Coordinating UAVs in Dynamic Environments by Network-Aware Mission Planning

Marcello Balduccini, Duc N. Nguyen, William C. Regli
Applied Informatics Group
Drexel University, Philadelphia, PA
{mb3368, dn53, regli}@drexel.edu

***Abstract*—Traditional AI planning has been used successfully in many domains, including logistics, scheduling and game playing. This paper examines how AI planning techniques can be extended to coordinate teams of unmanned aerial vehicles (UAVs) in dynamic environments. Specifically challenging are real-world environments where UAVs and other network-enabled devices must communicate to coordinate—and communication actions are neither reliable nor free. Such network-centric environments are common in military, public safety and commercial applications, yet most planning research (even multi-agent planning) usually takes communications among distributed agents as a given. The emerging application challenge of unmanned systems makes this problem of central focus. This work examines the problem of planning, plan monitoring and coordination of the mission of multiple UAVs in a communication-constrained environment. The work introduces several abstractions that enable AI planners to reason about communication and networking knowledge; and provides the underlying network system the means for including mission data as part of network operations. This work has been empirically validated using a distributed network-centric software evaluation testbed and the results provide guidance to designers in how to understand and control intelligent systems that operate in these environments.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) promise to revolutionize the way in which we use our airspace. From talk of automating the navigation for major shipping companies to the use of small helicopters as "deliverymen" that drop your packages at the door, it is clear that our airspaces will become increasingly crowded in the near future. This increased utilization and congestion has created the need for new and different methods of coordinating assets using the airspace. Currently, airspace management is the job for mostly human controllers. As the number of entities using the airspace vastly increases—many of which are autonomous—the need for improved autonomy techniques becomes evident.

The challenge in an environment full of UAVs is that the environment is highly dynamic and the communications environment is uncertain, which makes coordination very difficult. Communicative actions in such realistic environments are neither reliable nor free.

This paper presents a novel application of a network-aware planner and an intelligent plan-aware network layer and apply this to the problem of UAV coordination. Currently, AI planning does not incorporate network constraints in the planning step, nor does the network reason about its state and optimize for the plan. With network-aware planning, a planner (either centralized or decentralized) incorporates a network communications model and estimated conditions as part of state evaluation. With plan-aware networking, an intelligent network middleware service is provided the plan and ensures quality of service (QoS) for plan execution. The approach provided in this paper focuses on network-aware planning that incorporates a basic network model (communications range). This model is simple in nature, however, it demonstrates that even a slight increase of knowledge in the network state will affect the mission plan.

The paper is organized as follows: The next section describes relevant planner systems and reasoning techniques followed by a motivating scenario that applies to UAV coordination. The Technical Approach describes network-aware planning using example problem instances, the plan-aware networking layer components, and empirical results. The following section describes the network-centric evaluation testbed used for simulations. Finally, the paper concludes with a discussion and future work.

II. Related Work

Incorporating network properties into planning and decision-making has been investigated in [1]. The author's results indicate that plan execution effectiveness and performance is increased with the increased network-awareness during the planning phase. The UAV coordination approach in this current work combines network-awareness during planning phases with a plan-aware network layer.

The problem of mission planning for UAVs under communication constraints has been addressed in [2], where an ad-hoc task allocation process is employed to engage under-utilized UAVs as communication relays. In our work, we do not separate planning from the engagement of under-utilized UAVs, and do not rely on ad-hoc, hard-wired behaviors. Our approach gives the planner more flexibility and fine-grained control of the plan actions, and allows for the emergence of sophisticated behaviors.

The architecture adopted in this work is an evolution of [3], which can be viewed as an instantiation of the BDI agent model [4], [5]. Here, the architecture has been extended to include a centralized mission planning phase, and to reason about other agents' behavior. Recent related work on logical theories of intentions [6] can be further integrated into our approach to allow for a more systematic hierarchical characterization of the actions, which is likely to increase performance.

Traditionally, AI planning techniques have been used (to great success) to perform multi-agent teaming, and UAV coordination. Multi-agent teamwork decision frameworks such

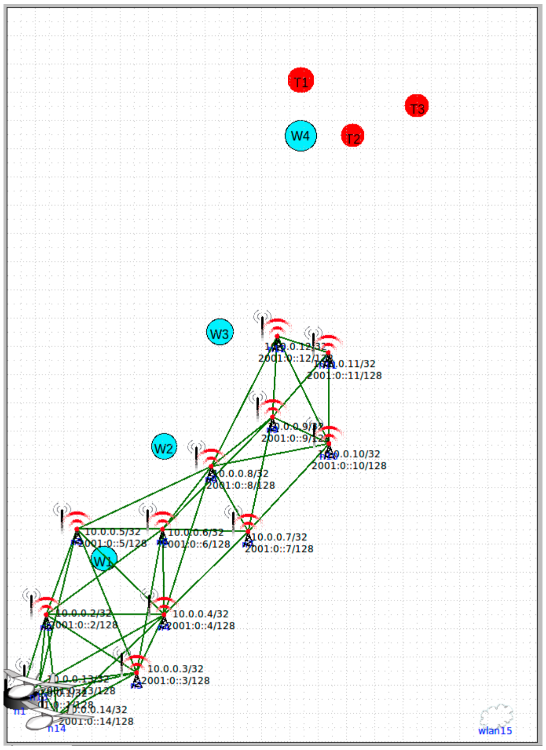


Fig. 1: An example problem instance for UAV coordination. The home base is the black node in the lower left corner and the targets are shown as red dots in the upper right corner. Relays form a mesh and extend the network. The blue nodes are waypoints for the UAVs. UAVs travel between waypoints to the targets and back.

as [7] may factor communication costs into the decision-making. However, the agents do not actively reason about other agent's observed behavior, nor about the communication process. Moreover, policies are used as opposed to reasoning from models of domains and of agent behavior.

The reasoning techniques used in the present work have already been successfully applied to domains ranging from complex cyber-physical systems [8], to agent-based negotiation [9] and to workforce scheduling [10]. To the best of our knowledge, however, they have never been applied to domains involving realistic communications.

High-fidelity multi-agent simulators (e.g., AgentFly [11]) do not account for network dynamism nor provide a realistic network communications model. For this reason, we base our simulator on the Common Open Research Emulator (CORE) [12]. CORE provides network models in which communications are neither reliable nor free.

III. Motivating Scenario

To motivate the need for network-aware planning and plan-aware networking, consider a simple UAV coordination problem, depicted in Figure 1, in which two UAVs are tasked with taking pictures of a set of three targets, and with relaying the information to a home base.

Fixed relay access points extend the communication range of the home base. UAVs can share images of the targets with each other and with the relays when they are within radio range. A naïve solution to this problem consists of disregarding the networking component of the scenario, and generating a mission plan in which each UAV flies to a different set of targets, takes pictures of them, and flies back to the home base, where the pictures are downloaded. This solution, however, is not satisfactory. First, it is inefficient, as it requires that the UAVs fly all the way back to the home base before the images can be used. The time it takes for the UAVs to fly back may easily render the images too outdated to be useful. Second, disregarding the network during the reasoning process may lead to mission failure in the case of unexpected events, such as obstructions disrupting transit to and from the home base after a UAV has reached a target. Even if the UAVs are capable of autonomous behavior, they will not be able to complete the mission unless they take advantage of the network.

Another adopted solution consists of acknowledging the availability of the network, and assuming that the network is available throughout plan execution. A corresponding mission plan would instruct each UAV to fly to a different set of targets, and take pictures of them, while the network relays the data back to the home base. This solution is *optimistic* in that it assumes that the radio range is sufficient to reach the area where the targets are located, and that the relays will work correctly throughout the execution of the mission plan.

This optimistic solution is more efficient than the previous one, since the pictures are received by the home base soon after they are taken. Under realistic conditions, however, the strong assumptions it relies on may easily lead to mission failure—for example, if the radio range does not reach the area where the targets are located.

In the present work, the mission planner takes into account not only the presence of the network, but also its configuration and characteristics, taking advantage of available resources whenever possible. A mission planner following this approach is given information about the radio range of the relays and determines, for example, that the targets are out of range. A mission plan that takes this information into account consists in having one UAV fly to the targets and take pictures, while the other UAV remains in a position to act as a network bridge between the relays and the UAV that is taking pictures. This solution is as efficient as the optimistic solution presented earlier, but does not rely on the same strong assumptions.

Conversely, when given a mission plan, an intelligent network middleware service capable of sensing conditions and modifying network parameters (e.g., modify network routes, limit bandwidth to certain applications, and prioritize network traffic) is able to adapt the network to provide optimal communications needed during plan execution. A relay or UAV running such a middleware is able to interrupt or limit bandwidth given to other applications to allow the other UAV to transfer images and information toward home base. Without this traffic prioritization, network capacity could be reached prohibiting image transfer.

IV. Technical Approach

In this section, we formulate the problem in more details and discuss our approach. We discuss separately network-aware planning and plan-aware networking, however, the two aspects of this work can be merged in a straightforward way.

A. Problem Formulation

A problem instance for coordinating UAVs to observe targets and deliver information (e.g., images) to a home base is defined by a set of UAVs, $u_1, u_2, \ldots$, a set of targets, $t_1, t_2, \ldots$, a (possibly empty) set of fixed radio relays, $r_1, r_2, \ldots$, and a home base. The UAVs, the relays, and the home base are called radio nodes (or network nodes). Two nodes are in radio contact if they are within a radius ρ from each other, called radio range[1]. The UAVs are expected to travel from the home base to the targets to take pictures of the targets and deliver them to the home base. A UAV will automatically take a picture when it reaches a target. If a UAV is within radio range of another UAV, or a relay, the picture is automatically shared. The communications network is possibly shared by multiple, concurrent missions. Thus, traffic over it may be due not only to the transfer of the pictures of the targets, but also to other communications, such as audio/video feeds from remotely controlled UAVs or intelligence reports from warfighters on the battlefield. From the UAVs' perspective, the environment is only partially observable. Features of the domain that are observable to a UAV u are: (1) which radio nodes u can and cannot communicate with by means of the network; and (2) the position of any UAV that is near u.

For this problem instance, the goal is to have the UAVs take a picture of each of the targets so that (1) the task is accomplished as quickly as possible, and (2) the total "staleness" of the pictures is as small as possible. Staleness is defined as the time elapsed from the moment a picture is taken, to the moment it is received by the home base. While the UAVs carry on their tasks, the relays are expected to actively prioritize traffic over the network in order to ensure mission success and further reduce staleness.

B. System Architecture

The architecture used in this project follows the BDI agent model [4], [5] which provides a good foundation because of its logical underpinning, clear structure and flexibility. In particular, we build upon instances of this model [13], [3], [14] that employ directly-executable logical languages with good computational properties while at the same time ensuring elaboration tolerance [15] and elegant handling of incomplete information, non-monotonicity, and dynamic domains.

Figure 2 shows a sketch of the information flow in the system. Given an initial description of the domain and of the problem instance, a centralized *mission planner* finds a plan that uses the UAVs to accomplish the above goals.

The UAVs execute the plan individually. As plan execution unfolds network state changes, affecting communications availability. The UAVs may move out of range of each other and the relays. Unexpected events, e.g. relays failing or temporarily becoming disconnected, also affect network connectivity. When network state changes, UAVs reason in a decentralized, autonomous manner to over come these issues. To avoid overheads and communications bottlenecks, reasoning is carried out locally with the UAV once the mission starts. The key to accounting and compensating for state changes is to actively employ current information about the communications state in the reasoning processes.

[1]For simplicity, the radio nodes use comparable network devices, and ρ is uniform throughout the environment.

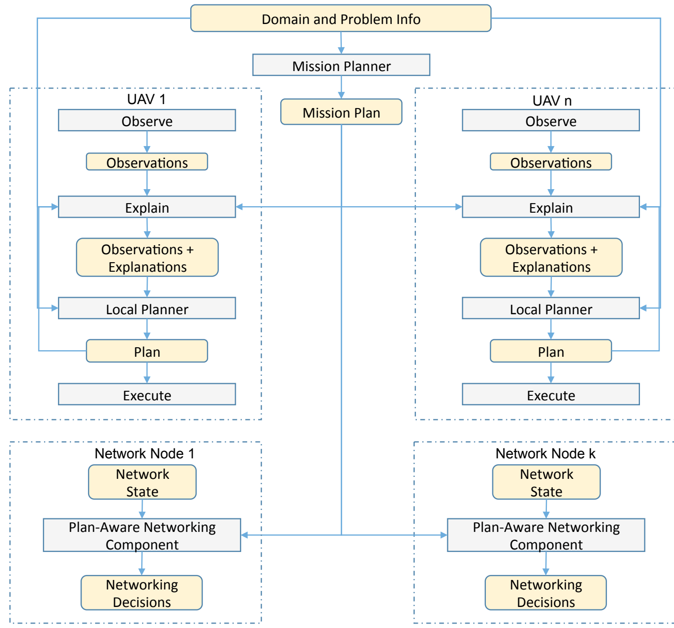


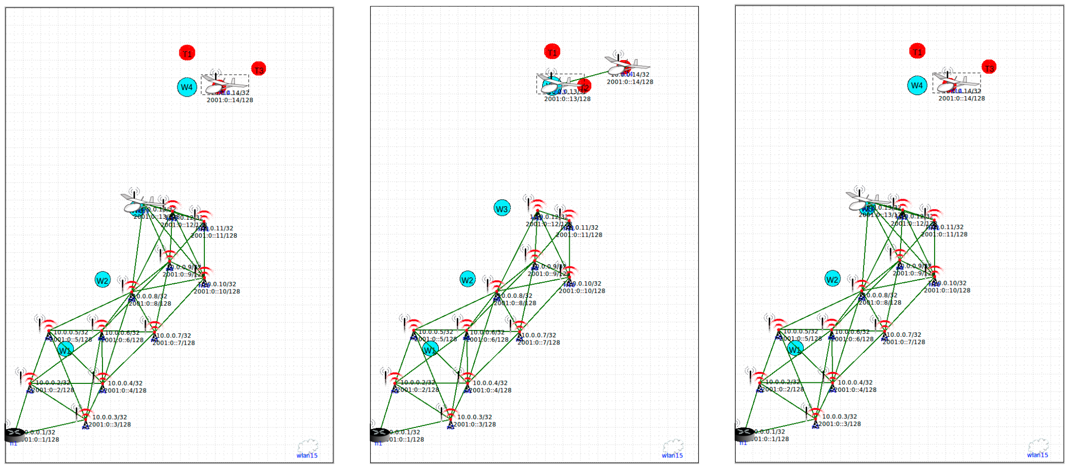
Fig. 2: Information flow in our framework. Note: the tasks in the various boxes are executed only when necessary.

The reasoning tasks (mission planning, explanation, planning within the agents) rely on a high-level description of the environment and of the mission tasks to be performed. The description also includes any relevant constraints (hard or soft), priorities, and policies to be taken into consideration [16], [17]. Solutions are found using state-of-the-art Artificial Intelligence algorithms [18]. These algorithms are the result of several decades of research in constraint-based reasoning and satisfiability solving, and employ learning mechanisms and other sophisticated techniques that make them capable of high performance. In fact, their performance is in many cases superior to that of ad-hoc algorithms. Furthermore, they are entirely general-purpose, which ensures substantial flexibility and extensibility of the approach: changes in the environment or additional requirements can typically be handled by simple changes in the description provided to the algorithms.

Network-Aware Planning. The planners for each UAV incorporates network state (for simplicity, just the communications range is factored into reasoning) into the reasoning process. For network-aware planning, the mission planner exploits information about the radio range and the fact that UAVs are able to relay images between each other.

The next paragraphs outline two experiments, in increasing order of sophistication, which showcase the features of our approach, including non-trivial emerging interactions between the UAVs and the ability to work around unexpected problems autonomously. The reader is directed to [19] and [20] for a description of how the agent's algorithms achieve the behavior outlined in the experiments.

Example Instance 1. Consider the environment shown in in Figure 3. Two UAVs, u_1 and u_2 are initially located at the home base in the lower left corner. The home base, relays and targets are positioned as shown in the figure, and the radio range is set to 7 grid units.



(a) Step 5: u_1 is disconnected from homebase.

(b) Step 6: u_2 connects with u_1 and transfers images t_2 and t_3.

(c) Step 7: u_2 reconnects with relays, transfers images to the home base.

Fig. 3: Example instance 1 illustrating "data mule" information relaying between u_1 and u_2.

The mission planner finds a plan in which the UAVs begin by traveling toward the targets. While u_1 visits the first two targets, u_2 positions itself so as to be in radio contact with u_1 (Figures 3a and 3b). Upon receipt of the pictures, u_2 moves to within range of the relays to transmit the pictures to the home base (Figure 3c). At the same time, u_1 flies toward the final target. UAV u_2, after transmitting pictures to home base, moves to re-establish radio contact with u_1 and to receive the picture of t_3. Finally, u_2 moves within range of the relays to transmit picture of t_3 to the home base.

Remarkably, in this problem instance the mission plan establishes u_2 as a "data mule" in order to cope with the limits of the network. The "data mule" behavior is well-known in sensor network applications [21], [22]; however, this behavior is not hard-coded in the mission planner but rather it emerges from the consideration of the available options. The data-mule behavior is selected because it optimizes the evaluation metrics (mission length and total staleness).

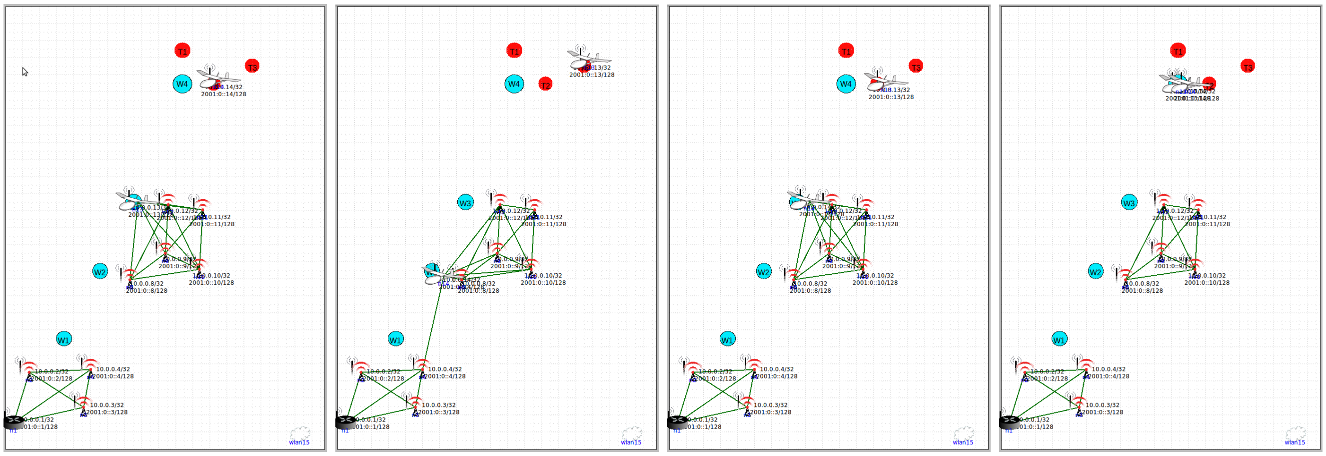
Example Instance 2. Now consider a more challenging and realistic example (Figure 4) in which the UAVs must cope with unexpected events occurring during mission execution. Environment and mission goals are as in the previous example.

The mission planner produces the same plan described earlier[2], in which u_2 acts as a "data mule." The execution of the plan begins as expected, with u_1 reaching the area of the targets and u_2 staying in radio contact with it in order to receive the pictures of the first two targets (Figure 4a). When u_2 flies back to re-connect with the relays, however, it observes ("Observe" step of the control loop from 2) that the home base is unexpectedly not reachable. Hence, u_2 uses the available observations to determine plausible causes ("Explain" step of the control loop). In this instance, u_2 observes that relays r_5, r_6, r_7 and all the network nodes south of them are not reachable via the network. Based on knowledge of the layout of the network, u_2 determines that the simplest plausible explanation is that those three relays must have stopped working while u_2 was out of radio contact (e.g., started malfunctioning or have been destroyed). As shown in Figure 4a this is indeed the case in our experimental set-up, although it need not be. The planner is capable of operating under the *assumption* that its hypotheses are correct, and later re-evaluate the situation based on further observations, and correct its hypotheses and re-plan if needed.

[2]This example's trajectory used to visit the targets is the same as the previous example. The corresponding plans are equivalent from the point of view of all the metrics, and the specific selection of one over the other is due to randomization used in the search process.

Next, u_2 replans ("Plan" step of the control loop). The plan is created based on the assumption that u_1 will continue executing the mission plan. This assumption can be later withdrawn if observations prove it false. Following the new plan, u_2 moves further South towards the home base (Figure 4b). Simultaneously, u_1 continues with the execution of the mission plan, unaware that the connectivity has changed and that u_2 has deviated from the mission plan. After successfully relaying the pictures to the home base, u_2 moves back towards u_1. UAV u_1, on the other hand, reaches the expected rendezvous point, and observes that u_2 is not where expected (Figure 4c). UAV u_1 does not know the actual position of u_2, but its absence is evidence that u_2 must have deviated from the mission plan at some unknown point in time (or possibly have been destroyed, but for the sake of this example, we disregard destruction of UAVs). Thus, it is now u_1's turn to replan. Not knowing u_2's state, u_1's plan is to fly South to relay the missing picture to the home base on its own. This plan still does not take into account the unavailability of r_5, r_6, r_7, since u_1 has not yet had a chance to get in radio contact with the relays and observe the current network connectivity state. The two UAVs continue with the execution of their new plans and eventually meet, unexpectedly for both (Figure 4d), and automatically share between each other the final picture. Both now determine that the mission can be completed by flying South past the failed relays, and execute the corresponding actions.

Experimental Comparison. The network-aware approach



(a) Step 6: u_2 moves toward relays. Relay nodes 5, 6, and 7 have failed.

(b) Step 7: u_2 re-plans and moves closer to home base.

(c) Step 8: u_2 moves toward u_1.

(d) Step 9: u_2 and u_1 reconnect and move back toward home base.

Fig. 4: Example instance 2 illustrates re-planning after relay node failure between steps 5 and 6 forcing the UAVs to re-plan.

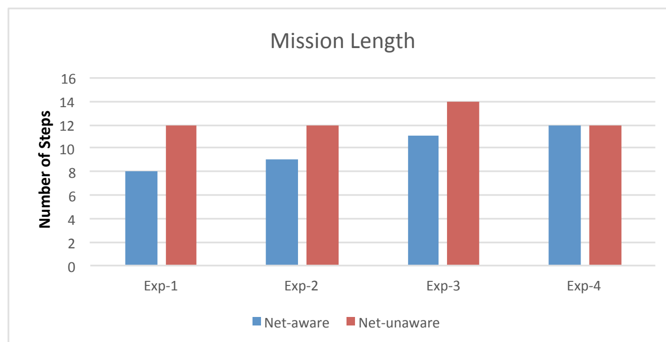
to planning provides advantages over the techniques that either disregard the network, or assume perfect communications. Figure 5 illustrates a quantitative experimental comparison between the network-aware approach and one in which the network is disregarded in terms of mission length and total staleness.[3] The comparison includes the two example instances discussed earlier (labeled Exp-2 and Exp-4). Of the other two experiments, Exp-1 was a variant of Exp-2 that could be solved by with the data-mule positioned in a static location, while Exp-3 was a variant of Exp-2 with 5 targets. As can be seen, the network-aware approach is always superior. In Exp-1, the UAV acting as a data-mule can extend the range of the network so that all the pictures are instantly relayed to the home base, reducing total staleness to 0. In Exp-4, it is worth stressing that the network, which the UAVs rely upon when using our approach, suddenly fails. One would expect the network-unaware approach to have an advantage under these circumstances, but as demonstrated by the experimental results, by identifying the network issues and working around them, our approach still ensures a lower total staleness of the pictures.

V. Simulation

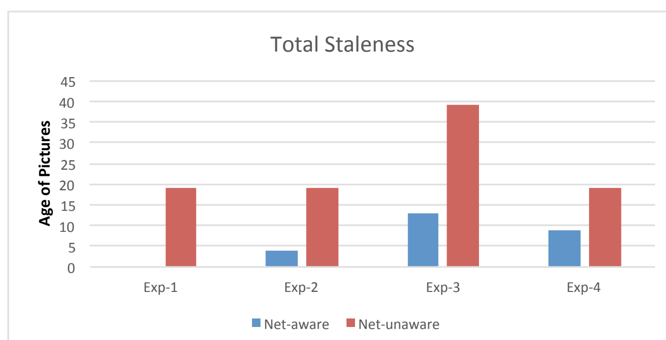
The simulation for the experimental component of this work was built using the Common Open Research Emulator (CORE) [12]. CORE is a real-time network emulator that allows users to create lightweight virtual nodes with full-fledged network communications stack. CORE virtual nodes can run unmodified Linux applications in real-time. The CORE GUI incorporates a basic range-based model to emulate networks typical in mobile ad-hoc network (MANET) environments. CORE provides an interface for creating complex network topologies, node mobility in an environment, and access to the lower-level network conditions, e.g., network connectivity.

Using CORE as a real-time simulation enviornment allows agents, represented as CORE nodes to execute mission plans in

[3]For simplicity we measure mission length and staleness in time steps, but it is not difficult to extend our approach to use action durations.



(a) Length of the mission in time steps for the example instances.



(b) The total staleness of the image transfers.

Fig. 5: Performance comparison.

realistic radio environments. For this work, CORE router nodes represent the home base, relays, and UAVs. The nodes are interconnected via an ad-hoc wireless network. As the UAVs move in the environment, CORE updates the connectivity between other UAVs and relays based on the range dictated by the built-in wireless model. The radio network model has limited range and bandwidth capacity. Each node is running the Optimized Link-State Routing protocol (OLSR) [23], a unicast MANET routing algorithm, that maintains the routing tables across the nodes. Maintaining the routing table establishes if a UAV can reach the home base at any given moment. Using CORE as the simulator allows us to account for realistic communications in ways not possible with multi-agent simulators such as AgentFly [11].

VI. Conclusion and Future Work

This paper presented a novel application of a network-aware planner and an intelligent plan-aware network layer to the problem of UAV coordination. The UAV scenarios considered in this paper are bound to be increasingly common as more levels autonomy are required to create large-scale systems. Prior work on distributed coordination and planning has mostly overlooked or simplified communications dynamics, at best treating communications as a resource or other planning constraint. Similarly, the effects of the plan on the communications have not been studied in details.

Our work features an end-to-end integration of AI planning with simple but more realistic communication models, and demonstrates the reliability and performance gains deriving from it. Our experimental evaluation approach yielded a reduction in mission length of up to 30% and in total staleness between 50% and 100%. We expect that, in more complex scenarios, the advantage of a realistic networking model will be even more evident. In our experiments, execution time was always satisfactory, and we believe that several techniques from the state-of-the-art can be applied to curb the increase in execution times as the scenarios become more complex. Because the reasoning is at a high level of abstraction (e.g. at the level of discrete waypoints rather than detailed navigation), the search space is contained, making this approach viable for UAV platforms. The communication model is simple, when the model deviates too much, the UAV's ability to cope with unexpected circumstances will allow them to continue mission execution.

For the future, we intend to expand the plan-aware networking layer with reasoning capabilities, integrate network-aware planning and plan-aware networking more tightly, and design and execute experiments demonstrating the advantages of such a tighter integration. In plan-aware networking, we take as input the mission plan and knowledge of past and present network states. The network layer uses this information at each moment to infer current needs and to make networking decisions (e.g., routing, QoS) that are in the best interests of the mission. Additionally, there is a need for more robust experimental evaluations. The focus of the planner and evaluation was on the planning algorithms that accounted for network communications state. CORE only provides realistic network communications. More robust planning and approaches require realistic UAV systems models to incorporate into the planning algorithms.

References

[1] K. Usbeck, J. Cleveland, and W. C. Regli, "Network-centric ied detection planning," *IJIDSS*, vol. 5, no. 1, pp. 44–74, 2012.

[2] A. N. Kopeikin, S. S. Ponda, L. B. Johnson, and J. P. How, "Dynamic Mission Planning for Communication Control in Multiple Unmanned Aircraft Teams," *Unmanned Systems*, vol. 1, no. 1, pp. 41–58, 2013.

[3] M. Balduccini and M. Gelfond, "Diagnostic reasoning with A-Prolog," *Journal of Theory and Practice of Logic Programming (TPLP)*, vol. 3, no. 4–5, pp. 425–461, Jul 2003.

[4] A. S. Rao and M. P. Georgeff, "Modeling Rational Agents within a BDI-Architecture," in *Proc. of the Int'l Conf. on Principles of Knowledge Representation and Reasoning*, 1991.

[5] M. Wooldridge, *Reasoning about Rational Agents*. MIT Press, 2000.

[6] J. Blount, M. Gelfond, and M. Balduccini, "Towards a Theory of Intentional Agents," in *Knowledge Representation and Reasoning in Robotics*, ser. AAAI Spring Symp. Series, 2014.

[7] D. V. Pynadath and M. Tambe, "The Communicative Multiagent Team Decision Problem: Analyzing Teamwork Theories and Models," *JAIR*, vol. 16, pp. 389–423, 2002.

[8] M. Balduccini, M. Gelfond, and M. Nogueira, "Answer Set Based Design of Knowledge Systems," *Annals of Mathematics and Artificial Intelligence*, vol. 47, no. 1–2, pp. 183–219, 2006.

[9] C. Sakama and T. C. Son, "Negotiation Using Logic Programming with Consistency Restoring Rules," in *Proc. of the Int'l Joint Conf. on Artificial Intelligence*, 2009.

[10] G. Grasso, N. Leone, M. Manna, and F. Ricca, "ASP at Work: Spin-off and Applications of the DLV System," in *Symp. on Constructive Mathematics in Computer Science*, Oct 2010.

[11] S. K. David Sislak, Premysl Volf and M. Pechoucek, *AgentFly: Scalable, High-Fidelity Framework for Simulation, Planning and Collision Avoidance of Multiple UAVs*. Wiley Inc., 2012, ch. 9, pp. 235–264.

[12] J. Ahrenholz, "Comparison of CORE network emulation platforms," in *IEEE Military Communications Conf.*, November 2010.

[13] M. Balduccini and M. Gelfond, "The AAA Architecture: An Overview," in *AAAI Spring Symp.: Architectures for Intelligent Theory-Based Agents*, Mar 2008.

[14] C. Baral and M. Gelfond, "Reasoning Agents In Dynamic Domains," in *Workshop on Logic-Based Artificial Intelligence*. Kluwer Academic Publishers, Jun 2000, pp. 257–279.

[15] J. McCarthy, "Elaboration Tolerance," 1998.

[16] M. Gelfond and V. Lifschitz, "Classical Negation in Logic Programs and Disjunctive Databases," *New Generation Computing*, vol. 9, pp. 365–385, 1991.

[17] C. Baral, *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, Jan 2003.

[18] M. Gebser, B. Kaufmann, and T. Schaub, "The Conflict-Driven Answer Set Solver clasp: Progress Report," in *Logic Programming and Nonmonotonic Reasoning*, Sep 2009.

[19] M. Balduccini, W. C. Regli, and D. N. Nguyen, "Towards an ASP-Based Architecture for Autonomous UAVs in Dynamic Environments (Extended Abstract)," in *30th International Conference on Logic Programming (ICLP14)*, 2014.

[20] ——, "An ASP-Based Architecture for Autonomous UAVs in Dynamic Environments: Progress Report," in *Proceedings of the 15th International Workshop on Non-Monotonic Reasoning (NMR 2014)*, S. Konieczny and H. Tompits, Eds., 2014.

[21] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, 2003.

[22] D. Jea, A. Somasundara, and M. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," *Distr. Computing in Sensor Sys.*, 2005.

[23] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *IEEE INMIC: Technology for the 21st Century*, 2001.