

ADDSSEN: Adaptive Data Processing and Dissemination for Drone Swarms in Urban Sensing

Di Wu, *Member, IEEE*, Dmitri I. Arkhipov, Minyoung Kim, Carolyn L. Talcott, Amelia C. Regan, *Member, IEEE*, Julie A. McCann, *Member, IEEE*, and Nalini Venkatasubramanian, *Senior Member, IEEE*

Abstract—We present *ADDSSEN* middleware as a holistic solution for Adaptive Data processing and dissemination for Drone swarms in urban SENSing. To efficiently process sensed data in the middleware, we have proposed a cyber-physical sensing framework using partially ordered knowledge sharing for distributed knowledge management in drone swarms. A reinforcement learning dissemination strategy is implemented in the framework. *ADDSSEN* uses online learning techniques to adaptively balance the broadcast rate and knowledge loss rate periodically. The learned broadcast rate is adapted by executing state transitions during the process of online learning. A strategy function guides state transitions, incorporating a set of variables to reflect changes in link status. In addition, we design a cooperative dissemination method for the task of balancing storage and energy allocation in drone swarms. We implemented *ADDSSEN* in our cyber-physical sensing framework, and evaluation results show that it can achieve both maximal adaptive data processing and dissemination performance, presenting better results than other commonly used dissemination protocols such as periodic, uniform and neighbor protocols in both single-swarm and multi-swarm cases.

Index Terms—Drone swarms, data dissemination, urban sensing.

1 INTRODUCTION

UNMANNED Aerial Vehicles (UAV), commonly known as *drones*, are aircraft without a human pilot on board. These flights are either controlled autonomously by computers in the vehicle, or under the remote control of a pilot on the ground or in another vehicle. Drones have a wide variety of shapes, sizes, configurations, and characteristics, capable of communicating voice, video, and other data. Drones are used in many military and civil applications [1], [2], such as surveillance, policing, firefighting, and search and rescue missions. Drones can form swarms for self-organization and collaboration in airborne mobile ad hoc networks [3]. In addition to their military applications, drone swarms will become an increasingly important component of smart cities [4], [5]. Governments are now in the process of enabling these uses by adopting policies that will allow them to be safely used (*e.g.* flying below 400 feet and always within visual sight according to Federal Aviation Administration [6]). Of particular interest is their use for demand and event responsive monitoring of transportation systems and air quality. Swarms can be dispatched as a sensing grid to sense traffic related information on the public roadway, or transmit real-time reports when accidents or natural disasters impact traffic networks.

1.1 Motivations

As an important part of urban sensing, transportation monitoring uses in- and over-roadway sensors to collect traffic data [7].

- *D. Wu, and J. A. McCann are with the Department of Computing, Imperial College London, London SW7 2AZ, UK. E-mail: {d.wu,j.mccann}@imperial.ac.uk.*
- *D. I. Arkhipov, A. C. Regan, and N. Venkatasubramanian are with the Department of Computer Science, University of California, Irvine, CA 92697-3435, USA. E-mail: {darkhipo,aregan,nalini}@ics.uci.edu.*
- *M. Kim and C. L. Talcott are with the Computer Science Laboratory, SRI International, Menlo Park, CA 94025-3493, USA. E-mail: {mkim,clt}@csl.sri.com .*

Manuscript received

Examples of in-roadway sensors include inductive-loop detectors, which are saw-cut into the pavement; and tape switches, which are mounted on the roadway surface. Examples of over-roadway sensors are video image processors that use cameras mounted on tall poles adjacent to the roadway; and laser radar sensors mounted on structures that span the traffic lanes. However, the installation and maintenance costs of loop detectors and video cameras are high, and they might not properly operate at night or in severe weather conditions. These sensing techniques sometimes are unreliable and frequently malfunction due to device failure, corrosion and infrastructure deterioration. Moreover, sensor placements are generally sparse over the transportation network (mainly at arterioles and intersections), without robust coverage especially for rural area and interstate highway.

Drone swarms are an emerging technology for wide area sensing that involves over-roadway sensors mounted on aerial platforms. The idea of using sensors in mobile vehicles to efficiently and inexpensively monitor traffic network has gained quite a bit of attention in the last few years [8]. Drone swarms provide an additional way to manage these kinds of sensors as a mobile platform with various sensing capabilities (*e.g.* traffic condition, air quality, noise perception). They are particularly effective when the city management centers need an integrated sensing solution with robust coverage to areas of interest on demand.

Quadcopter UAVs such as the state-of-the-art DJI Phantom 3 [9] (product dimensions: $8 \times 13 \times 18$ inches) and the Parrot AR.Drone 2.0 [10] (product dimensions: $23 \times 0.5 \times 23$ inches) are the most successful class of commercially available drones, due to their autonomous nature, cost effectiveness and size. Inspired by the use of drones in agriculture for pesticide dispersion and large-scale field imagery for data analysis [11], we propose the deployment of quadcopters as a suitable solution for swarming drones in urban sensing.

As shown in Fig. 1, in response to real-time traffic conditions on the I-405 freeway in Southern California (one of the

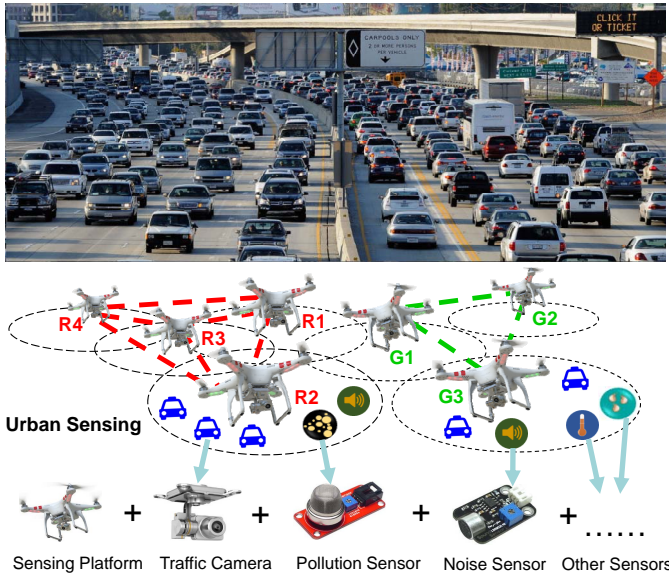


Fig. 1. Urban sensing in transportation using drone swarms.

most congested freeways in the US), two drone swarms can be dispatched for some typical urban sensing tasks over different roadway segments. One swarm is composed of 4 drones ($R1$, $R2$, $R3$ and $R4$), and one swarm is composed of 3 drones ($G1$, $G2$ and $G3$). The spatio-temporal communication links for these drones in each swarm are marked by red connections and green connections respectively. Three typical urban sensing tasks for the two drone swarms illustrated in Fig. 1 are traffic sensing, pollution sensing, and noise sensing. Each drone in each of the two swarms is outfitted with a traffic camera (e.g. sensing traffic speed, occupancy and volume), a pollution sensor (e.g. sensing ozone, nitrogen dioxide and carbon monoxide) and a noise sensor (e.g. sensing sound wavelength and intensity) on the sensing platform to execute these tasks; additional urban sensing tasks (for example temperature and moisture) can be achieved by adding corresponding sensors. Since each drone can only sense a partial area of a roadway segment along its flight path (see the black dotted circle in Fig. 1), using drone swarm can efficiently increase the coverage area by leveraging the sensing capabilities of grouped drones. The sensed data collected by these drone swarms can either be stored in the local memory of each drone as spatial-temporal records for future analysis after the drone returns to its originating depot, or immediately transmitted to the traffic management center through a cellular tower for real-time use. Therefore, drone swarms can provide robust and integrated sensing platforms for future cities [12], [13].

1.2 Our Contributions

In this paper, we propose *ADDSEN* (Adaptive Data processing and dissemination for Drone swarms in urban SENSing) as a holistic middleware solution for drone swarms executing urban sensing tasks. To efficiently process sensed data in the middleware, we have implemented a cyber-physical sensing framework using partially ordered knowledge sharing (POKS) for distributed knowledge management in drone swarms (open-source code is available at [14]). *Knowledge* is an entirely application defined concept for data processing and dissemination. For example, the noise sensing

application defines noise as one kind of knowledge in urban sensing with specific contents (sound values) as *knowledge items* to store. There a distinction between “data” and “knowledge” in our urban sensing applications. We consider “data” to be the raw bytes collected by drones from sensor hardware. After a drone collects data from their sensors this data is processed by the drone into a set of knowledge items. Knowledge items have informative headers and carry partially processed sensed data as their payload. Drones transmit, receive, and process knowledge items during the execution of their assigned tasks. While *ADDSEN* processes and disseminates sensed data, knowledge items containing partially processed data are the containers which are sent and received by drones during data processing and dissemination. The framework provides a generic service to represent, manipulate, and share knowledge across aerial disruption-tolerant networking (DTN) [15] under minimal assumptions about connectivity. To overcome disconnection in DTN, knowledge items are stored for an extended time and grouped as DTN *bundles* for dissemination. This networked cyber-physical system framework is the foundation for distributed urban sensing applications.

To better adapt to the link status for intra/inter-swarm data dissemination, a reinforcement learning based dissemination strategy is implemented in our cyber-physical sensing framework, where the wireless link status is estimated by knowledge exchange, and broadcast rate is determined by our online learning solution. Our use of reinforcement learning is similar to that of [16]. In our solution, *ADDSEN* uses online reinforcement learning techniques to adaptively balance the broadcast rate and loss rate periodically, based on a strategy function incorporating a set of variables to reflect the change in link status. The strategy function can guide sequential state transitions to approach the best broadcast rate, so that each drone can reduce the loss rate of data dissemination.

Periodic intra/inter-swarm broadcast dissemination also consumes the storage and energy of each drone. To maximize the in-field time of drone swarms as sensing grids, an optimal balancing method is designed in *ADDSEN* for the cooperative task of balancing storage and energy allocation. This optimization program can assign data dissemination and sharing tasks to drones with more residual storage and energy; it also regulates the reception behavior of drones after detecting a broadcast, to avoid reception by drones with limited capacity.

The rest of this paper is organized as follows. Section 2 addresses the challenges and related works in urban sensing using drone swarms. Section 3 introduces the distributed knowledge management in *ADDSEN*. Section 4 presents the data processing and dissemination strategies in *ADDSEN*. Section 5 evaluates the performance of *ADDSEN*. Finally, Section 6 concludes the paper.

2 CHALLENGES AND RELATED WORKS

Drones depart their depot as a swarm with assigned tasks, and they are expected to be out in the field together for a period of time before returning to the depot. To support efficient urban sensing in a scenario described in Fig. 1, several challenges must be addressed:

- 1) Data processing framework: each drone can only sense the area along its flight path (see the black dotted circle in Fig. 1); the drone swarm needs a way to aggregate the sensed data from individual drones into a sensing grid. Meanwhile, the same area may be sensed by multiple drones, resulting in

redundant information (see the overlapped areas of sensing circles in Fig. 1). Therefore, in-network data processing is needed to analyze the quality of sensed data so that low-quality/faulty and redundant data can be discarded.

Haghighi et al [17] has proposed a stochastic time-domain model for burst data aggregation in IEEE 802.15.4 wireless sensor networks. In-network data aggregation and processing have also been addressed in the case of VANETs in [18]. However, none of these works have proposed a feasible scheme for data aggregation and redundancy processing in a sensing swarm. In our early work [19], we implemented a networked cyber-physical system (NCPS) using a partial ordering framework to handle data aggregation and analysis, but its extension to support urban sensing applications is not well studied.

- 2) Data dissemination strategy: To reduce the delay introduced by real-time aggregation of data sensed from working drones, the data can first be sent back to the depot via cellular tower and then aggregated as needed. However, cellular service is not always available and frequent transmissions by cellular connections may quickly drain a drone's battery. Data dissemination between drones through WiFi Direct or Bluetooth Low Energy [20] to relay the data to the depot is a feasible solution. Data dissemination for drone swarms can be intra-swarm (*i.e.* in the same swarm) or inter-swarm (*i.e.* between different swarms); data is usually disseminated under disruption-tolerant networking (DTN) due to the changing distances between drones or swarms. In comparison with unicast, anycast and multicast, broadcast can better support vehicular dissemination due to its inherent data redundancy and complete transmission coverage [21].

The usual flooding approach results in the broadcast storm problem which wastes valuable and limited bandwidth and energy. Probabilistic approaches can optimize flooding, such as the OAPB/DB [22] protocol that rebroadcasts data by considering the geographical zone area and local vehicle density within two-hops of distance, and the REAR protocol [23] that relays broadcast data based on their estimated message received value. Low latency broadcast scheduling in duty-cycled multi-hop wireless networks has been investigated in [24]. However, under highly mobile connections and harsh communication environments, drones require fast, link-adaptive, and low-loss broadcast strategies [25].

- 3) Resource utilization: When drones in a swarm are sent out to execute tasks together in the field, they are expected to maximize their in-field time. However, drones are resource (*e.g.* memory, energy and others) restricted aerial vehicles. A small drone, such as the DJI Phantom 3, can fly for around 25 minutes on a full charge. The much larger InView Unmanned Aircraft System [26] has a flight duration for 7 hours, but is not suitable for large-scale swarming due to its cost. Periodic dissemination with broadcasting and communication with strong interference cause excessive energy consumption and impacts a drone's in-field time. The memory of drone is also limited because sensing tasks can generate a large volume of data.

To achieve efficient usage of mobile resources, mobile data gathering problem has been formulated as a network utility maximization problem which is constrained by flow, energy balance, link and battery capacity of the mobile collector in [27]. A game theory based data collection method was

also proposed to maximize the energy efficiency with fairness constraints in [28]. To cooperatively schedule resource, a self-adaptive strategy has been proposed for the evolution of cooperation in distributed networks in [29], and cost-aware cooperative resource provisioning for heterogeneous workloads has been studied in [30]. To increase the in-field time of a drone swarm as a sensing grid that collects heterogeneous data, we have to design an adaptive optimization method to cooperatively balance the memory and energy capacities in drone swarms so that the workload of heavy-duty drones can be delegated to others with more resources.

We will present the specific designs of *ADDSEN* to address above challenges in following sections.

3 DISTRIBUTED KNOWLEDGE MANAGEMENT

Knowledge is semantically meaningful information that can be generated, stored, processed, aggregated, and communicated. The distributed knowledge management in *ADDSEN* maintains a distributed knowledge base, which can be accessed by other system components via a common interface, for data processing and dissemination in networked and distributed cyber-physical environments. To this end, each node maintains a local knowledge base to store knowledge-related application content (knowledge items), and executes data dissemination protocols to exchange knowledge items with its neighbors. The two fundamental distributed knowledge management and dissemination schemes in *ADDSEN* middleware are partial ordering for knowledge sharing, and a cyber-physical sensing framework to schedule system components for various urban sensing applications.

3.1 Partially Ordered Knowledge Sharing

Partially ordered knowledge sharing (POKS) is organized asynchronously and can make explicit use of distributed computations in space and time. Each node's local clock increases monotonically by at least one unit in each instruction and is loosely synchronized with other nodes in the network whenever admitted by the networking conditions. Our implementation of time synchronization satisfies Lamport's axioms of logical time.

Each node uses some of its storage as a cache, which we refer to as a *knowledge base*. The local knowledge base is a set of *knowledge items*, and each knowledge item has at least the following attributes associated with it: the *creator*, *i.e.* the node that created it; its *creation time* and its *expiration time*. The additional attributes of a knowledge item depend on its application content related *knowledge classes*, namely parameters, status, and statistics. Knowledge based network caching allows the system to support communication even if no end-to-end path exists at a single point in time. In contrast to a shared-memory model, POKS allows each node to have its own (typically partial and delayed) view of the distributed state of knowledge. Unlike a message-passing model, knowledge in POKS is not directed toward a particular destination. Instead, each node decides based on the knowledge attribute if it wants to use the unit of knowledge.

Specifically, we define an equivalence relation \equiv and a partial ordering \prec on knowledge items. Intuitively, the equivalence $k \equiv k'$ means that k and k' contain the same information, and the relation $k \prec k'$ means that k' replaces k . A typical case in the relation $k \prec k'$ happens when k is obsolete given that k' is fresher than k . For the operation of the knowledge manager,

$k \prec k'$ implies that if k is in the local knowledge base and k' is received then k should be replaced by k' . Similarly, $k \equiv k'$ implies that if k is in the local knowledge base and k' is received then k' is redundant and can be ignored. In this paper, we use the notation of $k(c_k, ct_k, et_k, \dots)$ to represent knowledge k with creator c_k , creation time ct_k , expiration times et_k and application-specific attributes (if any). The ordering is specified by statements of the form $O: k(\dots) \prec k(\dots)$ if *condition*. For example, $O: k(c_k, ct_k, \dots) \prec k(c_k, ct'_k, \dots)$ if $ct_k < ct'_k$ represents an ordering based on creation time (in this case, fresher knowledge replaces older one) between knowledge items that are generated by same creator c_k .

POKS is of key importance for scalable implementations, because its in-network replacement of inferior knowledge decreases the communication overhead by discarding information in a semantically meaningful way and limiting the amount of knowledge that needs to be stored at each node.

3.2 Cyber-Physical Sensing Framework

The cyber-physical sensing framework in *ADDSSEN* middleware provides network caching and in-network processing mechanisms to implement distributed sensing, storage, computing and communication based on POKS. The framework consists of hosts, engines, applications, devices, storage, etc. In *ADDSSEN*, a *host* and an *engine* correspond to a specific drone and a process on which applications are running, respectively. Urban sensing application related information can be obtained from physical devices including sensors, actuators and others. This information is kept in local storage. Each drone uses some of its storage as a network cache and enables opportunistic knowledge sharing even in a disruptive environment.

Fig. 2 describes the operation and interaction of drones within the framework. Each drone is running an engine which is in turn hosting several transportation related sensing applications. Applications interact with each other and with the engine through the event-handling component of the engine. Drone applications may also register for exclusive access to drone hardware resources (such as sensors), which is handled by the scheduling component of the engine. In this framework, all local computations are event-based. There are two types of events: timed events, which can be posted to be activated at any time in Fig. 2 to schedule local action, and knowledge events, which can be posted to disseminate knowledge in the network and to respond upon after receiving new knowledge units. The event-handler allows applications to register actions that are triggered upon event detection, and also to allow for the triggering of such events. The event-handler serves as an interface to both the local storage for each drone, and system-wide distributed POKS.

Distributed knowledge sharing among 3 drones, as shown in Fig. 2, can be achieved by the data dissemination application. The data dissemination component in the framework implements specific strategies (e.g. broadcasting, load balancing) to propagate knowledge on top of the underlying physical network layer. Each knowledge base can serve as a distributed network cache to enable opportunistic knowledge sharing with intermittent connectivity so that senders and receivers are completely decoupled. The data dissemination protocols make essential use of the partial order defined on knowledge, by replacing and hence discarding a unit of knowledge whenever a unit that is higher in the ordering is received. We discuss the design of data dissemination protocols for *ADDSSEN* in Section 4.

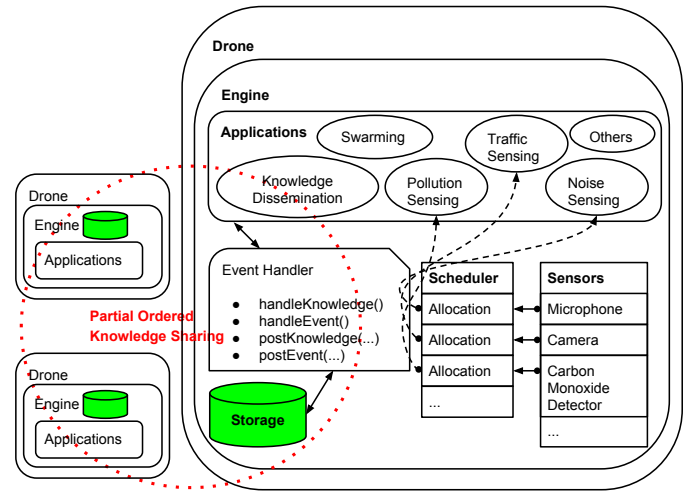


Fig. 2. Cyber-physical sensing framework and partially ordered knowledge sharing in distributed knowledge management.

4 DATA PROCESSING AND DISSEMINATION IN DRONE SWARMS

The implementation of data processing and dissemination functionality in *ADDSSEN* is based on our distributed knowledge management framework for drone swarms. This includes an online learning scheme for adaptive dissemination, a cooperative dissemination method for optimal balancing of storage and energy, and several POKS rules for various urban sensing tasks.

4.1 Swarming Drones

We propose an urban sensing scheme allowing independent depots to deploy drones into control zones for which each depot is responsible. Drones launched from a depot may form one or more swarms, and will be tasked (and programmed) by the depot to perform urban sensing tasks. Each depot will upload the appropriate applications as described in Fig. 2 to the drones to enable their sensing and communication capabilities in that depot's control domain. The programmed drone will perform its assigned urban sensing tasks around the way-points defined by the depot for its corresponding swarm. The urban sensing data collected by the drone can be either returned to its depot after the entire swarm returns to the depot, or disseminated to drones in other swarms to relay the information back to the depot (the latter is done when the swarm has other tasks to perform at the time the data is to be sent). The depot will process the returned data during execution of assigned tasks, and incorporate this information into a decision making processes which may result in a new swarm task assignment.

In *ADDSSEN* a swarm of drones performs a distributed surveillance mission by flying to sensing locations in a formation that creates an effective sensing grid to enable distributed and cooperative execution. Drones from the same swarm or different swarms can communicate with each other and assist others to accomplish tasks. To achieve our distributed sensing mission by a swarm formation, *ADDSSEN* leverages the concept of virtual potential field [31] for swarming drones. In a nutshell, virtual potential can be seen as a specification of the desired state of a system, where each member is driven by the desire to minimize its perception of, and hence its own contribution to the virtual

potential. The potential field in *ADDSEN* is designed to guide drones positioned at desired pairwise distances and to be adaptive when a new target location is selected. In the simplest case, the potential field can be a constant around the center of the desired location and increase when the distance from the center grows. To cope with local minimum, we use a distributed version of simulated annealing [32] instead of constructing the potential field using gradient descent.

In our prototype, the potential field is implemented using a potential function for the formation of the drone swarm. We define the potential p capturing the quality of a given swarm configuration as a weighted sum, $p = w_f p_f + w_l p_l$, where p_f and p_l are potentials corresponding to formation and desired location, respectively. The formation potential p_f should be minimal when a swarm creates a hexagonal lattice for distributed sensing grids. We define the formation potential of a swarm with n drones as $p_f = \frac{1}{n} \sum_{i=1}^n p_f^i$, where

$$p_f^i = \begin{cases} (Z - z_i), & \text{if } z_i < z_c \\ c_r \left(\frac{1}{z_i} - \frac{1}{Z} \right), & \text{if } z_c \leq z_i \leq Z \\ c_a \left(\frac{1}{Z} - \frac{1}{z_i} \right), & \text{if } Z < z_i \end{cases} \quad (1)$$

In the above equation, Z is the desired distance between drones in the sensing zone, and z_i is the current distance between the drone solving the optimization and another drone i . If they are located at the desired distance ($z_i = Z$), then p_f^i presents the lowest potential. Otherwise, p_f^i has repulsive impact if the drones are closer than Z and attractive impact if they are farther than Z . This corresponds to repulsive and attractive forces in artificial physics, where a force law is defined between two particles [33]. In [33] the repulsive force is restricted to remain constant when the distance between two particles gets too small, and we use this threshold distance as a range constant z_c in the above equation. When the drones are further apart, p_f^i is defined to be inversely proportional to the distance z with the constants for repulsive impact c_r and for attractive impact c_a . We define c_r to ensure that the potential field is continuous. A location potential p_l is defined as a function of distance between the center of a swarm and the desired location.

4.2 Mobile Dissemination in Drone Swarms

Due to redundant data reception and a wide transmission range, broadcasting can support data dissemination even when connections are highly mobile and the communication environments are harsh. In our multi-swarm scenario, each drone collects data (e.g., traffic, pollution, noise) at work locations above the public road and disseminate it by broadcasting it to other drones in the same or in different swarms when they are in transmission range, along the flying route. Fig. 3 presents the data dissemination scenario with two swarms at the meeting point, one with 4 drones (swarm “Red”) and one with 3 drones (swarm “Green”). Drones from different swarms disseminate DTN bundles with knowledge items to neighbor drones in the transmission range. The partial ordered knowledge sharing method introduced in Section 3.1 is applied to the received knowledge items for update and forwarding purposes.

For the DTN-based broadcast, each DTN bundle includes a number of knowledge items, and the broadcast decision is made from the tuples in each broadcast knowledge item with the following information:

- *swarm ID*, which is the unique ID to represent each swarm.

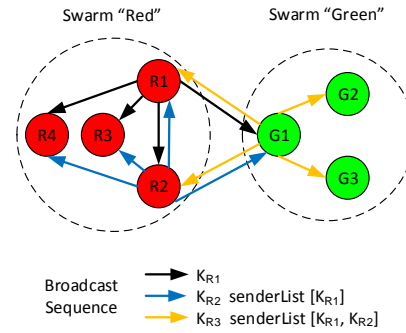


Fig. 3. Data dissemination in a multi-swarm scenario.

- *drone ID*, which is the unique ID to represent the drone in each swarm.
- *aware*, which is the set of drones that, according to the local knowledge base, are already aware of this knowledge item.

For security purposes, to protect the drone ID and local knowledge base, if a drone from a swarm wants to disseminate data to drones from another swarm, it first needs to check the swarm ID of another swarm to make sure the receiving swarm has been authorized to receive knowledge items from the sending swarm. If the swarm ID from the receiving swarm passes the verification test, then the drone in the sending swarm can obtain ID information of drones in the receiving swarm and disseminate data to these drones.

Our broadcast scheme uses meta-information in knowledge items to reduce the number of unnecessary retransmissions. To this end, we keep an additional awareness attribute for each knowledge item, which is the set of drones that, according to local knowledge, are already aware of this item. If a knowledge item k' is received and there is an existing knowledge item k in the local knowledge base such that $k \equiv k'$, then k will not be replaced by k' but the awareness set of k will be extended by the awareness set of k' . In this way awareness propagates through the network even if no fresh knowledge is generated. The awareness set is used to eliminate redundant transmissions of knowledge items. A knowledge item is only transmitted if the receiving neighbor is not already in its awareness set.

In addition, each drone locally maintains four sets of information for broadcast decisions as follows:

- *sentto*, which is the set of drones that the knowledge item has been sent to from the drone.
- *senderList*, which is the set of drones that send the knowledge item to the drone.
- *awareAll*, which indicates if all neighbor drones are aware of the knowledge item.
- *senttoAll*, which indicates if the drone sends the knowledge item to all neighbor drones.

awareAll and *senttoAll* are derived from the “aware” and “sentto” information recorded locally for each knowledge item.

During DTN-based broadcast, there is only one source of a knowledge item, called a creator, which is the drone that originally creates the DTN bundle and initiates its propagation. But the item can be disseminated many times by different senders.

Our broadcast scheme estimates the knowledge loss rate periodically by checking if a drone had already broadcasted a

knowledge item k before, and the drone is in the senderList of other received knowledge items k' . In Fig. 3, creator drone $R1$ generates an original knowledge item K_{R1} , and broadcasts it to neighbor drones in its transmission range, e.g. drone $R2$ from the same swarm "Red" and drone $G1$ from a different swarm "Green", so that drones $R2$ and $G1$ are in the senderList. The knowledge item K_{R1} will be received by $R2$ and $G1$, and then rebroadcasted by them, in the form of knowledge items K_{R2} and K_{G1} respectively. If drone $R1$ can receive the two items, it will know they are the same knowledge item as K_{R1} in its local knowledge base, by checking attribute information in these knowledge items (e.g. creator, creation time, and others), it can then ignore the two items, and put drone $R2$ and drone $G1$ in the aware bitset. Meanwhile, drone $R1$ can find its drone ID is in the senderList of K_{R2} and K_{G1} , so that drone $R1$ knows knowledge item K_{R1} has been successfully received by neighbor drone $R2$ and drone $G1$. Similarly, if the broadcast time of K_{G1} is later than K_{R2} , drone $R2$ also can receive K_{G1} with its ID in the senderList and then know that K_{R2} has been successfully broadcasted to neighbor drone $G1$.

If a drone broadcasts a knowledge item to neighbor drones and then cannot find its ID in the senderList of the rebroadcasted knowledge items from some neighbor drones after a period, the drone will determine the loss of the knowledge item. There are two major reasons that cause loss during broadcast:

- random change of link status (up/down) during dissemination, because of uncontrolled channel conditions,
- bit error rate (BER) during reception, because of the distance between drones and the number of neighbors around the receiver.

The loss rate information is used in *ADDSEN* to estimate the online link status and guide drones to adjust their dissemination strategy by the learning techniques discussed in Section 4.3.

4.3 Online Learning for Adaptive Broadcast

Online learning techniques are implemented in *ADDSEN* to periodically and adaptively balance broadcast rate and loss rate. The best broadcast rate is approached by executing state transitions during the process of online learning. A strategy function guides the state transitions, incorporating a set of variables to reflect changes in link status.

4.3.1 Q-Learning Techniques

Reinforcement Learning provides us with a mechanism, by which a system can learn to achieve a goal in control problems based on its experience. An agent in reinforcement learning chooses actions according to the current state of a system and the reinforcement it receives from the environment. Most reinforcement learning algorithms are based on estimating value functions, which are functions of states (or of state-action pairs) that estimate how good it is for the agent to be in a given state (or how good it is to perform a given action in a given state).

In general, the agent seeks to maximize the expected return R of going to the next state s' from state s with a given action a . The return at time t is defined as the sum of discounted rewards r the agent receives over the future:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (2)$$

where γ is a parameter, $0 \leq \gamma \leq 1$, called the discount rate. The discount rate determines the present value of future rewards: a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately.

The Bellman Optimality Equation for reinforcement learning is:

$$Q^*(s, a) = \sum_{s'} P_{ss'}^a \left\{ R_{ss'}^a + \gamma \max_{a'} Q^*(s', a') \right\}. \quad (3)$$

where $Q^*(s, a)$ denotes the expected value associated with taking action a in state s . Given any state s and action a , $P_{ss'}^a$ is the transition probability to each possible next state, s' , and $R_{ss'}^a$ is the expected return of going to the next state s' from state s with a given action a .

The optimal action-value function for Q^* in Eq. 3 can be directly approximated by following one-step Q -learning:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right], \quad (4)$$

where $\alpha \in (0, 1]$ is the learning rate, which models the rate of updating Q -values.

Q -learning technique is a model-free reinforcement learning technique that solves decision problems. By learning the environment and evaluating an action-value function (Q -value), which gives the expected reward of taking an action in a given state, the distributed learning agent is able to make a decision automatically. Q -learning can yield near-optimal policies without using many computations and without requiring a model of the environment.

Before learning has started, Q returns a fixed value, chosen by the designer. Then, in each time step the agent is given a reward (the state has changed). New values are calculated for each combination of a state s from the set of states, and action a from the set of actions. The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information. The algorithm ends when state s' is a final state. Note that for all final states s_f , $Q(s_f, a)$ is never updated and thus retains its initial value.

4.3.2 Online Adaptive Broadcasts

The online learning procedure for adaptive broadcast is called periodically in *ADDSEN* to update the data broadcast rate according to changes in link status, e.g., the loss rate of knowledge items.

There are two causes of knowledge item loss during the broadcast described in section 4.2. Changes in link status are random events; their occurrence may result in knowledge item loss. The probability of a change in link status occurring at any time is dependent on the attributes of the dissemination environment at that time. The second cause of knowledge item loss are bit errors introduced by the environment, we model the bit error rate (BER) in *ADDSEN* as related to the distance of the receiver to the transmitter and the number of neighbors around the receiver. If the calculated BER is larger than the tolerated error rate, knowledge items are lost. When knowledge item loss is detected the receiver will assign a "lost" status to this knowledge item. This "lost" status is indicated by a bitset assigned to the item.

In *ADDSEN*, we estimate the link quality with the knowledge loss rate on the sender side by using online Q -learning. After several rounds of DTN bundle transmissions within a short period, e.g. 2s, the drone collects statistical information on the communicated knowledge items, and periodically calculates the

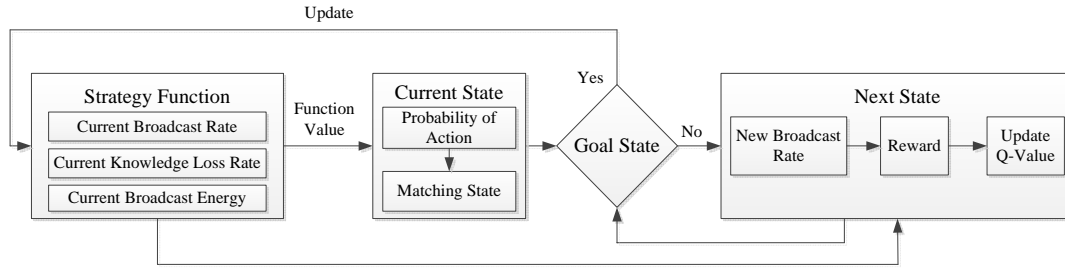


Fig. 4. Workflow of online learning in *ADDSEN*.

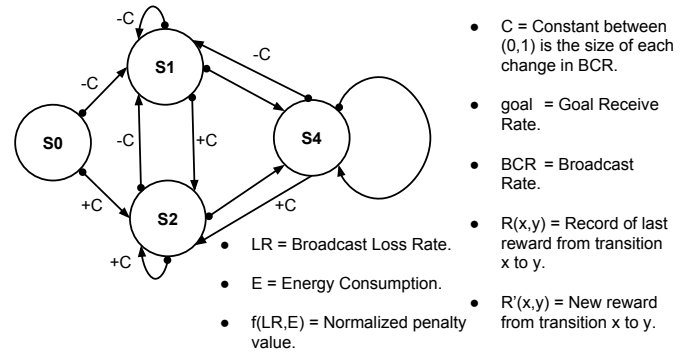
knowledge loss rate within a longer period, *e.g.* 10s, to evaluate the current broadcast strategy online. Based on these measurements, the drone uses *Q*-learning to find the best strategy for data dissemination under the current link state, by choosing a link-related broadcast rate.

The run-time of our online *Q*-learning process is illustrated in the state transition diagram in Fig. 4. The process state includes the broadcast rate, and residual energy. Each learning action represents a corresponding state transition and broadcast rate update. A high knowledge loss rate requires a high broadcast rate to guarantee the broadcast quality. However, a high broadcast rate will result in greater energy consumption. And, energy is a valuable resource that drones must carefully conserve. Therefore, we need a method to balance the need for a higher broadcast rate against high energy consumption resulting.

To approach the balanced result during learning, we define a strategy function to incorporate a set of variables that can impact the program execution and introduce a penalty value to reflect changes to these variables. The utility value in the strategy function for transition from the current state to the next state is based on the environmental conditions (*e.g.* broadcast loss rate, energy consumption). The *Q*-learning reward from the state transition is also related to the value returned by the strategy function.

The goal state in our learning process is the best achievable configuration of control variables in each drone. However, due to environmental changes such broadcast interference and energy consumption, drones need to adjust their broadcast rate and cannot always be in their goal states. The *Q*-learning algorithm in *ADDSEN* can adapt to the current link state by changing its control variables based on the knowledge loss rate derived from the current round of learning. The drone approaches its goal state through sequential transitions in the state transition diagram; the transition is made through a function of the drone's current state and the environmental variables.

Fig. 5 describes the strategy function and state transition table for the workflow of online learning in Fig. 4, where *S*₀ is the start state and *S*₄ is the goal state. In *ADDSEN* the tuple consisting of the penalty value indicated by the strategy function ($f(LR, E)$ in Fig. 5) and the ID of the current state can be used to look up the probabilities for each state transition from the current state to each possible next state. The actualized transition is determined probabilistically, based on these stored probabilities of different actions. If the current state is the goal state (*S*₄ in Fig. 5), then no state transition is made (as show in the back-edge from state *S*₄). Finally, after the state transition to the new state is made the reward is calculated based on the environmental values resulting from the transition to the new state; this in turn updates the *Q*



Current State	Next State	Cause	Effect	Probability of Action
S0	S1	Start	BCR = BCR - C	0.5
S0	S2	Start	BCR = BCR + C	0.5
S1	S1	$1 - f(LR, E) < goal$	BCR = BCR - C $R'(S1, S1) = 1 - LR$	$R(S1, S1)$
S1	S2	$1 - f(LR, E) < goal$	BCR = BCR + C $R'(S1, S2) = 1 - LR$	$R(S1, S2)$
S1	S4	$1 - f(LR, E) < goal$	BCR = BCR	1.0
S2	S2	$1 - f(LR, E) < goal$	BCR = BCR + C $R'(S2, S2) = 1 - LR$	$R(S2, S2)$
S2	S1	$1 - f(LR, E) < goal$	BCR = BCR - C $R'(S2, S1) = 1 - LR$	$R(S2, S1)$
S2	S4	$1 - f(LR, E) < goal$	BCR = BCR	1.0
S4	S1	$1 - f(LR, E) < goal$	BCR = BCR - C $R'(S4, S1) = 1 - LR$	$R(S4, S1)$
S4	S2	$1 - f(LR, E) < goal$	BCR = BCR + C $R'(S4, S2) = 1 - LR$	$R(S4, S2)$
S4	S4	$1 - f(LR, E) < goal$	BCR = BCR	1.0

Fig. 5. Illustration of state transitions in *ADDSEN*.

value.

Note that in Fig. 5, the probabilities from the start state (*S*₀) indicate an equal probability of transitioning to a lower rebroadcast rate than to a higher rate; this serves to bootstrap the system and either edge is traversed at most once. Each transition with probability 1.0 is a coerced transition; that means that the transition is deterministically triggered whenever the goal condition is met. Finally, the remaining probabilities are the previously recorded values for the normalized calculated rewards earned when using the associated state transition. Initially these values are bootstrapped to 0.5 each to indicate equal likelihood between two competing transitions from a single state.

$$\begin{aligned}
 & \text{maximize}_{t,d} \sum_t^T \sum_d^D A_{t,d} \\
 & \text{subject to} \\
 & 1) \forall_{t,d} E_{t,d} \leq P_d \vee C_{t,d} \leq M_d \iff A_{t,d} = 0 \\
 & 2) \forall_{t,d} E_{t,d} > P_d \wedge C_{t,d} > M_d \iff A_{t,d} = 1 \\
 & 3) \forall_{t,d} E_{t+1,d} = E_{t,d} - I_d - K_d \cdot B_{t,d} \\
 & \quad - K_d \cdot R_{t,d} \iff A_{t,d} = 0 \\
 & 4) \forall_{t,d} E_{t+1,d} = E_{t,d} - P_d - K_d \cdot B_{t,d} \\
 & \quad - K_d \cdot R_{t,d} \iff A_{t,d} = 1 \\
 & 5) \forall_{t,d} C_{t+1,d} = C_{t,d} - H_d \cdot B_{t,d} \\
 & \quad + H_d \cdot R_{t,d} \iff A_{t,d} = 0 \\
 & 6) \forall_{t,d} C_{t+1,d} = C_{t,d} - M_d - H_d \cdot B_{t,d} \\
 & \quad - H_d \cdot R_{t,d} \iff A_{t,d} = 1 \\
 & 7) \forall_{t,d} B_{t,d} + R_{t,d} < 2 \\
 & 8) \forall_{t,d} B_{t,d} = 1 \implies \sum_d^D R_{t,d} = 1
 \end{aligned}$$

Fig. 6. Optimization program for balancing dissemination

4.4 Balanced Dissemination in Swarms

Drones are resource restricted unmanned aerial vehicles. Their storage capacity and residual energy decrease as they complete spatial-temporal sensing tasks. Periodically intra/inter-swarm dissemination by broadcasting also reduces storage and energy of each drone. To maintain the lifetime of the sensing grid functionality of drone swarms, an optimal balancing method is needed to share and send knowledge items to drones with more residual storage and energy, while avoiding reception on drones with limited capacity. The online learning based dissemination strategy in Section 4.3.2 focuses on resource (*e.g.* energy) savings on the sender side, however the cost of resources involved in the dissemination process on the receiver side should also be considered in designing a balanced dissemination.

The optimization in Fig. 6 presents a mixed integer linear programming (MILP) for the task of balancing storage and energy allocation in drone swarms to maximize field-time, that is time spent by drones in performing useful activity (a drone in this state will be referred to as “active”). The integer program below discretizes time, memory, and energy units. Let T represent the set of discrete time intervals in each scenario. Let D be the set of drones in the scenario. Let M_d and P_d represent the number of memory units and energy units respectively expended in a single time unit by drone d , when drone d is active. Let I_d represent the number of energy units expended in a single time unit by drone d , when drone d is inactive. $B_{t,d} = 1$ if drone d is broadcasting data at time t , it is 0 otherwise. $R_{t,d} = 1$ if drone d is receiving data at time t , 0 otherwise. K_d is the per time-unit cost in energy for drone d to broadcast or receive data. H_d is the per number of memory capacity units a drone may transmit in a single time unit. $E_{t,d}$ is the residual energy of drone d at time t . This number takes into account the amount of energy needed for drone d would need to expend if it were to return to the nearest drone depot at time t . $C_{t,d}$ is the residual memory capacity of drone d at time t .

Note that $\forall_{t,d} M_d, E_{t,d}, C_{t,d}, P_d, I_d, K_d, H_d$ are problem

specific constants. Also, $\forall_d E_{0,d}, C_{0,d}$ are problems specific constants. $\forall_{t,d} B_{t,d}, R_{t,d}, A_{t,d}$ and $\forall_{t>0,d} E_{t,d}, C_{t,d}$ are variables assigned when the problems is solved. Residual energy is defined as the energy that remains after a drone has left its originating depot and traveled to its assigned work location minus the amount of energy needed for that drone to travel from the work location to an energy supply depot. This definition of residual energy supposes some pre-calculation when calculating the initial values for residual energy. For data dissemination in drone swarms, constraints 3 and 4 in Fig. 6 are energy constraints, and constraints 5 and 6 are storage constraints. Constraint 7 represents the half-duplex communication constraint on wireless networks, while constraint 8 describes the two way communication constraints. The optimization program in Fig. 6 is given in non-canonical form for clarity. We convert it into canonical form by applying some well known methods introduced in [34] to convert logical constraints to integer programming standard form. After it is in standard form the program can be input into a standard integer programming solver, such as CPLEX [35] or GUROBI [36] to derive a balanced data dissemination solution.

Our load balancing optimization is run periodically within a swarm. In each balancing period the drone with the most residual energy is selected to be the balancer. The balancer will then solve the optimized load balancing program for the new time window. The resulting solution will indicate which drones will offload data, and which will receive data. The balancer will send this solution to every drone in the swarm and select partial drones with better storage and energy capacity to disseminate and receive sensed data. The balancer can also coordinate with balancers from other swarms to run the optimization algorithm and distribute the solution to drones in different swarms to achieve cooperative balanced inter-swarm dissemination.

Note that if the remaining energy of drones in a swarm is significantly low they can be authorized by the balancer to leave the swarm and return back to the depot in advance. Meanwhile, new member drones from neighboring swarms could join the swarm to maintain its completed sensing grid. While we do not explicitly explore the case in which drones leave their original swarms and the join another swarm in the paper, there is nothing in the design of our system that prevents that. In addition, if in fact none of the drones in a swarm have sufficient residual energy to act as a balancer, then it indicates that all the drones in the swarm have residual energy below a critical threshold; therefore these drones will also not have enough energy to fulfill their urban sensing missions. In this case, all the drones in the swarm will have to return to their depot to recharge batteries.

4.5 POKS for Urban Sensing

Urban sensing requires reliable results with high-quality and trusted data. Drone swarms for urban sensing in transportation can collect various data (*e.g.* traffic, pollution, noise) through corresponding sensors. The redundant drones in swarms with the same type of sensors provide the possibility of in-network diagnosis for detecting faulty data or replacing low-quality data by comparing data quality related knowledge attributes with others through the data dissemination scheme described above. As a distributed knowledge management scheme, the partially ordered knowledge sharing (POKS) introduced in Section 3.1 can be used to analyze the quality of local knowledge items, discard faulty knowledge items, and replace low-quality knowledge items with

high-quality ones from other drones. We present the POKS rules for reliable urban sensing in transportation here.

Traffic Sensing

Traffic cameras enable drones to capture and record contiguous traffic flows at different roadway segments, and merge the resulting synchronous images to perform responsive traffic analysis; this is similar to recent deployments of drones in agriculture for large-scale field imagery [11]. In our urban sensing scenario, drones occupy a specific area (GPS location g) over a roadway segment, synchronize with each other and record contiguous traffic flow at a specific time (timestamp t). The flow information can then be returned to the traffic manager. In this scenario if any of the drones takes faulty images or video recordings during a synchronized reading all the drones in the tasked swarm will rate these recordings low in the partial ordering, and they will be overwritten by subsequent synchronous non-faulty recordings in the same area but with a different timestamp t' , as follows:

$$O_{traffic} : r(\dots, t, g, image, \dots) \prec r(\dots, t', g, image', \dots) \\ \text{if } quality(image) < quality(image').$$

Images or videos can be faulty for a variety of application specific reasons. In our case, poor quality can best be defined as images or videos taken at positions that preclude a high quality stitched panoramic image or video; this can be caused by for example unexpected air turbulence. More generally there are several causes for poor quality videos and images. Photos and videos may be blurred due to air turbulence at the time of recording. The image may experience lens-flare, a crooked horizon, image blurring, lens distortion, noise, chromatic aberration, or sensor occlusion (e.g by dust). Each of these effects occurs due to unexpected environmental conditions such as electromagnetic radiation, or increased dust pollution in the sensor area. However, each of these effects is easy to detect once the image has been taken.

Pollution Sensing

Swarms of drones equipped with carbon monoxide sensors may be sent on demand to different roadway segments to detect pollution levels. Sensing with several independent drones in a specific area, the localized effects of carbon monoxide concentrations in an area can be estimated by taking an average of the readings. In this scenario inaccurate or faulty pollution data, due to erroneous readings of the sensor or other mechanical/environmental impacts, has a lower confidence in the partial ordering than more accurate pollution data. Therefore, POKS should replace the inaccurate data by more accurate data sensed in the same area at the same timestamp as follows. Confidence could for example be measured by closeness of the value calculated by the drone to the mean value across the swarm. Note that *ppm* stands for “parts per million” and is a standard measure of carbon monoxide levels.

$$O_{pollution} : r(\dots, t, g, ppm, \dots) \prec r(\dots, t, g, ppm', \dots) \\ \text{if } confidence(ppm) < confidence(ppm').$$

Noise Sensing

Drone swarms with microphone sensors may be sent on demand to investigate noise levels in city sectors. Drones may be configured so as to converge on areas with higher sound reading to locate and isolate noise trouble spots. In this scenario inaccurate noise data also has a lower confidence in the partial ordering than more accurate noise data. Following the same confidence measurement and POKS rules in pollution sensing, inaccurate data will be discarded to save storage and replaced by more accurate data as

follow. Note that *decibel* is a standard measure of sound levels.

$$O_{noise} : r(\dots, t, g, decibels, \dots) \prec r(\dots, t, g, decibels', \dots) \\ \text{if } confidence(decibels) < confidence(decibels').$$

Note that different urban sensing applications in POKS could use some state attributes in different ways. For example, a different partial ordering is imposed on timestamps for a traffic sensing application (t comes before t') than the ordering imposed on the pollution and noise sensing applications. For most urban sensing applications, each drone in a swarm only can cover a small part of a roadway segment, as shown in Fig. 1. The panoramic traffic view of a roadway segment stitches images collected by drones located over different parts at time t . If a drone captures a low-quality image of its coverage area at time t , this image will impact the panoramic view of the roadway segment. In this case, the drone swarm needs to generate another panoramic view of the same segment at t' for traffic sensing; POKS will use the high-quality view to replace the old one generated at time t . In the pollution and noise sensing applications, *ppm* and *decibel* values with high confidence ratings collected by a drone may be representative of a reasonably wide area around the roadway segment. In these cases, POKS needs to compare all the values collected by different drones at the same timestamp t , and pick the one with the highest confidence rating as the sensing result of the roadway segment. Therefore, for drone swarms in urban sensing, attributes in POKS should be tuned to their intended use within the context of each individual urban sensing application.

5 PERFORMANCE EVALUATION

We implemented the *ADDSEN* middleware using our distributed knowledge management framework and Networked Cyber-Physical Systems framework developed by researchers at SRI [14] and related software packages for adaptive data processing and dissemination in drone swarms. For comparison purposes, we have also implemented three classic data dissemination protocols in our framework; we refer to them as periodic, uniform and neighbor-aware broadcast, respectively. The basic ideas of the these broadcast methods are as follows:

- Periodic broadcast: each drone periodically broadcasts its knowledge to all the neighbors.
- Uniform broadcast: each drone broadcasts its knowledge to all the neighbors with a uniformly distributed probability value between 0 and 1.
- Neighbor-aware broadcast: each drone broadcasts its knowledge to all the neighbors with a non-zero probability $p = r/n$, where r is a reflection parameter and n is the number of outgoing neighbors.

ADDSEN uses learning-based broadcasting and POKS for data processing and dissemination. If the potential receiver is not known to be aware of knowledge K , then K will be broadcast. Otherwise, K is broadcasted with a unique broadcast rate to achieve adaptive dissemination. The broadcast rate is learned from the neighbor and link status after a short period of transmission.

In Eq. 2, if the discount rate $\gamma < 1$, the infinite sum has a finite value as long as the reward sequence r_k is bounded. If $\gamma = 0$, the agent is “myopic” and is concerned only with maximizing immediate rewards: its objective in this case is to learn how to choose action a_t so as to maximize only r_{t+1} . As γ approaches 1, the agent becomes more farsighted, taking future rewards into account more heavily. In Eq. 4, the learning rate α determines

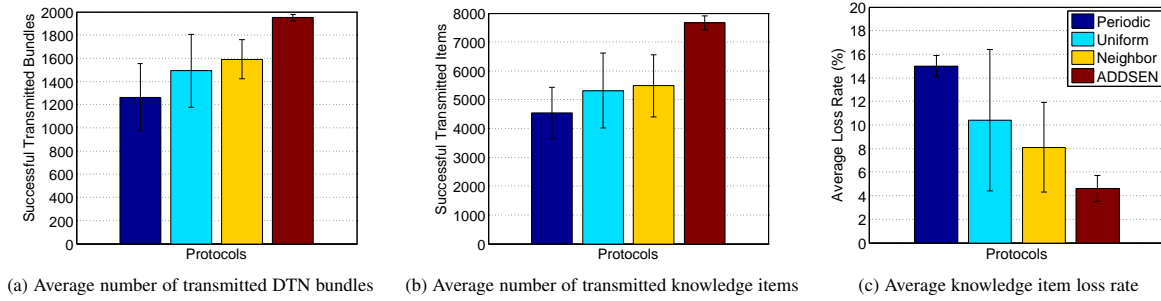


Fig. 7. Data processing and dissemination in one-swarm case.

to what extent the newly acquired information will override the old information. A factor of 1 results in only the most recent information being considered while a factor of 0 ensures that no learning takes place. We choose $\gamma = 0.9$ and $\alpha = 0.1$ for our evaluations. The example in Fig. 5 is used to model the states and actions of online learning.

5.1 Data Dissemination in Drone Swarms

We first test our algorithm in the single-swarm case with 10 drones. The transmission range of each drone is 10 meters, and the 10 drones are not fully connected in the swarm. For each mobile drone, the swarm formation limits the number of neighbors to be not more than 3, which can be achieved by predefined hexagonal lattice structure for sensing grids and neighboring distance constraints used in formation potential p_f (see our swarming method in Section 4.1). The test time is 1000 seconds.

For comparison purposes, we refer to the four broadcast protocols as periodic broadcast, uniform broadcast, neighbor-aware broadcast, and our own learning-based adaptive broadcast, respectively, as *Periodic*, *Uniform*, *Neighbor*, and *ADDSSEN* in Fig. 7.

Fig. 7 (a), (b) and (c) compare the average performance of the 10 drones, with confidence intervals on the number of successful transmitted DTN bundles, the number of successfully transmitted knowledge items, and the knowledge loss rate, respectively. We run the evaluation 10 times to compare the four broadcast protocols under the same settings. It can be observed that broadcasting using probabilistic methods has better performance than periodic broadcasts, since we see fewer transmission collisions with those schemes. Probabilistic broadcasting methods can use the network bandwidth to achieve more effective knowledge transmission and a lower knowledge loss rate. The neighbor-aware broadcast has better performance than the uniform broadcast, because its broadcast probability is selected by considering the number of neighbors of each drone. However, the performance variance of uniform broadcast and neighbor-aware broadcast are higher than periodic broadcast and *ADDSSEN*.

ADDSSEN can effectively estimate the link status based on the neighbors, the BER and piggyback information on the broadcasted knowledge items, thus *ADDSSEN* can learn the communication environment in an online manner and determine a reasonable broadcast rate for each drone. The POKS scheme and awareness attribute in each knowledge item can further reduce the transmission error and improve the transmission throughput. Because of this adaptive learning process in our framework, *ADDSSEN* can achieve higher knowledge transmission, lower

knowledge loss rate, and more stable performance, in comparison with the other schemes.

The performance of *ADDSSEN* is also verified for data dissemination in multi-swarm communication under a DTN scenario. We present the simulation results of data dissemination for a two-swarm case in Fig. 8, and its test scenario from which our evaluation was taken in Fig. 9. In the two-swarm case, there is one swarm with 3 drones and another swarm with 4 drones moving in a 500 meters by 500 meters road area. Fig. 9 clearly demonstrates how drone swarms can coalesce and separate at three different time instants, therefore achieving distributed deployment and co-operation of multiple swarms for urban sensing. The transmission range of each drone in our test is 20 meters. The test time is 1000 seconds. Drone swarms maintain formations specific to their sensing missions. The drones collect data (*e.g.* traffic, pollution, noise) at specific locations. Each drone disseminates knowledge items to other neighbor drones in the same swarm throughout the sensing mission, and to drones in different swarms at meeting points along the DTN route. Partially ordered sharing is applied the received knowledge items for rebroadcasting.

Fig. 8 (a), (b) and (c) compares the performance, on the average number of transmitted DTN bundles, the average number of transmitted knowledge items, and the average knowledge loss rate, respectively, of the four broadcast protocols under the same two-swarm setting. In general, *ADDSSEN* can achieve higher knowledge transmission, lower knowledge loss rate, and more stable performance, in comparison with other broadcast protocols.

Note that the number of transmitted knowledge items using two drone swarms (as shown by the two-swarm case in Fig. 8) is less than using one swarm to perform urban sensing (as shown by the one-swarm case in Fig. 8). In the two-swarm case there are 3 drones in one swarm and 4 drones in another swarm as shown in Fig. 9. In the one-swarm case there are 10 drones all in a unique swarm. Since the number of drones in any swarm in the two-swarm case is less than that in the one-swarm case, the number of transmitted knowledge items is correspondingly decreased as well. In addition, the communication mode of the two-swarm case here is under a DTN scenario with only a few meeting points where data dissemination can occur between two swarms. Therefore, in the two swarm case, most of time, each drone broadcasts fewer knowledge items than that in the one-swarm case.

5.2 Effects of Online Learning

We next investigate the effects of online learning solution in *ADDSSEN*. In order to study this, we collect the action values of State S_2 when the scenario in Fig. 5 is executed. A plot of these readings by time, is shown in Fig. 10(a). Initially, the

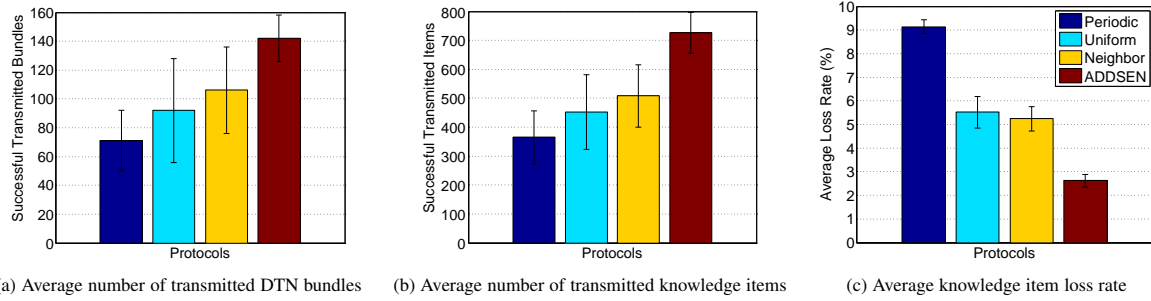


Fig. 8. Data processing and dissemination in the two-swarm case.

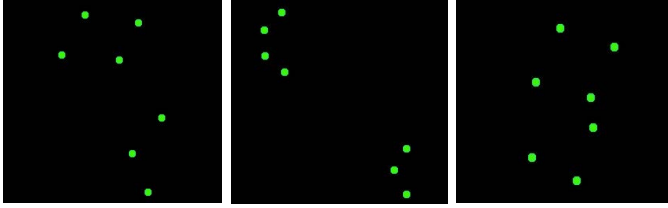
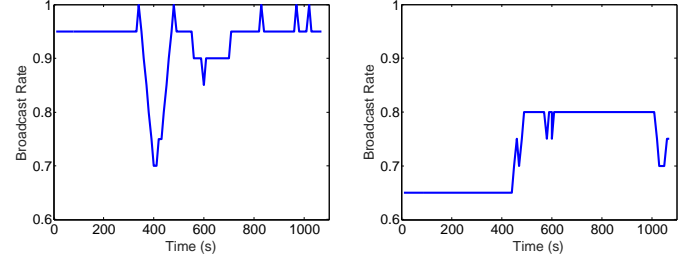


Fig. 9. Simulation scenario of data dissemination for two-swarm case at three different instants.



(a) Broadcast rate of drone $G1$. (b) Broadcast rate of drone $G3$.

Fig. 11. Effects of online learning on multiple drones in the same swarm.

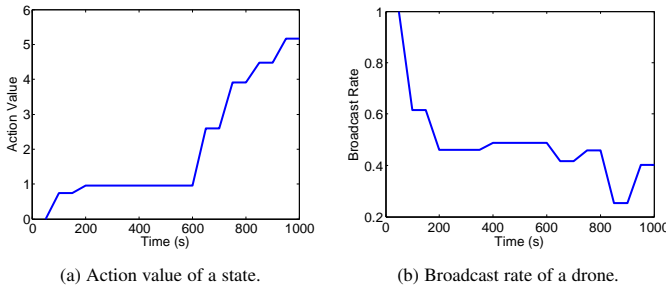


Fig. 10. Effects of online learning on action value and broadcast rate.

action value (Q value) for the state is 0, then it increases and becomes steady around 200s, which means that the agent learns the environment and has reached an accurate approximation of the environment. Each increase in the action value induces a change in the communication environment and the link status. The agent keeps increasing its action value while adapting to the changing environment. The decrease in the rate of change in the Q -values indicates that the action value has converged and tracks the link status well.

The broadcast rate keeps changing as learning progresses. We plot these broadcast rate changes in Fig. 10(b). The initial broadcast rate of this drone is 1. When a drone detects knowledge loss, it adaptively changes the broadcast rate to approach the best state to balance the broadcasting and energy consumption. The broadcast rate becomes steady around 200s, which means that the agent has learned the link status well as the action values have converged. There is a sharp decrease in the broadcast rate after 800s. This decrease is a result of the fact that the two swarms meet at that time and the number of neighbors increases for every drone. Therefore, at the time of the meeting the drone needs to decrease its broadcast rate to avoid broadcast collisions for intra-swarm data dissemination. After the other swarms flies away, the drone begins to broadcast with a higher probability once more.

To evaluate the impact of online learning on data dissemination in drone swarms, we choose two drones with ID $G1$ and $G3$ respectively from the same swarm “Green” as illustrated in Fig. 3. For intra-swarm dissemination, drones in the same swarm need to adjust their broadcast rate frequently due to their periodic broadcasts and communication interference from other drones in the transmission range. Online learning in *ADDSSEN* is proposed to mitigate the interference and improve transmission stability. As shown in Fig. 11 (a) and (b), $G1$ and $G3$ not only interfere with each other’s dissemination, but also suffer communication interference from other drones in the same swarm ($G3$), which may result in severe loss of knowledge items during dissemination and correspondingly trigger the change of broadcast rate as presented in the figure. Online learning allows drones to adjust their broadcast rates and adapt to changes in the communication environment. In addition, online learning always tries to improve the broadcast rate of each drone by approaching a better state whenever the current state is associated with a low broadcast rate. For example, when $G1$ decreases its broadcast rate around at time 400s, online learning in $G3$ can sense the change of link status and then adaptively increase the broadcast rate.

In our multi-swarm scenario, drones collect data at work locations above the public roadway and can disseminate the information to other drones from different swarms when they fall within its transmission range, along the flight path as shown in Fig. 9. Through inter-swarm communication, urban sensed data can be collected by multi-hop data relay under disruptive scenario; also the possibility of losing data can be decreased by data replication. Fig. 12 evaluates the effects of online learning on inter-swarm data dissemination. We choose two drones with ID $R1$ and $G2$ respectively from swarm “Red” and swarm “Green” as illustrated in Fig. 3. From Fig. 12 (a) and (b), we learn that $R1$ and $G2$ meet around at time 600s within each other’s transmission range. In the meeting period from approximately time 600s to time

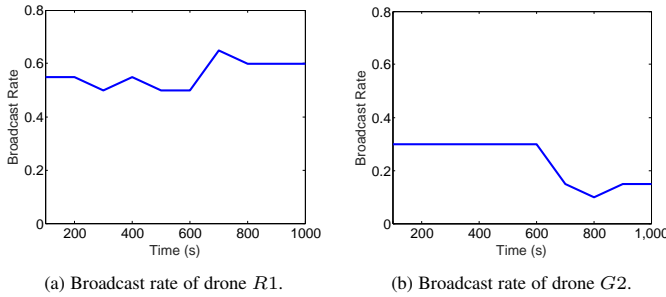


Fig. 12. Effects of online learning on multiple drones from different swarms.

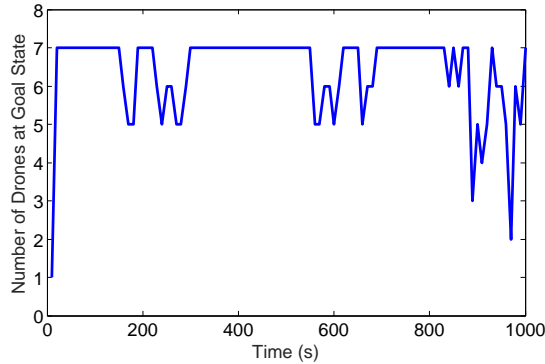


Fig. 13. Effects of cooperative dissemination on multiple drones.

700s, *R1* is trying to increase its broadcast rate, therefore *G2* adaptively decreases its broadcast rate to avoid interference. After time 700s, *R1* and *G2* fly out of each other's transmission ranges, then *G2* can increase its broadcast rate for data dissemination in its local swarm. Also, the broadcast rates of *R1* and *G2* become stable when the two swarms fly far apart.

Overall, our online learning scheme in *ADDSEN* can select satisfactory broadcast rates for each drone while accommodating other drones' transmissions (both intra/intre-swarm).

5.3 Effects of Balanced Dissemination

Periodic intra/inter-swarm dissemination may reduce the storage and energy reserves of each drone to a different extent. To maintain the in-field time of a drone swarm as a sensing grid, we have designed an optimization program, as introduced in Section 4.4, to balance the dissemination tasks in drone swarms so that we can relocate the work load for heavy-duty drones and extend their lifetime within their swarm.

To investigate the effects of cooperative dissemination in drone swarms, we make two swarms (one with 3 drones and another with 4) fly close to each other. Sensing coverage of the two swarms are overlapped for a period around 1000s; it means that at least one pair of drones, one from each swarm are within a sufficient transmission range to guarantee inter-swarm dissemination during this period. Each swarm selects the drone with the most residual energy as its balancer to run the optimization program for intra-swarm cooperative dissemination. This balancer also coordinates with another balancer from the different swarm for inter-swarm dissemination. As shown in Fig. 13, in the experimental scenario, our balancing optimization solution ensures that at least 5 drones reach the goal state by at most time 1000s. After time 800s, some

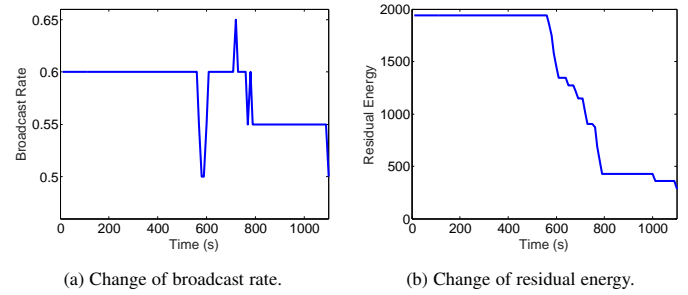


Fig. 14. Effects of cooperative dissemination on broadcast rate and energy.

drones with limited resources need to conserve these resources. The balancer adjusts the cooperative solution at every drone and selects partial drones with better storage and energy capacity to disseminate and receive sensed data.

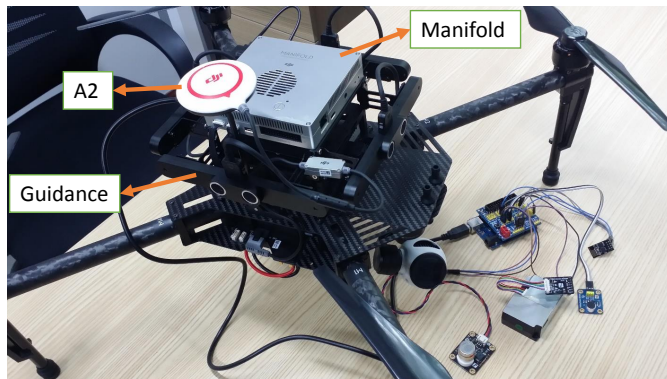
Fig. 14 evaluates the effect of cooperative dissemination on a single drone over time. Since each broadcast has a uniform and relatively high energy cost, we assume the energy consumption per broadcast is 1 unit and evaluate the changes in broadcast rate and residual energy in drones after applying our optimization program. As shown in Fig. 14 (a), for a drone with an average broadcast rate (probability of broadcasting of more than 0.6), between time 610s and 760s, its energy is consumed quickly given a high constant dissemination rate (shown in figure Fig. 14 (b)). Once a drone's residual energy is low (e.g. residual energy is only 425 units at time 790s), the load balancer, a selected drone running our optimization program within a swarm, will decrease the low energy drone's broadcast rate to conserve its energy, while identifying candidate drones with better capacity to disseminate the same knowledge items. Therefore, our optimization program can achieve a balanced dissemination and extend each drone's lifetime within the swarm.

5.4 Effects of POKS for Urban Sensing

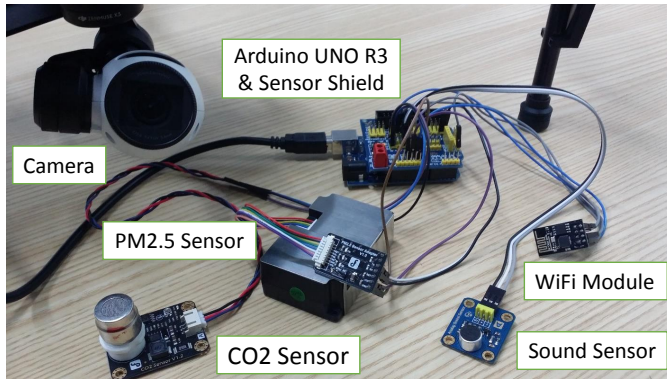
In previous evaluations on data dissemination, POKS has been used in our distributed knowledge management framework to process data before dissemination. Its equivalence relation (\equiv) and partial ordering (\prec) schemes on knowledge items can select fresh and correct data to broadcast, and remove redundant and faulty data in local base (see Section 3.1), therefore present an efficient way to avoid unnecessary data dissemination in *ADDSEN*.

To further evaluate the performance of POKS for urban sensing scenario, we have designed a drone testbed with urban sensing capabilities based on DJI Matrice 100 flight platform [37], as shown in Fig. 15 (a). Matrice 100 is a state-of-the-art quadcopter for developers. In comparison with other commercial quadcopters, it has some additional easy-to-fly features such as A2 model to provide GPS and compass functions, and Guidance as a visual sensing system to automatically detect obstacles and avoid collision in real time. Matrice 100 also supports the Manifold embedded computer for DJI SDK-based software and application development. The Ubuntu-based Manifold runs the NVIDIA Tegra K1 quad-core ARM Cortex-A15 processor with 192 GPU cores, and has standard and extended connectors and interfaces, therefore it can connect our third-party urban sensing platform and collects and analyzes data in the air.

The urban sensing platform we built is shown in Fig. 15 (b). Matrice 100 is configured with a Zenmuse X3 camera that can be



(a) Flight platform.



(b) Sensing platform.

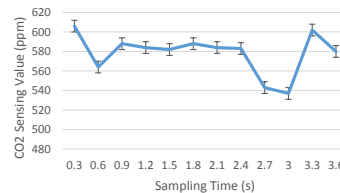
Fig. 15. Drone testbed with urban sensing capabilities.

used as the traffic camera in our platform. PM 2.5 sensor and CO2 sensor are deployed to detect various pollution sources. Sound sensor can determine the magnitude of noise level. WiFi module can support wireless data dissemination between neighboring drones. These sensing and communication devices are all connected to a sensor shield that is plugged on top of an Arduino UNO R3 [38] as a sensing extension, and the Arduino board can send data collected from the sensing extension to the Manifold that is running our POKS-based data processing schemes.

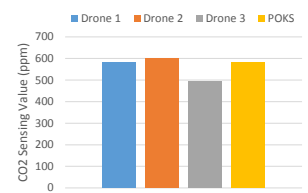
Some urban sensing data collected from the drone platform are listed by time sequence in our customized application, as shown in Fig. 16 (a). Each entry in the example displays PM 2.5, noise volume and CO2 values collected by corresponding environmental sensors at the same time. The PM 2.5 and CO2 are two major urban pollution sources. Noise level is measured by mapping units in the acoustic domain to units in the electrical domain such as analog output voltage. Due to the space limit, we only present the impact of POKS for CO2 data. As shown in Fig. 16 (b), our application running in every drone collects a CO2 sample per 0.3 second. Given a time window of nearly 4 seconds to sense a roadway segment, a single drone can totally collect 12 CO2 samples. Considering sensing variances among the 12 samples, *ADDSEN* applies POKS to evaluate CO2 concentration for the roadway segment. Specifically, POKS uses the metric “confidence” to replace inaccurate samples by more accurate samples sensed in the same geographical area; the sample with highest confidence is selected as the representative CO2 value for the area. Confidence for pollution evaluation is measured by closeness of the value calculated by the drone to the mean value of all samples in the same area (See Section 4.5).

PM2.5:	81 $\mu\text{g}/\text{m}^3$	VOL:	692 mv	CO2:	583 ppm
PM2.5:	83 $\mu\text{g}/\text{m}^3$	VOL:	695 mv	CO2:	543 ppm
PM2.5:	87 $\mu\text{g}/\text{m}^3$	VOL:	686 mv	CO2:	537 ppm
PM2.5:	87 $\mu\text{g}/\text{m}^3$	VOL:	681 mv	CO2:	602 ppm
PM2.5:	86 $\mu\text{g}/\text{m}^3$	VOL:	697 mv	CO2:	580 ppm

(a) An example of urban sensing results.



(b) POKS for single drone.



(c) POKS for drone swarm.

Fig. 16. Effects of POKS on sensing data.

Single drone may generate inaccurate or faulty pollution data due to erroneous readings of the sensor or other mechanical/environmental impacts, therefore drone swarm can be used to compensate this kind of error, where above POKS method is also applicable. As shown in Fig. 16 (c), we have evaluated CO2 samples collected by a drone swarm composed of three drones. For a given roadway segment, the estimated CO2 concentrations by drone 1, drone 2 and drone 3 are 583 ppm, 602 ppm and 495 ppm respectively. After the three drones disseminate and exchange these CO2 values, *ADDSEN* can first derive their mean value as 560 ppm. The confidence here is measured by closeness of the value calculated by the drone to the mean value across the swarm, therefore the CO2 value 583 ppm collected by drone 1 is selected by *ADDSEN* to represent the CO2 concentration in the roadway segment. In contrast, drone 3 has a lower confidence than drone 1 and drone 2 in the swarm, indicating faulty sampling happened in the area by its urban sensing platform.

6 CONCLUSION

In this paper, we present a middleware solution for adaptive data processing and dissemination for drone swarms in urban sensing. The features of *ADDSEN* include:

- distributed knowledge management based on partially ordered knowledge sharing and a cyber-physical sensing framework,
- online learning based data dissemination to adaptively select the best broadcast rate according to link status,
- and, a cooperative dissemination approach to balancing storage and energy allocation in drone swarms.

We implemented *ADDSEN* using our framework, and provide evaluation results to show that it can achieve adaptive data processing and dissemination, and extend the lifetime of a drone formation.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their insightful comments. This work was partially supported by

the National Science Foundation under Grant No. 0932397, the Office of Naval Research under Grant No. N00014-10-1-0365, University of California Center on Economic Competitiveness in Transportation (UCCONNECT), and the Intel Collaborative Research Institute for Sustainable Connected Cities (ICRI Cities). Professor Regan was partially supported during the period of this research by the Reuben Smeed Professorial Research Fellowship at University College London.

REFERENCES

- [1] Z. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks," *IEEE Network*, vol. 30(1), pp. 100–105, 2016.
- [2] K. Krishnamoorthy, D. Casbeer, and M. Pachter, "Minimum time UAV pursuit of a moving ground target using partial information," in *International Conference on Unmanned Aircraft Systems*, 2015, pp. 204–208.
- [3] S. Chaumette, R. Laplace, C. Mazel, R. Mirault, A. Dunand, and Y. L. J.-N. Perbet, "CARUS, an operational retasking application for a swarm of autonomous UAVs: First return on experience," in *IEEE MILCOM*, 2011.
- [4] S. Jwa and U. Ozguner, "Multi-UAV Sensing Over Urban Areas via Layered Data Fusion," in *IEEE/SP 14th Workshop on Statistical Signal Processing*, 2007, pp. 576–580.
- [5] H. Weinschrott, F. Drr, and K. Rothermel, "StreamShaper: Coordination algorithms for participatory mobile urban sensing," in *IEEE MASS*, 2010, pp. 195–204.
- [6] Unmanned Aircraft Systems, <https://www.faa.gov/uas/>.
- [7] L. A. Klein, D. R. P. Gibson, and M. K. Mills, *Traffic Detector Handbook: Third Edition—Volume I*. Federal Highway Administration, Turner-Fairbank Highway Research Center, 2006.
- [8] U. Lee and M. Gerla, "A survey of urban vehicular sensing platforms," *Computer Networks*, vol. 54(4), pp. 527–544, 2010.
- [9] DJI Phantom 3, <http://www.dji.com/product/phantom-3>.
- [10] Parrot AR.Drone 2.0, <http://ardrone2.parrot.com>.
- [11] Agriculture the New Game of Drones, <http://www.futuristspeaker.com/2013/08/agriculture-the-new-game-of-drones/>.
- [12] A. Zanella, N. Bui, A. P. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1(1), pp. 22–32, 2014.
- [13] S. Mahmoud and N. Mohamed, "Toward a Cloud Platform for UAV Resources and Services," in *IEEE Symposium on Network Cloud Computing and Applications*, 2015, pp. 23–30.
- [14] Networked Cyber Physical Systems at SRI, <http://ncps.csl.sri.com>.
- [15] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, "Fair content dissemination in participatory DTNs," *Ad Hoc Networks*, vol. 10(8), pp. 1633–1645, 2012.
- [16] A. Förster and A. L. Murphy, "Froms: A failure tolerant and mobility enabled multicast routing paradigm with reinforcement learning for WSNs," *Ad Hoc Networks*, vol. 9(5), pp. 940–965, 2011.
- [17] M. S. Haghighi, Y. Xiang, V. Varadharajan, and B. G. Quinn, "A Stochastic Time-Domain Model for Burst Data Aggregation in IEEE 802.15.4 Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 64(3), pp. 627–639, 2015.
- [18] S. Dietzel, F. Kargl, G. J. Heijenk, and F. Schaub, "Modeling in-network aggregation in VANETs," *IEEE Communications Magazine*, vol. 49(11), pp. 142–148, 2011.
- [19] M.-O. Stehr, M. Kim, and T. McCarthy, "A Distributed Computing Model for Dataflow, Controlflow, and Workflow in Fractionated Cyber-Physical Systems," *Computing with New Resources, Lecture Notes in Computer Science*, vol. 8808, pp. 379–393, 2014.
- [20] W. Hu and G. Cao, "Energy optimization through traffic aggregation in wireless networks," in *INFOCOM*, 2014.
- [21] B. Garbinato, A. Holzer, and F. Vessaz, "Context-aware broadcasting approaches in mobile ad hoc networks," *Computer Networks*, vol. 54(7), pp. 1210–1228, 2010.
- [22] H. Alshaer and E. Horlait, "An optimized adaptive broadcast scheme for inter-vehicle communication," in *Vehicular Technology Conference*, 2005, pp. 2840–2844.
- [23] H. Jiang, H. Guo, and L. Chen, "Reliable and Efficient Alarm Message Routing in VANET," in *Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems Workshops*, 2008, pp. 186–191.
- [24] X. Jiao, W. Lou, J. Ma, J. Cao, X. Wang, and X. Zhou, "Minimum Latency Broadcast Scheduling in Duty-Cycled Multihop Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23(1), pp. 110–117, 2012.
- [25] K. Daniel, A. Wolff, and C. Wietfeld, "Protocol Design and Delay Analysis for a MUAV-Based Aerial Sensor Swarm," in *IEEE WCNC*, 2010.
- [26] InView Unmanned Aircraft System, <http://www.barnardmicrosystems.com/inview/overview.html>.
- [27] S. Guo, C. Wang, and Y. Yang, "Joint Mobile Data Gathering and Energy Provisioning in Wireless Rechargeable Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 13(12), pp. 2836–2852, 2014.
- [28] A. Abdulla, Z. Fadlullah, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "Toward Fair Maximization of Energy Efficiency in Multiple UAS-Aided Networks: A Game-Theoretic Methodology," *IEEE Transactions on Wireless Communications*, vol. 14(1), pp. 305–316, 2015.
- [29] D. Ye and M. Zhang, "A Self-Adaptive Strategy for Evolution of Cooperation in Distributed Networks," *IEEE Transactions on Computers*, vol. 64(4), pp. 899–911, 2015.
- [30] J. Zhan, L. Wang, X. Li, W. Shi, C. Weng, W. Zhang, and X. Zang, "Cost-Aware Cooperative Resource Provisioning for Heterogeneous Workloads in Data Centers," *IEEE Transactions on Computers*, vol. 62(11), pp. 2155–2168, 2013.
- [31] W. Kowalczyk and K. Kozłowski, "Artificial potential based control for a large scale formation of mobile robots," in *4th Int. Workshop on Robot Motion and Control*, 2004, pp. 285–291.
- [32] M. Miki, T. Hiroyasu, and K. Ono, "Simulated annealing with advanced adaptive neighborhood," in *Second Int. Workshop on Intelligent Systems Design and Application*, 2002, pp. 113–118.
- [33] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Auton. Robots*, vol. 17(2-3), pp. 137–162, 2004.
- [34] H. P. Williams, *Logic and Integer Programming*. Springer, 2009.
- [35] CPLEX Optimizer, <http://www.aimms.com/aimms/solvers/cplex/>.
- [36] GUROBI Optimizer, <http://www.gurobi.com/>.
- [37] Matrice 100: The quadcopter for developers, <http://www.dji.com/product/matrice100>.
- [38] Arduino - ArduinoBoardUno, <https://www.arduino.cc/en/main/arduinoBoardUno>.



Di Wu received the B.E. degree in Communication Engineering and the M.E. degree in Communication and Information Systems from Hunan University, Changsha, China. He received the M.S. and Ph.D. degree in Computer Science from the University of California, Irvine, USA. He was a Staff Research Associate at the University of California, Irvine, a Visiting Researcher at IBM Research, and a Student Research Associate at the SRI International, Menlo Park, USA. He is currently a Research Associate at Imperial College London and Intel Collaborative Research Institute for Sustainable Connected Cities. He is also an Adjunct Researcher at the University of California Transportation Center. His research interests include wireless networks and mobile computing, Internet-of-things and cyber-physical systems, smart cities and big data. He has actively served on many conference committees and is currently Associate Editor for the IEEE Transactions on Intelligent Transportation Systems. He is a member of the IEEE and the ACM.



Dmitri I. Arkhipov received the B.S. degree in information and Computer Science and the M.S. degree in Computer Science from the University of California, Irvine in 2009 and 2012 respectively. He is currently working toward his Ph.D. degree in Computer Science at the University of California, Irvine. His research interests include parallel and distributed systems, large scale combinatorial optimization, and cyber-physical systems.



Minyoung Kim is a Computer Scientist in the Computer Science Laboratory at SRI International. She received her B.S. (1999) in Computer Engineering and M.S (2001) degrees in Electrical Engineering and Computer Science from Seoul National University. She received her Ph.D. degree (2008) in Information and Computer Science from University of California, Irvine. Her research interests include networked cyber physical systems, declarative information-centric networking, cross-layer optimization, and

formal methods.



Carolyn L. Talcott is a Program Director in the Computer Science Laboratory of SRI International. She holds a Ph.D. in computer science from Stanford University (1985) and a Ph.D. in chemistry from the University of California, Berkeley (1966). She has over 25 years experience in formal modeling and analysis. For the last 11 years she has also been working in computational modeling of biological systems. At SRI She is leading research in symbolic systems biology, security protocol analysis, and formal

analysis applied to cyber-physical systems and next-generation networks. She was named an SRI Fellow in 2011.



Amelia C. Regan received a B.A.S. degree in Systems Engineering from the University of Pennsylvania, an M.S. in Applied Mathematics from the Johns Hopkins University and an M.S. and Ph.D. in Transportation Systems Engineering from the University of Texas, Austin. She is currently a professor of computer science and transportation systems engineering at the University of California, Irvine. Her research is focused on algorithm development for optimization of transportation and communication systems.

She has also taught short courses at the Athens University of Business and Economics and the National Technical University of Denmark and was an operations research analyst with the Association of American Railroads and United Parcel Service prior to earning her Ph.D. degree.



Julie A. McCann is a professor in Computer Systems at Imperial College London. Her research centers on highly decentralized and self-organizing scalable algorithms for spatial computing systems *e.g.* wireless sensing networks. She leads both the Adaptive Embedded Systems Engineering Research Group and the Intel Collaborative Research Institute for Sustainable Cities, and is currently working with NEC and others on substantive smart city projects. She has received significant funding through bodies

such as the UK's EPSRC, TSB and NERC as well as various international funds, and is an elected peer for the EPSRC. She has actively served on, and chaired, many conference committees and is currently Associative Editor for the ACM Transactions on Autonomous and Adaptive Systems. She is a member of the IEEE and the ACM as well as a Chartered Engineer, and was elected as a Fellow of the BCS in 2013.



Nalini Venkatasubramanian received the MS and PhD degrees in computer science from the University of Illinois, Urbana-Champaign. She is a professor in the School of Information and Computer Science, University of California, Irvine. Her research interests include distributed and parallel systems, middleware, mobile environments, multimedia systems/applications, and formal reasoning of distributed systems. She was a member of technical staff at Hewlett-Packard Laboratories in Palo Alto, California,

for several years, where she worked on large-scale distributed systems and interactive multimedia applications. She has also worked on various database management systems and on programming languages/compilers for high performance machines. She is a senior member of the IEEE and ACM.