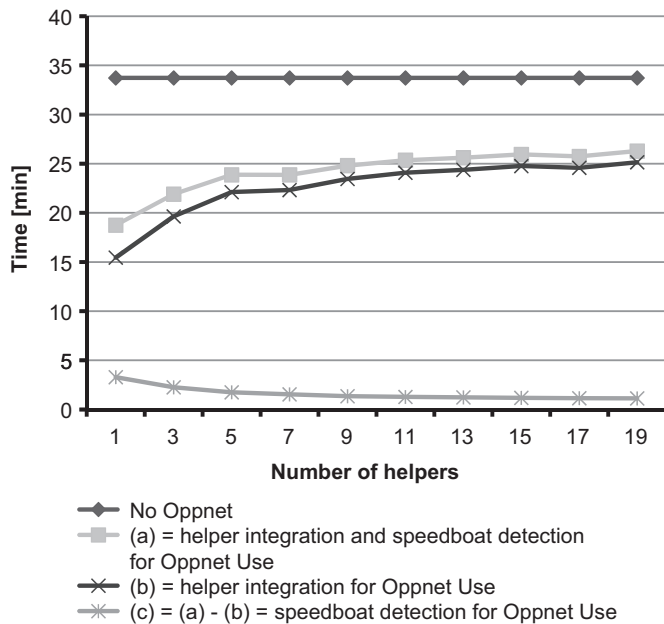
Figure 5 shows the average time between the start of a simulation run and the detection of the speedboat by the Oppnet (which includes the helper integration phase and the speedboat detection phase) for a single helper. We observe that the *Oppnet Use* results are about 2–3 times better than the *No Oppnet* results.

Figure 6 compares the *Oppnet Use* results with the baseline *No Oppnet* results for the runs with successful speedboat detection only, for multiple fighter helpers. First of all, comparison of Curves *No Oppnet* (baseline) and *a* shows that the *Oppnet Use* results are about 1.3–1.7 times better than the *No Oppnet* results.

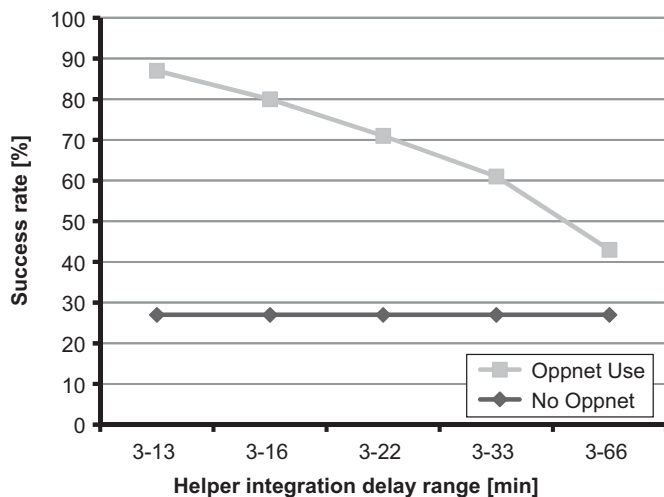
Curve *a* represents the average total time needed for helper integration and speedboat detection vs. the number of fighter helpers. Curve *b* represents the average time needed alone for helper integration vs. the number of fighter helpers. The delays (times) needed for integrating helpers for all curves are in the range 3–66 min. Both curves grow when we increase the number of helpers. The reason for the growth of Curve *b* is that the Oppnet needs more time to integrate more helpers. The reason for the growth of Curve *a* is that the growth of time needed for its helper integration phases exceeds the time savings due to shrinking of time needed for its speedboat detection phases. However, as shown by Curve *c* (which is the difference between Curves *a* and *b*), the average time needed alone for speedboat detection decreases (although at a decreasing rate) with the number of helpers. The reason for this is that a higher number of helpers increases the probability of faster speedboat detection.



**Fig. 5.** Comparison of the average total times needed for helper integration and speedboat detection for the *No Oppnet* and the *Oppnet Use* scenarios vs. helper integration time (delay).



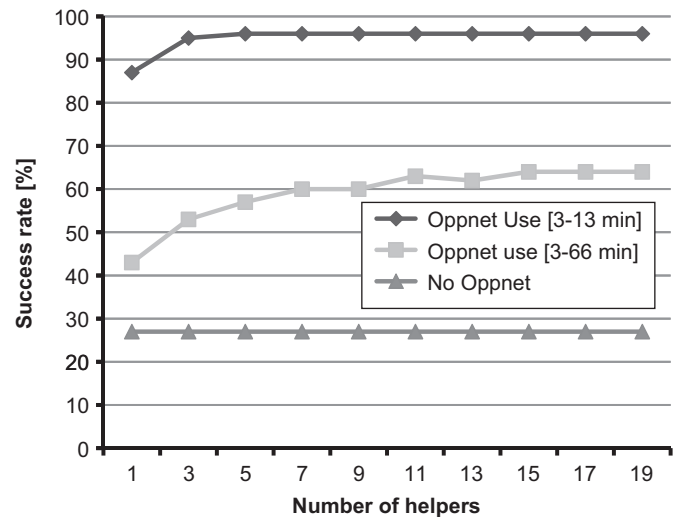
**Fig. 6.** Comparison of the average times till speedboat detection for the *No Oppnet* and the *Oppnet Use* scenarios vs. the number of helpers (for the integration delay range of 3–66 min). Curve *No Oppnet* is the baseline. The remaining curves, for the *Oppnet Use* cases, are: (a) the average total time needed for helper integration and speedboat detection; (b) the average time needed alone for helper integration; (c) the average time needed alone for speedboat detection, which is equal to the difference between the two preceding average times.



**Fig. 7.** Comparison of the success rates for the *No Oppnet* and *Oppnet Use* scenarios vs. the helper integration delay (for a single helper).

Figure 7 provides a comparison of the success rates for the *No Oppnet* and the *Oppnet Use* scenarios. The immediate observation in this graph is that with *Oppnet Use* we achieve a significantly higher success rate than for *No Oppnet*, especially when the fighter helper integration delay is low (in the 3–13 min range). The reason is that for short integration delays the speedboat will have less time to move away from its reported location ( $xval2$ ,  $yval2$ ) before the helper approaches this location. This results in a higher probability of the speedboat being in the detection range of the helper when it is flying over the location ( $xval2$ ,  $yval2$ ).

Figure 8 provides a comparison of the success rates for the *No Oppnet* and the *Oppnet Use* scenarios in the presence of multiple helpers for the integration delay ranges 3–13 and 3–66. We



**Fig. 8.** Comparison of the success rates for the *No Oppnet* and *Oppnet Use* scenarios vs. the number of helpers (for the integration delay ranges of 3–13 and 3–66 min).

observe that for *Oppnet Use* we achieve close to 100% success rate in the presence of sufficiently many helpers and with a low integration delay. Depending on the integration delay, for *Oppnet Use* we achieve from about 1.5 to over 3 times higher success rates than with *No Oppnet*, with the best improvements with respect to *No Oppnet* achieved for many helpers and shorter integration delays.

## 6.2. Less invasive use of fighter helpers

All subsections except this one cover the cases of *invasive* use of all helpers in which the Oppnet can *order* any helper to help with detecting the speedboat.

In this section we consider a *less invasive* use of *fighter* helpers (all other helpers must still follow the Oppnet's orders). This means that a candidate fighter helper is not ordered but just asked by the Oppnet to help. A request (in contrast to an order) can be refused, in which case the fighter just continues on its original course instead of changing its route. (As a consequence the fighter's pilot needs only a low-priority notification, which might be barely visible and even go unnoticed by the pilot.)

This does *not* mean that *all* help is refused since speedboat detection tasks might still be performed (albeit less effectively) along the *original* (unchanged) fighter's route. Our simulation assumes that such less invasive help is never refused. In this case, the probability of speedboat detection by the fighter helper depends on the probability that the positions of the speedboat and the fighter helper get close enough for detection at some time.

Figure 9 compares the average times till speedboat detection (including helper integration) for *invasive* and *less invasive* helper cases vs. different helper integration range values (and for a single helper). We observe that the average time till speedboat detection is about 5–8 min (about 42–45%) longer for the less invasive help mode than its *invasive* mode counterpart for any integration delay range.

Figure 10 compares the average times till speedboat detection (including helper integration) for *invasive* and *less invasive* helper cases vs. the number of helpers (and the helper integration delay of 3–66 min). The curves for the *invasive* and *less invasive* help modes are very close to each other for more helpers, becoming almost indistinguishable when there are 11 or more helpers. These results suggest that in the presence of sufficiently many helpers, the less *invasive* mode of helper operation (with the helpers continuing on their original routes) does not impair the speedboat

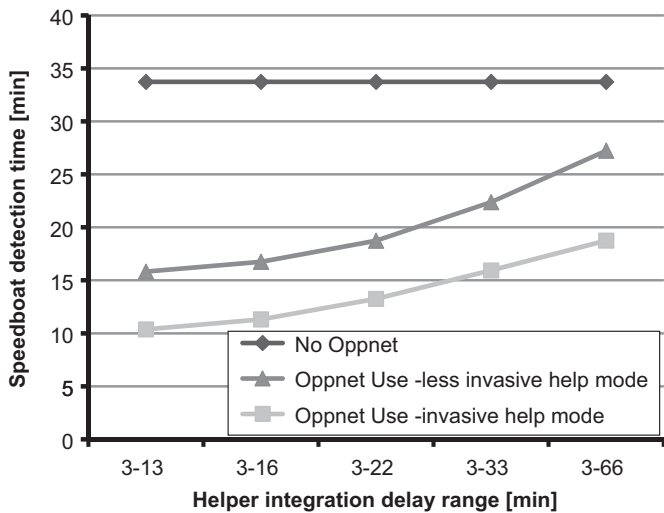


Fig. 9. Comparison of the average times till speedboat detection for the *Oppnet Use* scenarios with invasive and less invasive use of the helper vs. the helper integration delay (for a single helper).

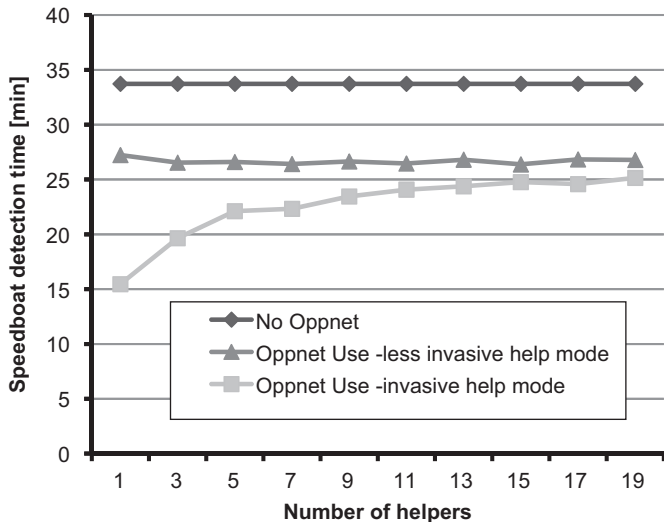


Fig. 10. Comparison of the average times till speedboat detection for the *Oppnet Use* scenarios with invasive and less invasive use of the helper vs. the number of helpers (for the helper integration delay 3–66 min).

detection task. The reason is that with more fighter helpers flying on many different routes, one of them will get sufficiently close to the suspicious watercraft to identify and locate it. This is promising for the real world application of the proposed approach, since in reality the fighter planes might be involved in other tasks too, making it difficult to change their route immediately.

Figure 11 shows the comparison of the success rates for the *Oppnet Use* scenarios with invasive and less invasive use of the helper vs. the helper integration delay (for a single helper). An important observation here is that the success rate for the less invasive help mode is about 1.2–1.8 times lower than for the invasive help mode (with the gap growing with the helper integration delay). This is mainly due to the presence of only a single helper, which continues on its original course. In this case, the fighter helper could even be moving away from the location of the suspicious watercraft while continuing its original route, which will make the speedboat detection impossible.

Figure 12 shows the comparison of the success rates for the *Oppnet Use* scenarios with invasive and less invasive use of the helper vs. the number of fighter helpers (for the helper integration

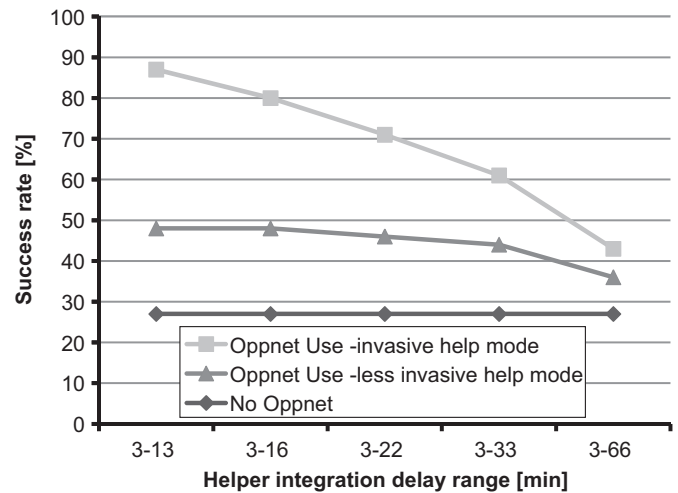


Fig. 11. Comparison of the success rates for the *Oppnet Use* scenarios with invasive and less invasive use of the helper vs. the helper integration delay (for a single helper).

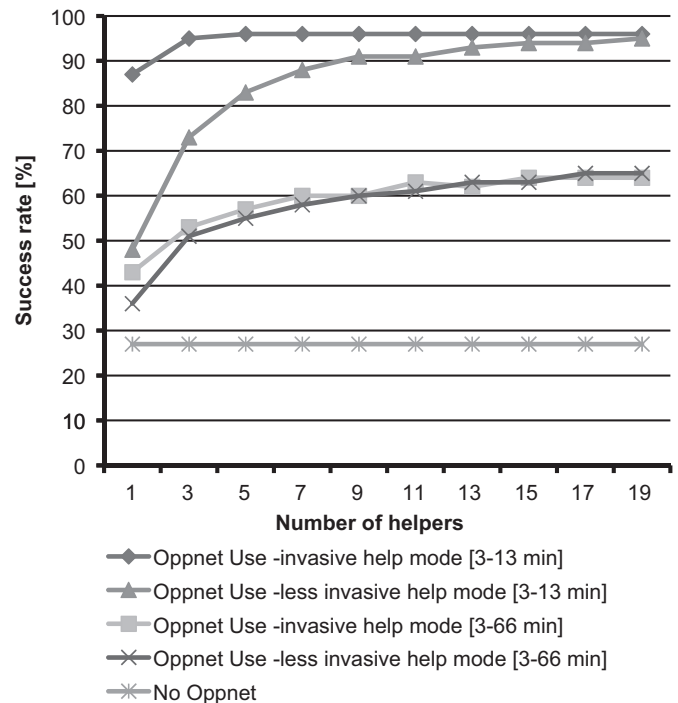


Fig. 12. Comparison of the success rates for the *Oppnet Use* scenarios with invasive and less invasive use of the helper vs. the number of helpers (for the helper integration delays 3–13 min and 3–66 min).

delays 3–13 min and 3–66 min). We see that the curves for the invasive and less invasive help modes for the integration delay range 3–66 min are almost indistinguishable in the presence of 3 or more helpers. We observe a similar trend for the integration delay range 3–13 min, except that the curves for the invasive and less invasive help modes become nearly identical only for at least 13 helpers.

This means that invasive help results in faster detection than less invasive help for quicker helper integration (3–13 min) till we have a lot of (at least 13) fighter helpers; in contrast, invasive help provides no detection speedup as compared to less invasive help for slower helper integration (3–66 min) for multiple helpers. This is due to the fact that slower helper integration allows the speedboat to get so far away from the point of suspicious watercraft sighting that deflecting



the routes of fighter helpers to that point (in the invasive help mode) does not provide advantages as compared to not deflecting routes of fighter helpers to that point (in the less invasive help mode).

Even in the worst simulated case of *Oppnet Use* with the less invasive help mode (with only a single helper and integration delay range 3–66 min), we still achieve about 32% higher success rate than for the *No Oppnet* case.

### 6.3. Denial of help by a candidate helper

In this set of experiments we consider the possibility of denial of help by some of the candidate helpers upon request to join the *Oppnet* due to their involvement in a higher priority mission. This means that a candidate refuses helping the *Oppnet* at all—even in the less invasive manner. If a candidate helper denies help, it is not integrated in the *Oppnet* and has no role in the speedboat detection.

We experimented with three different probabilities of help denial values: 0.2, 0.6 and 0.8, for any single candidate at each run of the simulation.

Figure 13 shows the comparison of the success rates for the *Oppnet Use* scenarios with no help denial and help denials (with different probabilities) vs. the helper integration delay (for a single helper).

The curves indicate that the success rate for the *no denial* cases is from about 1.43 times higher (for the highest helper integration delays) to even 2.15 times higher (for the lowest helper integration delays) than the success rate for the case of denial with probability 80%. Even the lowest simulated denial probability 20% reduces the success rate from barely noticeable (for the highest helper integration delays) to 1.17 times (for the lowest helper integration delays).

The reason of the low/lower degradation of success ratios for the highest helper integration delays can be explained as follows. A slower helper integration allows the speedboat to get so far away from the point of suspicious watercraft sighting that the higher number of fighter helpers (all that were asked to help)

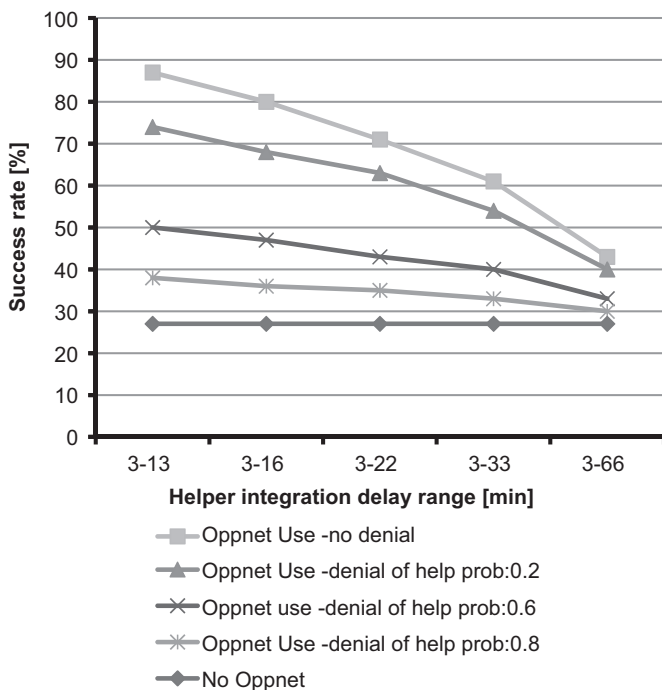


Fig. 13. Comparison of the success rates for the *Oppnet Use* scenarios with no help denial and help denials (with different probabilities) vs. the helper integration delay (for a single helper).

deflected toward that point does not provide advantages as compared to having fewer fighter helpers deflected to that point (because some refused to help).

Figures 14 and 15 provide comparisons of the success rates for the *Oppnet Use* scenarios with no help denial and help denials (with different probabilities) vs. the number of helpers, for helper integration delays 3–13 min and 3–33 min, respectively.

As shown in Fig. 14 for the smaller helper integration delays (3–13 min), the success rates for help denial cases are significantly affected when fewer fighter helpers are used. For a single helper, they can be even 2.1 times lower than the success rates for the *no denial* cases. However, as the number of fighter helpers grows, the success rates for the *help denial* cases become close and very close to the success rates for the *no denial* cases. This is due to the fact

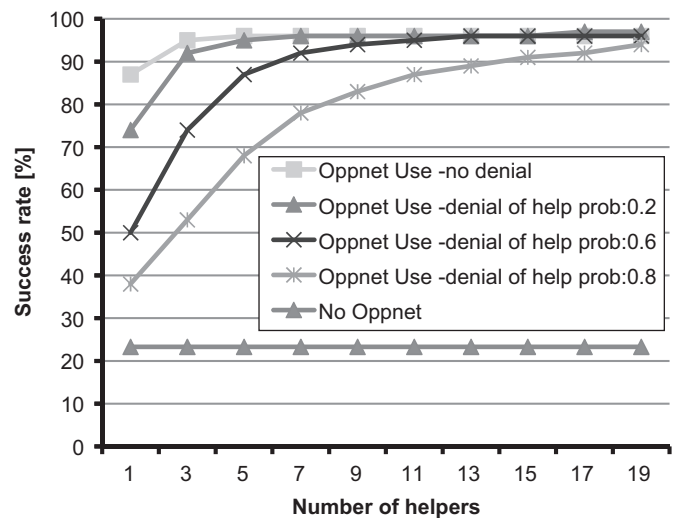


Fig. 14. Comparison of the success rates for the *Oppnet Use* scenarios with no help denial and help denials (with different probabilities) vs. the number of helpers (for helper integration delay 3–13 min).

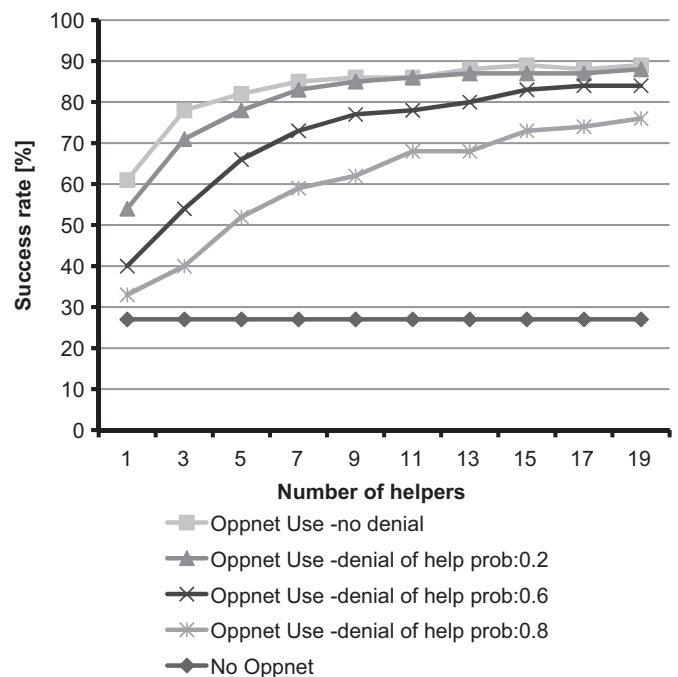


Fig. 15. Comparison of the success rates for the *Oppnet Use* scenarios with no help denial and help denials (with different probabilities) vs. the number of helpers (for helper integration delay 3–33 min).

that the availability of extra fighter helpers that decided to help makes up for the denial of help by others.

For the larger helper integration delays (3–33 min), shown in Fig. 15, the success rates for help denial cases are significantly affected when fewer fighter helpers are used. For a single helper, they can be even 1.8 times lower than the success rates for the *no denial* cases. However, as the number of fighter helpers grows, the success rates for the *help denial* cases become just 1.12 times lower than the success rates for the *no denial* cases. So, even for 19 helpers, for higher helper integration delays the success rate gap is noticeable, while this gap was negligible for the lower helper integration delays.

Why is it not negligible now? First of all, with the 80% help denial ratio, about 15<sup>5</sup> of the 19 candidate helpers deny the Oppnet's request for help. This means that the Oppnet has only 4 fighter helpers that start flying toward the point of the reported suspicious watercraft sighting as soon as their integration is completed. In contrast, for the *no denial* case, all 19 fighter helpers do so. For smaller integration delays (cf. Fig. 14 with delays 3–13 min), when the speedboat cannot get far away from the point of the sighting before helpers approach it, the 4 helpers can be as effective in terms of the speedboat detection as 19 fighters; however, for larger integration delays (cf. Fig. 15 with delays 3–33 min), when the speedboat can get further away from the point of the sighting before any helper can approach it, the 4 helpers are not a match for 19 helpers (the 19 helpers are able to cover more diverse "bands" within AOR along their 19 diverse routes).

We also observe in both Figs. 14 and 15 that for both integration delay ranges (3–13 and 3–33 min), denial of help by 20% of the candidate helpers has a quite limited effect on the speedboat detection rate (with the exception of a single helper case for smaller helper integration delays). This means that an Oppnet will perform quite well even when 20% of the candidate helpers decline to help.

#### 6.4. Effects of changing the size of AOR

Recall that Area of Responsibility (AOR) is defined as a rectangle defined by the coordinates  $\{(0,0), (0, AreaMaxY), (AreaMaxX, 0), (AreaMaxX, AreaMaxY)\}$ . In this set of experiments we consider different AOR sizes (i.e., different values of  $AreaMaxX$ ,  $AreaMaxY$ ), with square AORs ( $AreaMaxX = AreaMaxY$ ).

Figures 16 and 17 provide a comparison of the success rates for the Oppnet with multiple helpers for integration delay ranges 3–13 min and 3–66 min, respectively, with varying AOR sizes.

As seen in Fig. 16 for lower integration delays (3–13 min), the increase in the size of the AOR does not impose a significant penalty on the speedboat detection success rate. Actually, even for the 25-fold increase in the area of AOR (from  $50 \times 50$  miles to  $250 \times 250$  miles), the success rate is always close to 100% in the case of sufficiently many fighter helpers (say, 7). Only for the 100-fold increase in the area of AOR (from  $50 \times 50$  miles to  $500 \times 500$  miles), the success rate is much lower, even about 4.1 times lower for a single fighter helper; however with 19 helpers, even for this largest simulated AOR the success rate approaches the success rate for the smallest simulated AOR. Apparently, 19 fighters suffice to compensate for the AOR increase (and for the smallest AOR many fighters are "redundant"—they "detect" again an already detected speedboat).

The latest remark suggests that this figure can be used to specify how many fighter helpers suffice to reach the highest success rate. For example, to approach the highest detection rate for the  $50 \times 50$  AOR, we need just 1 helper; for the  $100 \times 100$  AOR,

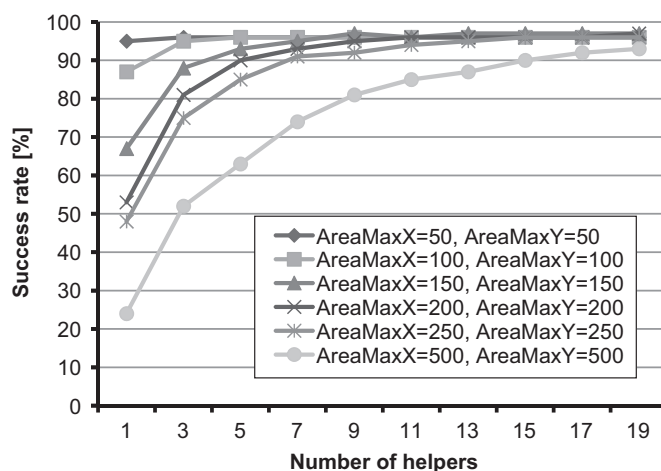


Fig. 16. Comparison of the success rates for Oppnet for different AOR sizes vs. the number of helpers (for helper integration delay 3–13 min).

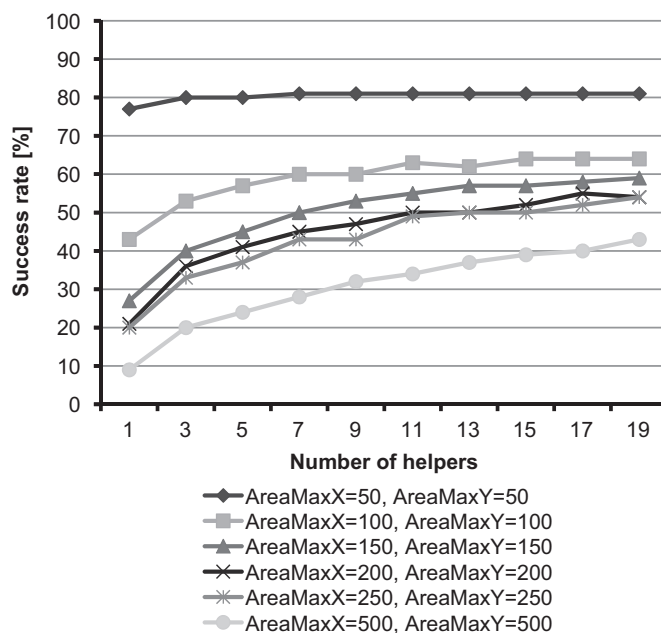


Fig. 17. Comparison of the success rates for Oppnet for different AOR sizes vs. the number of helpers (for helper integration delay 3–66 min).

we need 3 helpers; for the  $150 \times 150$  AOR, we need 5–7 helpers; ...; and for the  $500 \times 500$  AOR, we need 17–19 helpers.

We observe a more marked difference in the speedboat detection success rates for different sizes of AOR in the case of a higher integration delay (3–66 min), as can be seen in Fig. 17. In this case, the gap between the curves for different AOR sizes still tends to close as the number of helpers increases, but at a much lower rate than for low integration delay (shown in Fig. 16). A 100-fold increase in the area of the AOR (from  $50 \times 50$  to  $500 \times 500$ ) results here in about 8.6 times lower success rate for a single fighter helper; however with 19 helpers, the success rate for this largest simulated AOR is only about 1.9 times lower than the success rate for the smallest simulated AOR.

Note that the success ratio for the smallest AOR has decreased from about (almost uniform) 95% for the integration range 3–13 min to about (almost uniform) 80% for the integration range 3–66 min, that is about 1.2 times. Similarly, the success rate for the largest AOR has decreased from about 93% for 19 helpers for the integration range 3–13 min to about 43%

<sup>5</sup>  $19 \times 80\% = 15.2 = \text{approx. } 15$ .

for 19 helpers for the integration range 3–66 min, that is by about 2.2 times.

Why the best performance (for 19 helpers) for the largest AOR has been reduced approx. 2.2 times while the best performance for the smallest AOR has been reduced only approx. 1.2 times? It is due to the fact that for the smallest AOR 16–18 of 19 helpers were “redundant” (because, as Fig. 16 shows, just 1–3 helpers could do the job equally well); in contrast, for the largest AOR none of the 19 helpers was “redundant”. The redundant fighter helpers available in the smallest AOR helped to assure a smaller success rate reduction for the case with the larger helper integration delay; in contrast, lack of redundant fighter helpers in the largest AOR could not “compensate” for success rate reduction for the case with the larger helper integration delay.

Figure 18 provides a comparison of the average times till speedboat detection for different AOR sizes vs. helper integration delay (for a single helper). As expected, the speedboat detection is slower for longer integration times.

Once again, we see a marked performance difference for a 100-fold increase in the AOR area (from 50 × 50 miles to 500 × 500 miles); the ratio of speedboat detection times achieves the maximum value of about 7.2 times (about 86 min vs. about 12 min) for the highest integration delay (3–66 min). This is due to the fact that we have a single fighter helper here, so the larger AOR means more area to be searched by the Oppnet.

6.5. Effects of probability of speedboat detection by the helpers

All previous experiments assumed that a fighter helper never misses a speedboat if only it falls within its sensor range. The experiments in this section consider imperfect detection capabilities of helpers, by introducing the probability *dp* of the speedboat detection by a helper having the speedboat within its sensor range. These experiments bring the described scenario closer to real-life conditions, as imperfect sensor data collection is still a common problem in today’s systems. (Additionally, *dp* can also be

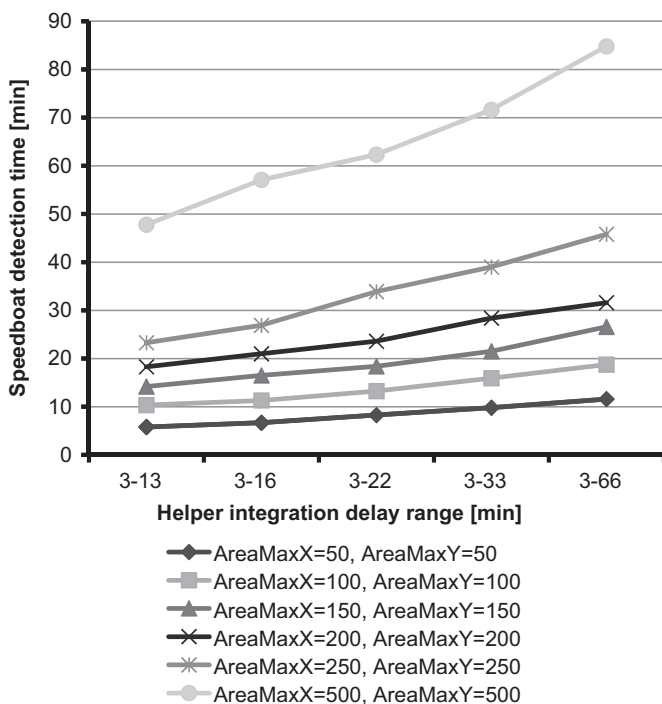


Fig. 18. Comparison of the average times till speedboat detection for different AOR sizes vs. helper integration delay (for a single helper).

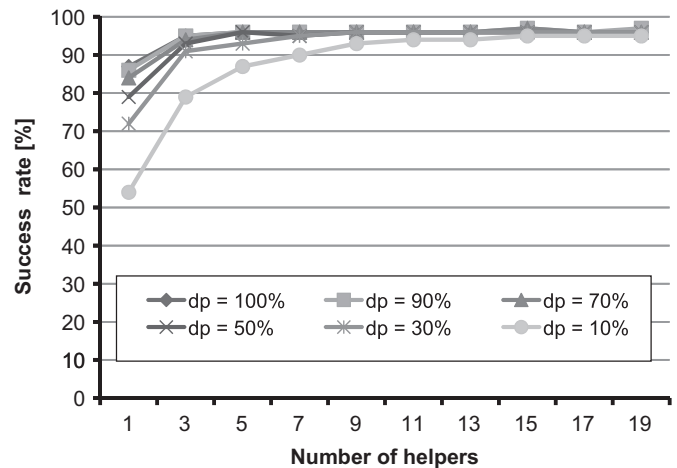


Fig. 19. Comparison of the success rates for different probabilities of speedboat detection (*dp*) by helpers vs. the number of helpers (for helper integration delay 3–13 min).

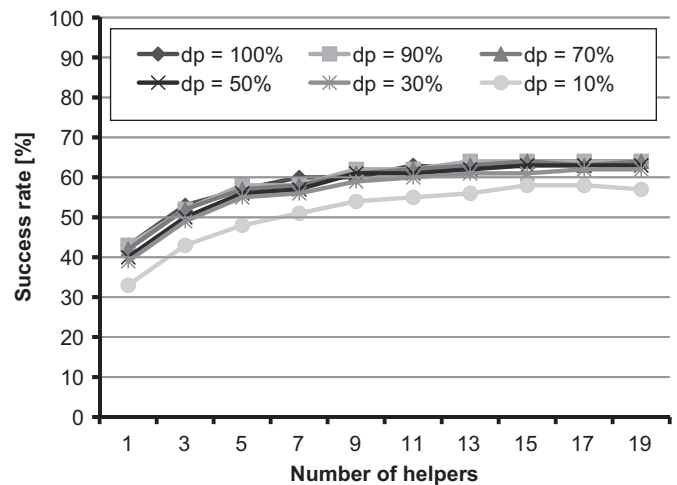


Fig. 20. Comparison of the success rates for different probabilities of speedboat detection (*dp*) by helpers vs. the number of helpers (for helper integration delay 3–66 min).

viewed as modeling environmental factors, such as rough seas, fog, etc.)

Figures 19 and 20 provide comparisons of the success rates for different probabilities of speedboat detection (*dp*) by helpers vs. the number of helpers (for helper integration delays 3–13 min and 3–66 min, respectively). The value of *dp* is varied, and selected from the set {10%, 30%, 50%, 70%, 90%}.

As we can see in Fig. 19 with curves for lower helper integration delays (3–13 min), with the exception of the cases with the lowest *dp* value (10%) and fewer than 7–9 fighter helpers, an imperfect helper’s speedboat detection capability has a very insignificant effect on the success rate. The highest impact of a low *dp* value is seen for a single helper: the worst performance—for *dp* = 10%—is about 1.6 times worse than the best performance—for *dp* = 90%.

A significant performance penalty is observed for *dp* = 10% and no more than 5–7 helpers, since for so few helpers the low detection probability *dp* is not compensated for by having sufficiently many “backup” helpers (when the next helper could detect what the previous one missed).

Figure 20 shows analogous results but for higher helper integration delays: 3–66 min). We observe here a similar behavior in terms of the effect of the speedboat detection probability *dp* on

the success rate. However, we can pinpoint a few differences. First, the curves are closer to each other; the worst performance—for  $dp=10\%$ —is now only about 1.3 times worse than the best performance—for  $dp=90\%$ .

Second, the curve for the lowest detection probability never “touches” the curve for the highest detection probability. Based on the trend seen in the “twin” Fig. 19, we believe that we would see these curves touching if we simulated cases with more than 19 helpers.

## 7. Conclusions

### 7.1. Summary of simulation results

In this paper we proposed Opportunistic Resource Utilization Networks for UAV ad hoc networking. We simulated a use case to demonstrate the effectiveness of the proposed Oppnets in homeland defense operations, where we compared two scenarios for the task of detecting a suspicious watercraft (a speedboat): not using an Oppnet vs. using an Oppnet.

In short, the results of the experiments with the simulation scenario suggest that Oppnets are a promising framework for high-performance ad hoc UAV networking, and using the Oppnet paradigm and technology for a “matching” application domain is very beneficial. Oppnets achieve high performance even under imperfect conditions—such as a less invasive use of helpers, denial of help by some of the candidate helpers, and imperfect detection of the speedboat by the helpers—which brings the simulation closer to real-life conditions. More details with most important findings follow.

#### 7.1.1. Invasive use of helpers

In the case of sufficiently many helpers (in our simulations: 5 or more) and a low delay in integrating the helpers (3–13 min), Oppnets achieve close to 100% success in the target mission. Oppnets complete the speedboat detection 2–3 times faster than when no Oppnet is used; however, slower completion of helper integration slows down speedboat detection by the Oppnet.

#### 7.1.2. Less invasive use of helpers

As compared to the invasive helper mode (when the helper is deflected toward the location of a suspicious watercraft), a delay of about 42–45% in speedboat detection by the Oppnet is observed in the case of a single fighter helper in the less invasive help mode (when the helper is *not* deflected from its original route). However, in the presence of sufficiently many (say 13–15) fighter helpers the performance with less invasive help is almost as good as the performance with invasive use of helpers.

The less invasive help mode has 1.2–1.8 times lower success rate than the invasive help mode for a single fighter helper, while the difference in the success rates is less than 10% for 7 or more fighter helpers. But even in the worst simulated case of *Oppnet Use* with the less invasive help mode (with only a single helper), we still achieve about 32% higher success rate than for the *No Oppnet* case.

#### 7.1.3. Denial of help by a candidate helper

The success rate for the no denial cases is from about 1.43 times higher to 2.15 times higher than the success rate for the case with a high denial probability 80%. The lowest simulated denial probability, 20%, reduces the success rate from a barely noticeable up to 1.17 times. The effect of denial of help on success rate is mitigated when more helpers become available; when integration delays decrease, the effects of denial of help are mitigated even further.

#### 7.1.4. Effect of changing the AOR size

Experiments with different AOR sizes show that a larger AOR does not affect the success rate of the Oppnet significantly for a low helper integration delay, but a high integration delay with a large AOR has a significant effect on the success rate.

For the lowest simulated helper integration delay, the success rate is always close to 100% if the Oppnet has sufficiently many fighter helpers (7 or more) even for the 25-fold increase of the AOR area (from  $50 \times 50$  miles to  $250 \times 250$  miles). Only for a 100-fold increase of the AOR area (from  $50 \times 50$  miles to  $500 \times 500$  miles), the success rate is much lower (about 4.1 times lower for a single fighter helper); however, with 19 helpers, even for this largest simulated AOR the success rate approaches the success rate for the smallest simulated AOR.

In terms of the speedboat detection time, we see a marked performance disadvantage only for a 100-fold increase of the AOR area (from  $50 \times 50$  miles to  $500 \times 500$  miles) where the ratio of speedboat detection times reaches the maximum value of about 7.2 times for the highest integration delay and a single helper.

#### 7.1.5. Effects of probability of speedboat detection by the helpers

The impact of imperfect helper's detection capabilities on the mission success rate is insignificant in cases with low helper integration delays. The performance is hurt only when the detection probability of any single helper is very low (10%) and there are very few helpers (fewer than 5) available for the Oppnet. The highest impact of a low detection probability value ( $dp$ ) is seen for a single helper, in which case the worst performance (for  $dp=10\%$ ) is about 1.6 times worse than the best performance (for  $dp=90\%$ ).

### 7.2. Future work

Future work will include extensions to the simulation program as well as expansion of the set of experiments. The simulation program extensions will include additional variables and components for the simulation program that will improve the realism of the simulation environment and the accuracy of simulation results.

The experiment set expansion will focus on the effects of Oppnets on computational resources (e.g., the radar plot integration services that comprise Helper 6 of the use case). We will also experiment with varying movement patterns for the speedboat, consider more helper integration delay ranges, use more random variables to quantify the average time till the Oppnet completes helper integration, and use more precise metrics (for example, introduce the vertical distance metric).

## Acknowledgments

This research was supported in part by the STTR grant, DoD Navy N10A-T006 (BP10-611).

## References

- Berioli M, Courville N, Werner M. Emergency communications over satellite: the WISECOM approach. In: Proceedings of 16th IST mobile and wireless communications summit (ISTMWC07). Budapest, Hungary; July 2007. p. 1–5.
- Chau C-K. Analysis of latency of stateless opportunistic forwarding in intermittently connected networks. *IEEE/ACM Transactions on Networking* 2011;19(4):1111–24.
- F/A-18E Hornet strike fighter. ([http://www.navy.mil/navydata/fact\\_display.asp?cid=1100&tid=1200&ct=1](http://www.navy.mil/navydata/fact_display.asp?cid=1100&tid=1200&ct=1)); November 2012.
- Fujiwara T, Watanabe T. An ad hoc networking scheme in hybrid networks for emergency communications. *Ad Hoc Networks* 2005;3(5):607–20.
- Haddal CC, Gertler J. Homeland security: unmanned aerial vehicles and border surveillance. CRS report for congress, congressional research service. (<http://www.fas.org/spp/crs/homesecc/RS21698.pdf>); July 2010.



- Han Z, Swindlehurst A, Liu K. Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles. *IEEE Transactions on Vehicular Technology* 2009;58(7):3533–46.
- Homeland Security Unmanned Aerial Vehicles (UAVs). (<http://www.homeland1.com/homeland-security-products/unmanned-aerial-vehicles-uav/>); 2013.
- Homeland security. ([http://en.wikipedia.org/wiki/Homeland\\_security](http://en.wikipedia.org/wiki/Homeland_security)); March 2013.
- Jang H-C, Lien Y-N, Tsai T-C. Rescue information system for earthquake disasters based on MANET emergency communication platform. In: Proceedings of international conference on wireless communications and mobile computing (IWCMC'09). Leipzig, Germany; June 2009. p. 623–7.
- Lilien L. A taxonomy of specialized ad hoc networks and systems for emergency applications. In: Proceedings of the first international workshop on mobile and ubiquitous context aware systems and applications (MUBICA'07). CD-ROM. Philadelphia (PA); August 2007. 8 p.
- Lilien L, Kamal Z, Bhuse V, Gupta A. Opportunistic networks: The concept and research challenges in privacy and security. In: Proceedings of the international workshop on research challenges in security and privacy for mobile and wireless networks (WSPWN'06). Miami (FL); March 2006. p. 134–47.
- Lilien L, Gupta A, Kamal Z, Yang Z. Opportunistic resource utilization networks—a new paradigm for specialized ad hoc networks. *Computers and Electrical Engineering, Special Issue: Wireless Ad Hoc, Sensor and Mesh Networks* 2010;36(2):328–40.
- Lilien L, Gupta A, Yang Z. Opportunistic networks for emergency applications and their standard implementation framework. In: Proceedings of the first international workshop on next generation networks for first responders and critical infrastructure (NetCri'07). Baltimore (MD); April 2007. p. 588–93.
- Northrop Grumman X-47B. (<http://www.as.northropgrumman.com/products/nucasx47b/index.html>); November 2012.
- Oh S, Lau D, Gerla M. Content-centric networking in tactical and emergency MANETs. In: Proceedings of IFIP wireless days conference (IFIP'10). Venice, Italy; October 2010. p. 1–5.
- Pelusi L, Passarella A, Conti M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine* 2006;44(11):134–41.
- Reidt S, Wolthusen S. Exploiting UAVs capabilities in tactical MANETs. In: Proceedings of 2nd annual conference of the international technology alliance (ACTA'08). London, UK; September 2008. p. 322–3.
- SimJava. (<http://www.icsa.inf.ed.ac.uk/research/groups/hase/simjava/>); November 2012.
- Sistla P, Wolfson O, Xu B. Opportunistic data dissemination in mobile peer-to-peer networks. In: Proceedings of 9th international symposium on advances in spatial and temporal databases (SSTD'05). Angra dos Reis, Brazil; August 2005. p. 346–63.