Opportunistic Sharing of Airborne Sensors

Jacob Beal, Kyle Usbeck, Joseph Loyall
Raytheon BBN Technologies
Cambridge, Massachusetts 02138
{jakebeal, kusbeck, jloyall}@bbn.com

James Metzler
Air Force Research Laboratory
Rome, New York 13441
james.metzler@us.af.mil

***Abstract*—Airborne sensors are often idle for much of their flight, e.g., while the platform carrying them is in transit to and from the locations of sensor tasks. The sensing needs of many other potential information consumers might thus be served by sharing such sensors, allowing other information consumers to opportunistically task them during their otherwise unscheduled time. Toward this end, we have developed Mission-Driven Tasking of Information Producers (MTIP), a prototype system for opportunistic sharing of airborne sensors. This paper describes its implementation as an agent-based task allocation system on top of the Marti Quality of Service (QoS)-managed publish-subscribe information management system, and presents simulations of a disaster response scenario demonstrating how MTIP can increase the number of sensor tasks served as well as reducing the number of UAVs required to serve a given set of sensor tasks.**

I. Introduction

Airborne sensor platforms are increasingly important sources of information for many organizations, particularly in time-critical situations such as management of emergency situations like natural disasters, industrial disasters, or civil unrest. Even as these platforms decrease in cost and increase in availability and accessibility, improvements in Geographic Information Systems (GIS) and information integration have also greatly increased the number of potential information consumers and potential uses for such information. In many cases, however, airborne sensors are currently greatly underutilized. For example, an air asset sent on a mission typically spends much of its time in transit to and from the locations where sensor tasks are intended to be executed.

This under-utilization provides an important opportunity: if the operator of an airborne platform is willing to share their sensor, then the sensor's "down time" can instead be used to gather information opportunistically as the platform passes near locations or objects of interest to other information consumers. Consider, for example, the scenario illustrated in Figure 1, in which a disaster response team is using a ScanEagle UAV to check the San Luis Dam for damage following a large earthquake in the San Francisco Bay area. The planned flight path to check the dam also flies close to many other pieces of critical infrastructure, including airports, cell phone towers, and fire departments. Sensor sharing can allow other disaster response teams, which may want information about these other locations, to survey them while the UAV is in transit to and from the dam.

In this way, the benefits of airborne sensing can be made available to organizations without their own sensor platforms.

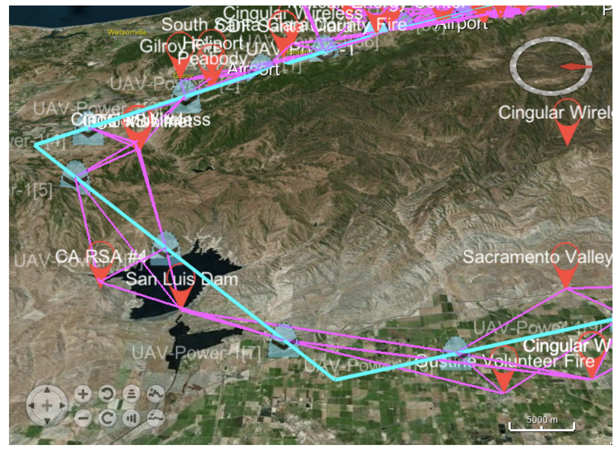


Fig. 1. Example of sensor sharing with MTIP: the planned path of a UAV flight intended to check the San Luis Dam for earthquake damage (cyan line with translucent blue air-asset tactical symbols) also flies close enough (magenta sight lines) to survey other critical infrastructure (red icons), including the San Martin airport, the South Valley Hospital and its heliport, and a number of fire departments and cell phone towers.

Organizations with their own platforms can also benefit, either by increasing efficiency by having some of their needs satisfied by others' platforms, or by increasing resilience through increasing the number of platforms asked to execute a given task.

To realize this vision of shared distributed sensing, we have implemented Mission-Driven Tasking of Information Producers (MTIP), a prototype system for sharing of airborne sensors. Following a brief review of related work in Section II, Section III describes the MTIP architecture, which joins an agent-based task allocation mechanism with the publish-subscribe framework provided by the Marti QoS-driven information management system [1], [2]. Section IV presents experimental validation of both the airborne sensor-sharing concept in general and the MTIP prototype in particular using a disaster response scenario, and finally Section V summarizes contributions and discusses future work.

II. Related Work

Many organizations already implement schemes for sharing airborne sensor data (along with many other types of information) through a variety of information management systems, especially for maintaining situational awareness in emergency or military situations (e.g., [2], [3], [4], [5], [6], [7], [8], [9]).

While these systems work quite well for sharing information to interested parties (and MTIP is in fact built upon the system described in [2]), they provide no mechanism for allocating sensors to gather that information. Sensor sharing is thus implemented by interactions between humans, leading to problems in discovery and competing with other critical tasks for their attention, which contributes to the current under-utilization of sensors.

Collective sensing from multiple platforms, whether airborne or otherwise, has been an object of considerable interest for many years [10]. Much of the work in this area has focused on networks comprising many sensors controlled by a single organization (e.g., [11], [12], [13], [14]), rather than the sharing of underutilized sensors between different "owners" and organizations. More recently, however, interest has developed in opportunistic sensing systems [15], [16], [17], particularly those that take advantage of human-carried sensors such as those found in smart phones. While a number of such systems and mechanisms for such systems have been developed (e.g., [18], [19], [20], [21], [22]), these have focused primarily on diffuse tasks of large space and time extent, such as noise or pollution monitoring, in which any given sensor is only able to contribute a small portion of the sensing capability needed for the task. For these systems, the typical use case considers allocation of a large number of sensors to a relatively small number of diffuse tasks. Moreover, there is typically little competition between tasks for sensor resources, since most scenarios investigated have primarily involved non-directional sensors. MTIP, by contrast, focuses on higher-performance sensors such as airborne cameras and more specific and localized tasks, in which the focus is instead on effective allocation of a large number of potentially competing individual tasks to individual sensors.

III. Mission-Driven Tasking of Information Producers (MTIP)

The purpose of MTIP, from a high level, is to address a deficiency the currently exists in most publish-subscribe Information Management Systems (IMS). Publish-subscribe systems decouple information producers from information consumers (thereby enabling scalability and resilience desirable in an IMS). Subscriptions are made by consumers as a way to inform the IMS what information is of interest to that consumer. This subscription request/filter is used for "brokering" messages — i.e., if the IMS receives a message that matches the subscription's criteria, it is forwarded to the consumer who made the subscription. However, it is possible that no information producers collect information that matches the subscription's criteria; in this case the consumer's information needs go unmet.

MTIP addresses this deficiency by assigning tasks to information producers that have the capability and availability to collect information that is of interest to a consumer. In particular, by opportunistically increasing sensor utilization, MTIP can be used to achieve any or all of the following goals (depending on choices in deployment):

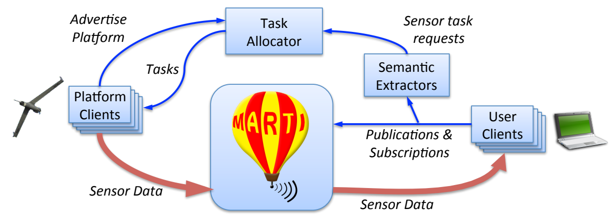


Fig. 2. Information requests into a shared information management system, such as Marti [1], [2], may go unanswered if no information producer is currently planning to gather the requested information. MTIP uses semantic information extracted from both information requests and other indications of user interest (e.g., publications of plan or location-of-interest information to other users) to task shareable sensors to gather the requested information, which is then returned to them normally via the shared information management system.

- Increasing the number of information gathering tasks that can be served by a given set of sensor platforms.
- Reducing the number of sensor platforms required to perform a given set of information gathering tasks.
- Increasing resilience of information gathering tasks by serving them on multiple platforms.

A. Context: Marti and TAK

In order to give a clear path to adoption and deployment, MTIP is designed to operate with and extend the capabilities of existing systems. In particular, MTIP augments the Team Awareness Kit (TAK) ecosystem of Situation Awareness (SA) tools. The TAK ecosystem is composed of a set of clients such as ATAK [6] and/or its Windows or iOS ports (WinTAK and iTAK respectively) and an IMS, *Marti* a.k.a. TAKServer, which is an advanced tactical information management system that can be deployed on a variety of hardware platforms and any IP network [1], [2]. While ATAK can function in serverless environments, there are times when a server is needed to bridge networks to provide Beyond Line-of-Sight (BLOS) communication [2] or provide a communications substrate when networks otherwise prevent clients from communicating directly (e.g., those separated by firewalls or lacking IP addressability, such as commercial cellular) in austere conditions [23]. In those cases, Marti acts as the IMS to decouple information producers from information consumers, allowing new producers or consumers to be added on demand.

Marti uses a publish-subscribe model: information producers (publishers) are decoupled from information consumers (subscribers). Publishers submit information to Marti with metadata describing the time, location, and content of the information. Subscribers register a request for their information needs (i.e., their *subscription*) and Marti delivers published information matching that subscription. Marti can also archive published information for consumers to query for information that was published in the past. Further, Marti provides QoS management to prioritize and shape traffic to the network's available bandwidth. Full details of the Marti system are available in [1] and [2].

B. MTIP Architecture

The MTIP architecture is built around Marti, as shown in Figure 2. With MTIP, users share information normally, by contributing it to or requesting it from Marti. These publications and subscriptions are intercepted and passed through a set of *Semantic Extractors*, which derive user interest information from this traffic, transforming it into sensor task requests that would satisfy the end-user's information needs.

At present, MTIP's semantic extractors recognize five types of information:

- Sensor Points of Interest (SPoI),
- Routes (i.e., paths),
- Areas of Interest (AoI),
- Information Requests (Subscriptions / Queries),
- Geospatial Annotations (e.g., KML).

From these, the semantic extractors produce three classes of sensor tasking requests compliant to a specified platform interface: "point" tasks focused on a discrete location, route tasks, and polygonal area tasks. Each task request can also be annotated with additional requirements information derived from the initial communication flow by the semantic extractors, including the following properties:

Property	Description
Name	Textual description of the request for display
Priority	Importance of the request
Start Time	Time by which the request must start execution
End Time	Time by which the request must be completed
Resolution	Desired ground resolution of the resultant image(s)
Sensor Mode	Desired mode for multi-modal sensors
Radius	The radius of interest surrounding the AoI
Revisit Time	Desired repetition frequency of a recurring task

These task requests are passed to a *Task Allocation* component in order to be assigned to particular platforms whose sensors can fulfill the requests. The task allocation component also receives advisements of the position, status, and path plan (if available) of all the platforms via advertisements from the platforms themselves. With this information and the current state of task allocation, the task allocation component determines which tasks will be assigned to which platforms (as elaborated in Section III-C): any given task might be assigned to one platform, multiple platforms, or even no platforms at all if none is available that can satisfy the task. Task requests are then dispatched to platforms, whose operators decide whether and how to carry them out, returning information to the task allocation component about whether they accept or reject requests they have been given, when tasks will be started and completed, and the expected quality of the resultant product.[1] The task allocation component can then adjust its assignment in response to this feedback, canceling or reassigning tasks as necessary.

[1]Note that MTIP does not currently include any explicit representation of costs, payments, or other information about incentives: this is because for the types of scenarios initially considered, these issues are typically handled separately at an organizational level through mechanisms such as mutual aid agreements and directives to organization members. Likewise, every platform is assumed to be "owned" by some organization, which has ultimate authority over its route and availability for tasking.

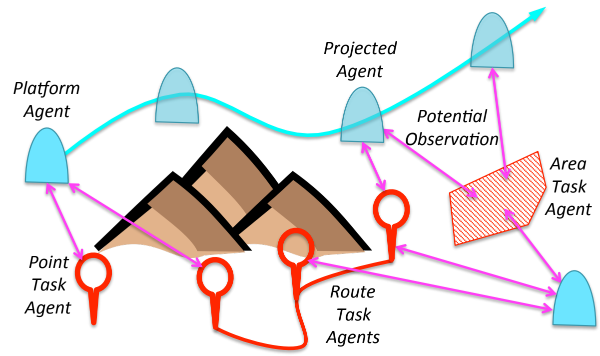


Fig. 3. In MTIP's agent-based task allocation, agents for current platforms (light blue) and projections (translucent blue) along the platform's anticipated trajectory communicate (purple arrows) with task agents (red) within line-of-sight and sensor range limitations to determine which tasks will be assigned to which platforms and on which segments of their anticipated route.

Finally, as platforms carry out their assigned tasks, the sensor data that they produce is placed under management by the Marti system, where it is ultimately delivered to users in accordance with their subscriptions.

C. Agent-Based Task Allocation

MTIP uses an agent-based approach to task allocation, which we have chosen over typical planning or optimization algorithms for three key reasons. First, with an appropriate choice of algorithm, agent-based planning executes rapidly and its current state always provides a viable (though possibly still improving) allocation plan, which means that it can be safely used in real-time situations with strict time constraints. The cost of this speed is that task allocations will generally be somewhat sub-optimal, though this need not be a significant cost (as we will see in Section IV-C).

Second, our agent-based approach is also well-suited for rapid dynamic replanning, which can be executed simply by modifying the agent representation and allowing the agents to continue interacting. Third, although currently all agents are located in the task allocator, the agent-based approach provides a natural path toward further decentralization whereby platform agents and nearby goal agents are mirrored in platform clients. This mirroring would then enable platform clients to dynamically reallocate tasks amongst themselves even when communication with the task allocator is limited or unavailable, yet still allow fast and efficient planning under normal communication conditions.

The MTIP agent-based task allocation mechanism uses two classes of interacting agents: platform agents and task agents (Figure 3). For each airborne platform, an agent is created for its current position along with a set of projected agents at intervals of a parametrically specified resolution along its anticipated future trajectory (similar to the polyagents approach presented in [24]). Likewise, one or more agents is created for each task: point tasks and other tasks with small extent in space and time are represented by a single agent;

```
let task = if(self.isTask()) { [self.getTask()] } else { [] };
let visibleTasks = unionHood(nbr(task));
// Computation of assignment state variable:
rep(assignment <- []) {
  // Task agents determine which platform agents have assigned them...
  let assignedBy = unionHood(mux(nbr(assignment).intersection(task).size()>0) { [nbr(self.getDeviceUID())] } else { [] });
  // ... and platform agents note the unassigned (but assignable) tasks nearby
  let unassigned = self.validTasks(unionHood(nbr(mux(assignedBy.size()==0) { task } else { [] })));
  // Computed the updated assignment
  if(self.isPlatform()) {
    let maybeExpanded = // With probability p_assign, assign a random unassigned task
      if(unassigned.size() > 0 && random() < p_assign) {
        [unassigned.get(floor(random()*unassigned.size()))].mergeAfter(self.validTasks(assignment))
      } else {
        self.validTasks(assignment)
      };
    // Fill out any extra time slots with visible tasks already assigned to other platforms, and discard any assignments more than the maximum
    let unassignedVisible = self.validTasks(visibleTasks).subtract(maybeExpanded);
    self.setAssignment(maybeExpanded.mergeAfter(unassignedVisible).subTupleStart(max_assigned));
  } else {
    assignedBy
  };
};
```

Fig. 4. Simple Protelis task assignment algorithm used by MTIP for the work presented in this paper.

tasks with larger extent are broken up into intervals (routes) or tiles (areas and circles) at a parametrically specified resolution.

A bipartite network between task and platform agents is then formed by creating an edge between each task agent and every current or projected platform agent that could plausibly carry out that task. This is determined by sensor range (e.g., close enough to achieve a specified camera resolution) and by line of sight, as computed from the planetary horizon and GIS terrain data—in particular, our implementation accomplishes this with the aid of NASA's WorldWind GIS environment [25]. In the work reported in this paper, all sensors and task requirements were assumed to be homogeneous and satisfiable at any time, but arbitrary information and constraints may be used to determine task/platform compatibility. The agents then communicate with one another along these edges in order to determine which tasks will be allocated to which platforms and on which segments of their anticipated route.

For the work reported in this paper, we have used the simple sub-optimal algorithm[2] shown in Figure 4, implemented in the Protelis aggregate programming framework [26], [27]. As with all programs based on field calculus [28], this algorithm executes in rounds. The `nbr` statements indicate values that are shared between agents in each round (i.e., "values from this agent's neighbors"), while `unionHood` computes the union of sets shared by neighbors, thus together implicitly implementing inter-agent communication. In this simple algorithm, devices track one state variable, `assignment`, which for platform agents is the set of tasks they are assigned and for task agents the set of platforms to which they are assigned. In each round, task agents recompute the set of platforms to which they are currently assigned and share this information with neighboring platform agents. Meanwhile, platform agents select up to `max_assigned` of their neighboring (i.e., potentially visible) and valid (i.e., non-rejected and with compatible time and sensor properties) tasks, in the following priority order:

1) One randomly selected neighboring task that is currently not assigned to any platform (but only with probability p_{assign}, for purposes of symmetry-breaking; for the experiments reported in this paper, we use $p_{assign} = 0.5$).
2) Tasks that have already been assigned to the platform agent, to promote assignment stability. If displaced by an unassigned task, these tasks may be able to be taken up by another nearby platform agent.
3) Any other visible tasks, up to the assignment limit.

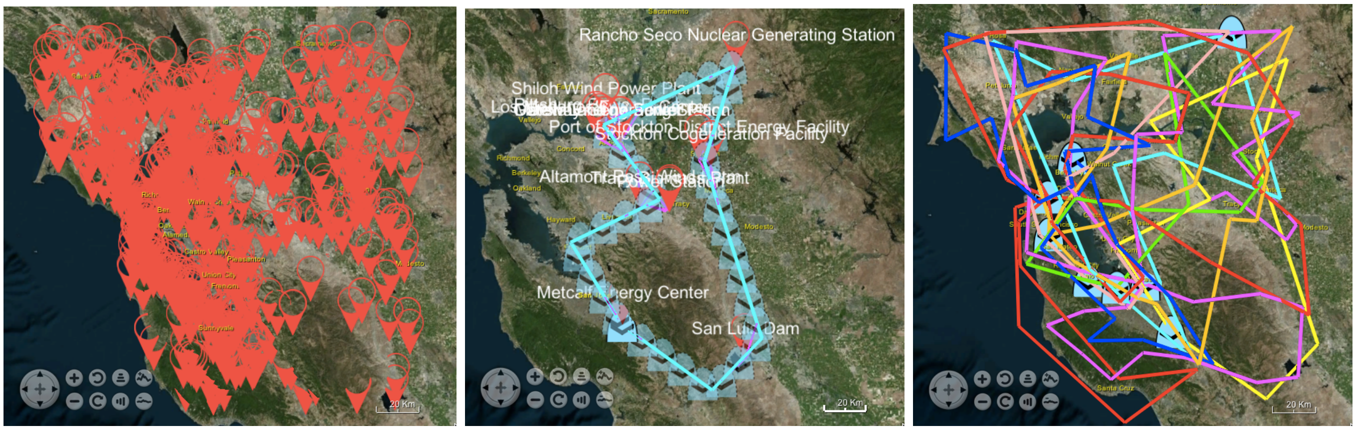
The algorithm is executed until it converges or until a set number of rounds or timeout have elapsed, at which point any changes to the set of task assignments is dispatched to platform clients to allow them to update their execution plans. When the situation is updated, the algorithm executes again.

Note that while better algorithms are clearly possible and this algorithm can be enhanced in many ways (e.g., by including task priority information and accounting for variable expected execution time for tasks), this algorithm is fast, simple, and already enables a vast improvement over the common current case of highly underutilized airborne sensors (as we demonstrate in Section IV-C), and therefore serves well enough for a first implementation of MTIP.

IV. Experimental Validation

In order to validate the MTIP sensor-sharing concept and to test our system implementation, we created a simulation of a disaster response scenario using publicly available GIS data about critical infrastructure. Evaluating this scenario, we find that there is a high potential for sensor sharing between different classes of infrastructure, providing evidence in support of the general sensor-sharing concept. We further find that our MTIP architecture implementation produces fast and effective task allocations in this scenario, as expected.

[2]Note that a great variety of potential alternative algorithms exist; here we do not make any comparison with alternatives, but assert only that this simple algorithm is sufficient to validate the MTIP approach.



(a) All infrastructure locations

(b) Power infrastructure and UAV route

(c) All UAV routes

Fig. 6. (a) Set of all 666 critical infrastructure survey targets in our scenario, indicated by standard emergency management points of interest icons or polygonal areas. (b) UAV route example, of the single UAV route planned for surveying the 14 critical infrastructure power plants, with the base position indicated by a standard airborne asset icon and projected positions indicated by translucent icons. (c) Set of all 14 UAV routes, with each color indicating a different infrastructure class and air asset icons indicating class bases.

Infrastructure Class	# Objects	# UAVs	UAV Base (Lat/Lon)
Airports	25	2	37.625°, -122.383°
Cell phone towers	251	3	37.418°, -121.883°
Dams	152	2	37.941°, -122.261°
Fire Departments	160	3	37.779°, -122.390°
Heliports	28	1	38.466°, -121.423°
Hospitals	28	1	37.432°, -122.178°
Military Installations	8	1	37.404°, -122.028°
Power Plants	14	1	37.219°, -121.747°
Total	666	14	

Fig. 5. The San Francisco disaster response scenario considers eight classes of critical infrastructure: for each class of critical infrastructure, we use a publicly available GIS data set for survey targets and specify a manually-planned set of UAV routes to provide coverage of all of the targets in that class.

A. San Francisco Disaster Response Scenario

For evaluation of MTIP, we consider a scenario in which a major earthquake has just hit the San Francisco Bay area and a number of different disaster response teams are attempting to assess damage with UAVs, prefatory to preparing a response.

In particular, we consider eight classes of critical infrastructure, as listed in Figure 5. Given typical organization structures and the specialization needed to deal with managing different classes of infrastructure, it is reasonable to assume that each class of critical infrastructure will have its own dedicated response team or teams. Thus, for instance, a disaster response team evaluating the integrity of dams will likely be different personnel in a different organization than one that is attempting to ensure that the wireless infrastructure on cell towers is operational.

For our scenario, survey targets for each class of infrastructure are populated from publicly available GIS data sets, filtered to consider only locations in the range latitude 37.0° to 38.5° and longitude -123.0° to -121.0° (roughly, North/South from Santa Cruz to Sonoma County and inland through the Sacramento and San Joaquin Valleys). The set of all 666 critical infrastructure survey targets is shown in Figure 6(a).

For each class of critical infrastructure, we assumed one of the survey targets is the operating base for a disaster response team and manually planned a set survey routes for 1-3 UAVs from that base to cover the rest of the survey targets in the class, with the number of routes depending on the number and geographical dispersion of targets to be surveyed. Manual planning was carried out for each infrastructure class independently, on a map showing only survey targets and UAV routes for that infrastructure class. The preferred altitude planned for surveys is 500 meters, but higher over the various coastal mountain ranges to a maximum of 1500 meters. Figure 6(b) shows an example, the single UAV route planned for surveying the 14 critical infrastructure power plants, while Figure 6(c) shows the set of all 14 UAV routes, with each color indicating a different infrastructure class. Hereafter, when we refer to a "set of UAVs" it means the group of UAVs associated with a particular infrastructure class.

For survey UAVs, we chose to consider the Boeing ScanEagle, a small high-endurance UAV with an approximately 3-meter wingspan and 20 kg mass, used by a number of military and civilian organizations around the world. Based on its published specifications, we assume a flight speed of 40 m/s, a 6000 meter operating ceiling, and up to 24 hours endurance (though the planned survey routes of our scenario are all less than four hours). Standard equipment options for ScanEagles includes a high-resolution imager with up 170x zoom, so we also assume an effective visual survey range of up to 20 km for initial damage assessment.

For planning purposes the survey routes are quantized into projections at intervals of 5 minutes between projections (i.e., for UAV locations 12 km apart along the planned route). Finally, each UAV is assumed to be able to adequately survey three targets per minute, thus implying a maximum of 15 survey targets per planning location.

All told, the set of planned survey routes spans 454 planning locations, with the average mission planned to survey approximately 50 targets over a period of approximately two and a half hours. Even for the densest planned route, surveying dams, the mean number of targets per planning location is only 2.81, meaning that under these conditions most time is indeed spent in transit rather than on mission execution, and that there is ample opportunity for sensor sharing. This is, of course, dependent on our assumptions, and slower surveys or faster UAVs will reduce the opportunity for sharing; given the use of real GIS data and UAV specifications, however, we believe that it is reasonable to expect qualitatively similar sparseness to be the case for many real-world deployments.

B. Potential for Sensor Sharing

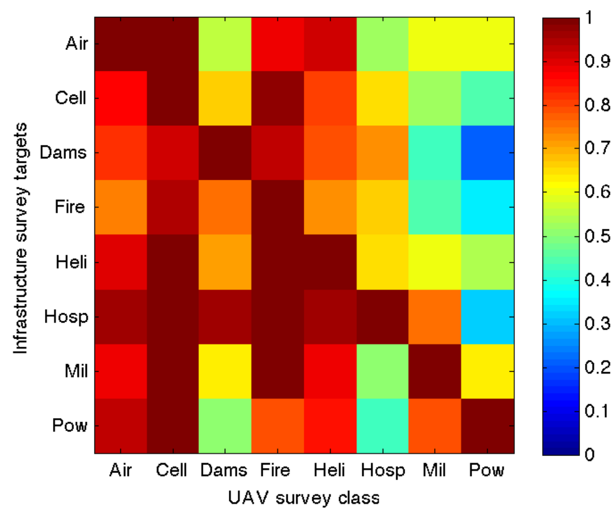
We begin by assessing the theoretical potential for sensor sharing across different organizations, regardless of our particular implementation of this capability in the MTIP platform. The key hypothesis of MTIP is that commonalities in the physical and human geographical environment are likely to mean that an aerial platform route planned for one set of sensor tasks will likely pass close to other classes of sensor tasks as well. Critical infrastructure, for example, is not placed arbitrarily, but tends to be highly constrained by the distribution of population and major physical features such as rivers and mountains. In our scenario, we would thus expect that any given set of UAVs is likely to have good visibility on many of the survey targets for other classes of infrastructure, despite being planned independently. This is particularly likely to be true in the case of UAV routes that have been planned for the more numerous and pervasive classes of infrastructure: Figure 7(a), for example, shows that between them, the three UAV routes planned for covering cell phone towers also cover all hospitals and power plants.

To evaluate this hypothesis, we ran two experiments in which we ran the MTIP system with no constraints on the number of targets that could be covered for a given route segment. In this configuration, every survey target is assigned to every route segment from which it is visible and within sensor range, thereby showing the maximum potential for sensor sharing.

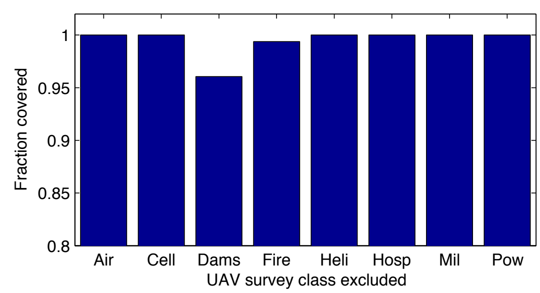
For the first test, we ran the planned UAV routes for each class of infrastructure individually against the survey targets of each class of infrastructure. Figure 7(b) shows the fraction of survey targets in each infrastructure class that can be observed for each of the 64 pairings. On the diagonal, of course, every set of UAVs perfectly covers the infrastructure class for which its routes were planned. UAVs planned to cover highly dispersed classes of infrastructure, such as cell phone towers and fire departments, cover most other infrastructure classes quite well. Even the most small and geographically constrained classes of infrastructure, however, such as power plants and military installations, turn out to provide coverage of a fairly high proportion of most other classes of infrastructure.

Complementarily, for the second test we ran a "leave one out" test in which we ran the survey targets for one infrastructure against the set of planned UAV routes for all other classes of infrastructure, showing how well an infrastructure

(a)

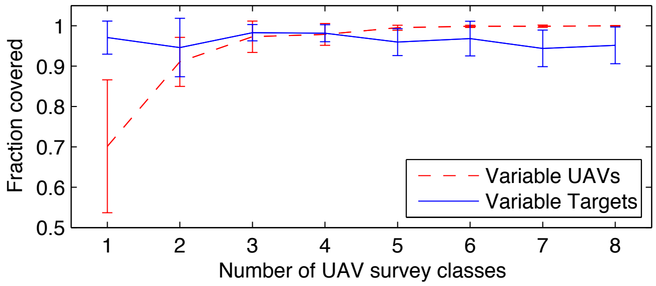


(b)

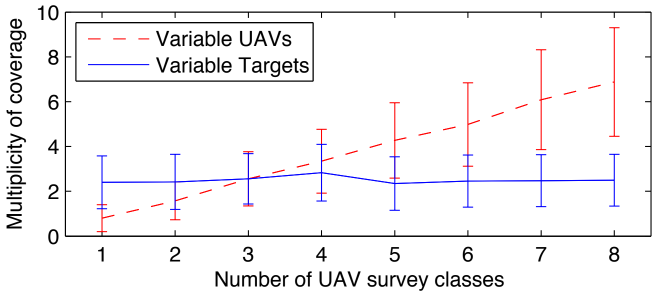


(c)

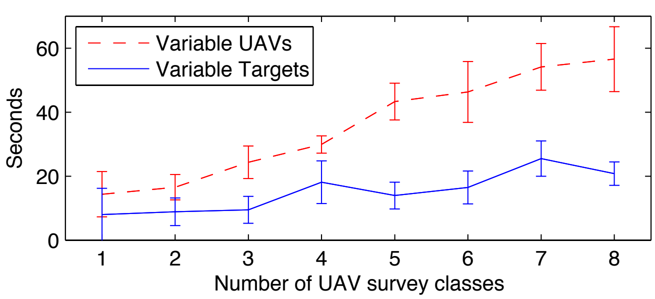
Fig. 7. A shared geographical environment makes it likely that UAVs planned for one task will provide good coverage for others as well: (a) for example, UAVs tasked to survey cell towers also cover all hospitals and power plants. Unconstrained task assignment shows sensor sharing potential (b) for individual sets of UAVs and classes of survey targets, and (c) for a class of targets to be covered by some UAV from any other class.



(a) Fraction of targets covered



(b) UAVs per target



(c) MTIP execution time

Fig. 8. Performance of MTIP sensor sharing with respect to variable numbers of UAVs (u) or survey targets (t): (a) even with few UAVs and many targets, the vast majority of targets are covered by at least one UAV, (b) most targets receive coverage from multiple UAVs, and (c) the time required for MTIP to make and dispatch allocation plans scales approximately linearly with both number of UAVs and number of targets. All graphs show mean over 10 trials ± 1 standard deviation.

class might potentially be covered by relying entirely on sensor sharing from the UAVs of other teams. Figure 7(c) shows the fraction of survey targets in each infrastructure class that can be observed by the UAV routes planned for the other classes. Here, every class of survey targets is better than 95% covered, and in fact only two are not 100% covered, the only exceptions being a set of six small dams in isolated valleys in the Point Reyes National Seashore and a single volunteer fire department in a small town in a rural zone at the Eastern edge of the area under consideration.

These experiments thus indicate that the MTIP sensor sharing concept is indeed likely to be valid and of significant use in real-world tasks such as disaster response, in which there is significant geographical correlation in the likely interests of different sensor users.

C. *Efficacy of MTIP Sensor Sharing*

Having established that most potential targets have the potential to be observed by most sets of UAVs, we now evaluate the efficacy of our MTIP implementation in allocating survey targets to UAVs with constrained time resources. For these experiments, we run the full MTIP system in an emulated network environment, in which each UAV is represented by a dummy process implementing a simplified version of a Marti platform client, and with the previously stated limit of 15 survey targets per UAV planning location.

We evaluate both the efficacy and scalability of the system through two tests with randomly selected subsets of the system. For the first test, we randomly select u sets of UAVs and execute MTIP to plan for surveying all eight sets of critical infrastructure, ranging u from 1 to 8 and running 10 trials for each condition. The second test fixes the number of randomly selected sets of UAVs to $u = 3$ and executes MTIP to plan for surveying t randomly selected sets of survey targets, ranging t from 1 to 8 and running 10 trials for each condition.

Figure 8 summarizes the results of these two experiments. Figure 8(a) shows that MTIP effectively implements sensor sharing even with many consumers and few airborne resources: just two sets of UAVs is enough to reliably cover the vast majority of survey targets and 5 UAV sets are enough to reliably cover all but a few particularly difficult to see survey targets, Complementarily, with a fixed number of sets of UAVs, performance degrades by only a small amount as the most densely populated areas begin to saturate UAV sensor sharing capacity. Indeed, as Figure 8(b) shows, there is enough excess sensor capacity to allow most targets to be surveyed by multiple UAVs. This also points out a more subtle capability of sensor sharing enabled by MTIP: even for those groups that are operating their own UAVs, mission resilience can be improved by "backing up" their UAVs with sensor sharing to take advantage of spare sensor capacity on other UAVs that fly nearby routes.

Finally, Figure 8(c) shows the time required for MTIP to make and dispatch allocation plans for these complex scenarios on a portable COTS machine (MacBook Air with 1.7 GHz Intel Core i7 and 8 GB RAM). The time scales approximately linearly with both number of UAVs and number of targets, remaining quite reasonable even for cases comprising more than 1000 interacting agents. Note also that in the current prototype implementation, a significant majority of this time is spent in computation of terrain intersections via a fairly simple and unoptimized method and in dispatching thousands of assignments of survey targets to UAVs using a separate HTTP session for each assignment. There is thus much room for improvement to enable even larger scale sensor sharing.

V. Contributions

As we have demonstrated, there are significant opportunities to improve the overall efficacy of airborne sensor platforms by making their sensors available for opportunistic use by other information consumers. Our prototype MTIP system provides an implementation of such a framework, in the context of a publish-subscribe IMS, and we have validated both this system and the overall sensor sharing concept using an emulated scenario of disaster response in the San Francisco area. While the present work tests only this region, the region tested is highly heterogeneous, including the flat rural lands of the Sacramento and San Joaquin Valleys, extremely high population density urban areas near San Francisco, and largely unpopulated regions of high topographic relief. Furthermore, spot inspection of some infrastructure classes in other areas indicates the observed patterns of overlap are likely to hold elsewhere as well.

Directions for future work include improvement of the agent-based allocation algorithm (e.g., to make use of time constraints and additional information about sensors and task requirements), investigation of the dynamics of replanning and response to emergent events, and leveraging the agent-based allocation architecture to further decentralize MTIP and allow nearby platforms to communicate and replan even when their communication with the central dispatcher is limited or unavailable. There are also a number of opportunities for improving efficiency and scalability in the current prototype. It may also be of interest to consider adaptation of the MTIP approach to other architectures, such as incentive driven sensing or non-publish-subscribe information management systems. Finally, there are a number of improvements that can be made in the MTIP prototype in order to move it toward deployment, such as elaboration of the priority metrics and taking sensor task complexity into account in planning.

ACKNOWLEDGMENT

This work has been supported by the United States Air Force under Contract No. FA8750-14-C-0096. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views, opinions, and/or findings contained in this article are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for public release; distribution is unlimited.

REFERENCES

[1] M. Gillen, J. P. Loyall, and J. Sterling, "Dynamic quality of service management for multicast tactical communications," in *Proc. of the 14th IEEE Computer Society Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC)*, Mar. 2011.

[2] M. Gillen, J. Loyall, K. Usbeck, K. Hanlon, A. Scally, J. Sterling, R. Newkirk, and R. Kohler, "Beyond line-of-sight information dissemination for force protection," in *Proc. of the Military Communications Conference (MILCOM)*, Orlando, Florida, October 29-November 1 2012.

[3] OMG Data Distribution Service Portal, `http://portals.omg.org/dds/`, Retrieved March 6, 2012.

[4] M. Xiong, J. Parsons, J. Edmondson, H. Nguyen, and D. Schmidt, "Evaluating technologies for tactical information management in net-centric systems," in *Defense and Security Symposium*. International Society for Optics and Photonics, 2007, pp. 65 780A–65 780A.

[5] M. Carvalho, A. Granados, K. Usbeck, J. Loyall, M. Gillen, A. Sinclair, and J. Hanna, "Integrated information and network management for end-to-end quality of service." in *Proceedings of MILCOM*, 2011.

[6] K. Usbeck, M. Gillen, J. Loyall, A. Gronosky, J. Sterling, R. Kohler, K. Hanlon, A. Scally, R. Newkirk, and D. Canestrare, "improving situation awareness with the Android Team Awareness Kit (ATAK)," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2015, pp. 94 560R–94 560R.

[7] D. S. Rosenblum and A. L. Wolf, *A design framework for Internet-scale event observation and notification*. ACM, 1997, vol. 22, no. 6.

[8] C. Ma and J. Bacon, "Cobea: A corba-based event architecture," in *Proceedings of the 4th conference on USENIX Conference on Object-Oriented Technologies and Systems-Volume 4*, 1998, pp. 9–9.

[9] B. Essendorfer and W. Mueller, "Interoperable sharing of data with the Coalition Shared Data (CSD) server," *North Atlantic Treaty Organization (NATO)/Research and Technology Organization (RTO): C3I in crisis, emergency and consequence management*, 2009.

[10] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[11] K. A. Delin, "The sensor web: A macro-instrument for coordinated sensing," *Sensors*, vol. 2, no. 7, pp. 270–285, 2002.

[12] R. A. Burne, A. L. Buczak, V. R. Jamalabad, I. Kadar, and E. R. Eadan, "Self-organizing cooperative sensor network for remote surveillance," in *Enabling Technologies for Law Enforcement and Security*. International Society for Optics and Photonics, 1999, pp. 124–134.

[13] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni, "Sensorflock: An airborne wireless sensor network of micro-air vehicles," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, ser. SenSys '07. New York, NY, USA: ACM, 2007, pp. 117–129. [Online]. Available: http://doi.acm.org/10.1145/1322263.1322275

[14] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, "Socio-economic vision graph generation and handover in distributed smart camera networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 2, p. 20, 2014.

[15] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: opportunistic or participatory?" in *9th Workshop on Mobile Computing Sys. and App.*, 2008, pp. 11–16.

[16] A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing," in *7th Int'l Conf. on Information Processing in Sensor Networks*. IEEE Computer Society, 2008, pp. 481–492.

[17] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *Communications Magazine, IEEE*, vol. 52, no. 8, pp. 29–35, 2014.

[18] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "Anonysense: A system for anonymous opportunistic sensing," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 16–30, 2011.

[19] S. B. Eisenman, N. D. Lane, and A. T. Campbell, "Techniques for improving opportunistic sensor networking performance," in *Distributed Computing in Sensor Systems*. Springer, 2008, pp. 157–175.

[20] X. Sheng, X. Xiao, J. Tang, and G. Xue, "Sensing as a service: A cloud computing system for mobile phone sensing," in *Sensors, 2012 IEEE*. IEEE, 2012, pp. 1–4.

[21] P. P. Jayaraman, C. Perera, D. Georgakopoulos, and A. Zaslavsky, "Efficient opportunistic sensing using mobile collaborative platform mosden," in *Int'l Conf. on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 2013, pp. 77–86.

[22] Q. Liang, X. Cheng, S. Huang, and D. Chen, "Opportunistic sensing in wireless sensor networks: theory and application," *Computers, IEEE Transactions on*, vol. 63, no. 8, pp. 2002–2010, 2014.

[23] J. Loyall, M. Gillen, J. Cleveland, K. Usbeck, J. Sterling, R. Newkirk, and R. Kohler, "Information ubiquity in austere locations," *Procedia Computer Science*, vol. 10, no. 0, pp. 170 – 178, 2012, aNT 2012.

[24] H. V. D. Parunak and S. Brueckner, "Concurrent modeling of alternative worlds with polyagents," in *Multi-Agent-Based Simulation VII*. Springer, 2007, pp. 128–141.

[25] D. G. Bel, F. Kuehne, C. Maxwel, R. Kim, K. Kasraie, T. Gaskins, P. Hogan, and J. Coughlan, "NASA World Wind: Opensource GIS for mission operations," in *IEEE Aerospace Conference*, 2007, pp. 1–9.

[26] D. Pianini, M. Viroli, and J. Beal, "Protelis: Practical aggregate programming," in *ACM Symposium on Applied Computing 2015*, April 2015.

[27] J. Beal, D. Pianini, and M. Viroli, "Aggregate programming for the Internet of Things," *IEEE Computer*, vol. 48, no. 9, 2015.

[28] M. Viroli, F. Damiani, and J. Beal, "A calculus of computational fields," in *Advances in Service-Oriented and Cloud Computing*, ser. Communications in Computer and Information Sci., C. Canal and M. Villari, Eds. Springer Berlin Heidelberg, 2013, vol. 393, pp. 114–128.