3-11-2011

Routing UAVs to Co-Optimize Mission Effectiveness and Network Performance with Dynamic Programming

Spenser D. Lee

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Digital Communications and Networking Commons, and the Systems and Communications Commons

Recommended Citation

ROUTING UAVS TO CO-OPTIMIZE MISSION EFFECTIVENESS AND NETWORK

PERFORMANCE WITH DYNAMIC PROGRAMMING

THESIS

Spenser D. Lee, Second Lieutenant, USAF

AFIT/GCS/ENG/11-04

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/11-04

ROUTING UAVS TO CO-OPTIMIZE MISSION EFFECTIVENESS AND NETWORK PERFORMANCE WITH DYNAMIC PROGRAMMING

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Spenser D. Lee, BS

Second Lieutenant, USAF

March 2011

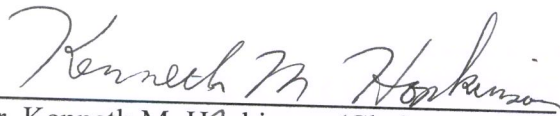AFIT/GCS/ENG/11-04

ROUTING UAVS TO CO-OPTIMIZE MISSION EFFECTIVENESS AND NETWORK PERFORMANCE WITH DYNAMIC PROGRAMMING
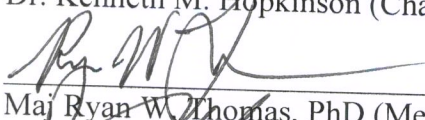
Spenser D. Lee, BS

Second Lieutenant, USAF
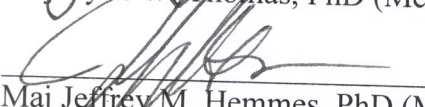
Approved:

_____
Dr. Kenneth M. Hopkinson (Chairman)

10 MAR 11
_____
Date

_____
Maj Ryan W. Thomas, PhD (Member)

10 MAR 11
_____
Date

_____
Maj Jeffrey M. Hemmes, PhD (Member)

10 MAR 11
_____
Date

Abstract

In support of the Air Force Research Laboratory's (AFRL) vision of the layered sensing operations center, command and control intelligence surveillance and reconnaissance (C2ISR) more focus must be placed on architectures that support information systems, rather than just the information systems themselves. By extending the role of UAVs beyond simply intelligence, surveillance, and reconnaissance (ISR) operations and into a dual-role with networking operations we can better utilize our information assets. To achieve the goal of dual-role UAVs, a concrete approach to planning must be taken. This research defines a mathematical model and a non-trivial deterministic algorithmic approach to determining UAV placement to support ad-hoc network capability, while maintaining the valuable service of surveillance activities.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor and committee, Dr. Kenneth Hopkinson, Major Ryan Thomas, and Major Jeffrey Hemmes, for their guidance and support throughout the course of this thesis effort. I would, also, like to thank my sponsors at RYAA, Mr. Bruce Preiss, 1Lt Matt Lenzo, and Mr. Matthew McClure, for both the support and latitude provided to me in this endeavor. I would like to thank everyone in the Cyber Animal Lab for their assistance and for setting a high standard of excellence. And finally, I would like to thank my loving fiancée, Rebecca Morgan, for her support and patience over the two years it took to complete this research.

Spenser D. Lee

.

Table of Contents

List of Figures

List of Tables

ROUTING UAVS TO CO-OPTIMIZE MISSION EFFECTIVENESS AND NETWORK

PERFORMANCE WITH DYNAMIC PROGRAMMING

I. Introduction

1.1 Research Motivation

The motivation of this research is to determine a strategy to deploy and route

unmanned aerial systems (UASs) to take full advantage of their surveillance and network

routing capabilties. Though historically UAS have been used solely for intelligence,

surveillance, and reconaissance (ISR) purposes, recent developments have augmented

their role with network routing. An emerging dual-role in network routing and

surveillance seems fitting because the vast majority of network traffic for these deployed

systems is either ISR-related or command-and-control (C2) related. The high-tempo

environment in which military forces now operate demands fast response to significant

events, and thus, command, control, intelligence, surveillance, and reconaissance

(C2ISR) streamlines the decision-making process. Determining policies to govern UAS

placement at a tactical level helps to maximize their mission effectiveness, cementing

their strategic importance.

Large military operations that demand high situational awareness have fostered heavy

reliance on large UAV platforms, such at the General Atomics MQ-1 Predator, instead of

smaller platforms, such as the RQ-11B Raven, and because of the Predator's ubiquity, it

has become natural to deploy them with line-of-sight (LOS) communications equipment,

such as the Harris Co. AN/PRC-117G, to support ground troops with ad-hoc network

routing capability as an alternative to satellite communication (SATCOM). Saturation of

SATCOM links strains routers with a deluge of information, hosted by a myriad of different services, confounded with high overheads, and ultimately increasing end-to-end latency with long queues at the routers and dropped packets. LOS networking capabilites alleviate this problem by circumventing low-priority traffic from constrained, high-value communication links, such as SATCOM.

Large unmanned aerial vehicles (UAVs) are ideal for the dual-role of surveillance and network routing because they have high endurance, allowing for high persistence, and a high operating altitude, allowing for a long LOS, effectively shortening the list of "hops" a given packet must take to reach its destination. However, large UAVs are expensive to operate. Aside from heavy fuel consumption, they require entire maintenance crews, complex onboard and ground-based electronics, SATCOM support, and remote human operators.

In contrast, small platforms, such as the RQ-11B Raven are hand-launched and require little more than a hand-carried remote control. Despite their low payload capacity, ongoing research is being conducted by multiple agencies to investigate the plausibility of their design, their potential mission effectiveness, and policies that govern their use. Though small UAVs have distinct disadvantages to large UAVs, and may never replace the role of those large platforms, their potential operational improvement lies in their low cost, ease-of-use, rapid deployability, and provision of increased situational awareness.

1.2 Problem Statement

Given a particular area of responsibility, with fixed nodes, moving high value targets, fixed traffic demand (based on the expected volume of routine C2 traffic from ground

2

troops), and a set of small UAVs equipped with network routers and known, fixed communication range, determine a schedule of placement for each UAS, such that the mission effectiveness, represented by UAS coverage of the high-value targets (HVTs), and network performance, represented by the network topology's ability to satisfy network demands, of the entire system are co-optimized.

1.3 Scope

The problem will be constrained by reasonable assumptions that compromise between a problem space that is manageable and maintaining the reality of the mission to keep the results relevant to operational use. In this case, simplifications to the problem space will maintain the algorithm's ability to predict real-world UAS placement within enough range to conduct normal network operations.

The notional UAVs in this scenario will be RQ-11B Ravens equipped with a radio repeater that flies at relatively low altitudes. Though many small UAVs are suitable for this mission, the Raven is the most widely used UAV of its size and abilities. Ongoing research is developing its ability to both perform surveillance and relay network traffic using an onboard router. Although small UAVs have the ability to fly as high as 10,000 feet, their strategic advantage of high-fidelity surveillance information is highly diminished at this altitude. An assumption of low-altitude gives better assurance that the information it collects is of high tactical value, and reflects a likely real-world tasking.

The model of signal propagation is abstracted from the problem space. Although real communication assurance depends on much more than simply Euclidean distance, the assumption of "line of sight" simplifies the process of determining whether or not a

signal is received by avoiding the dynamics of signal attenuation, physical obstacles, weather, and similar realities related to physical signal reliability.

To that end, the spatial model is simplified even further. Instead of dealing with continuous space, the geographic map is discretized into non-overlapping, equally-sized blocks. The blocks are of a parameterized size. For example, the principal scenario uses a map of a 10 km by 10 km area. Given a parameter of 0.25 km square blocks, the problem space would consist of a 40 by 40 block area of distinct spaces. This simplification provides the algorithm with a more manageable, organized space from which to choose, and aligns closely with the manageable goal of finding loiter-positions, rather than the goal of finding exact locations.

Real world nodes that travel slowly in comparison to airborne UASs will be modeled as fixed nodes. For example, mine resistant ambush protected (MRAP) vehicles, tanks, and ground troops will be fixed nodes because, within a given information transmitting opportunity of a few minutes or so, these entities move minimally, at least within the context of network range and connectivity. Modeling these relatively slow entities as fixed should not interfere strongly with the reality of network routing because the amount of time it takes for a network packet to travel through hops to the destination is a figurative "blink of the eye" in comparison to the speed of a humvee, which actually spends a lot of time at a standstill or loitering within a small geographic space. UAVs, on the other hand, are unaffected by the position of buildings or constraint of roadways and simply fly to the location at which they are needed.

1.4 Approach and Concept of Operations

The proposed combat preparation and tasking approach is derived from the concept of operations (CONOPS) from the Davis and Kabban research [1]. Davis and Kabban outline a sequence of actions necessary to implement a layered sensing (LS) architecture that remains fairly open. Refining this process with more strict activities will allow multiple operators to use their collective efforts to achieve a more complete common operating picture (COP) to military commanders and decision makers.

The CONOPS is a detailed document that:

> …provides an overview of the principle LS components and capabilities, the expected operational environment, notional missions, architecture, necessary and enabling capabilities, sequenced actions, and LS-specific scenarios and vignettes

While this document considers a wide range of different missions in which ISR operations are needed, it leaves out tactical details. According to the CONOPS the overall LS system must:

- Acquire, sort, prioritize, and display accurate, uncorrupted, actionable and timely data or information to humans and/or machines.

- Avoid centralized decision flow in favor of flexible communications to, from, and between decision makers at all levels of war.

- Produce and/or display and share a common joint operating picture to facilitate improved interoperability, communication, and situational awareness.

The LS CONOPS requires that the implementation of an LS architecture, to include this network-level structure, be capable of performing these tasks, among others. These tasks

are relevant to this research because they emphasize the importance of acquiring timely, accurate information over a flexible communication network. Effective network-level architecture should be able to account for future traffic, adjust its structure appropriately, and transport to military decision makers the data in a timely fashion.

For this research, supporting these goals means tasking UAVs to be in locations where they are best suited to observe critical events, and also to be in locations where they support the group goal of creating an ad-hoc mesh network. The ad-hoc mesh network allows surveillance information originating from UAVs and routine GIG traffic from ground nodes to "hop" to centralized locations where commanders are located, such as forward operating bases and command centers.

1.4.1 Task Elaboration

The CONOPS includes a sequence of actions which addresses the full process, from addressing the circumstances to making decisions based on them. The sequence of actions defined in the CONOPS is as follows:

1. Receive collection of requirements

2. Prioritize collection requirements

3. Assign platforms and sensors to prioritized collection requirements

4. Task operational units to perform ISR activities

5. Deploy and position assets to execute taskings (land, air, space, sea, cyber)

6. Command, control, and communicate

7. Sense

8. Exploit

9. Track

10. Redeploy assets

This research does not define every single one of these steps, but rather, focuses on a subset. Receiving and prioritizing information are tasks that are conceptually separate from tasking assets. Sensing generally occurs at the hardware level. Exploiting data and tracking targets are tasks that are difficult for machines to perform and are best performed by humans. Mainly, this approach looks to perform steps 3, 4 and 5. At these steps in the process, focus switches to the physical placement of assets, including UAVs, which has a direct impact on the coverage of events and the routing of network traffic.

1.4.2 Matching the Correct Assets to the Situation

To mitigate the problem of "information overload," both with physical bandwidth constraints and human comprehension capacity, this approach focuses on streamlining the data collection model. The past Davis and Kabban research considered a10km^2 urban area (more information in I.A.Appendix C:),, and called for the following extensive list of hardware: 1 E-8C (MTI), 1 RQ-4 (MTI, SAR/EO), 2 ACS (MTI), Multiple UGSs, 2 Shadows (Video/IR), 2 Hunters (Video/IR), 3 MQ-1 Predators (Video/IR) [1]. Many of the aforementioned platforms rely on SATCOM to some degree and the addition of more devices only exacerbates the problem.

SATCOM, although often a safe choice for BLOS communication, can occasionally fail due to weather or physical obstructions. The increasing number of devices that rely on SATCOM has caused high saturation of the links to those satellites, which causes heavy delays for the end-user. A relief from the reliance on these SATCOM links will help to allow these pathways to be open for extremely critical services.

The proposed model calls for a change in roles to achieve mission compliance with fewer devices. The Raven's ISR mission has generally been limited to a "mirror around the corner" role for ground troops, due to its limited flight time, high expendability, and lack of data-transfer ability. By adding a small network router, the expectation is for the Raven to be able to collect Video/IR data and ferry it to a semi-persistent data ferry, such as a loitering MQ-1 Predator. Figure 1 shows the operational concept that incorporates this strategy.
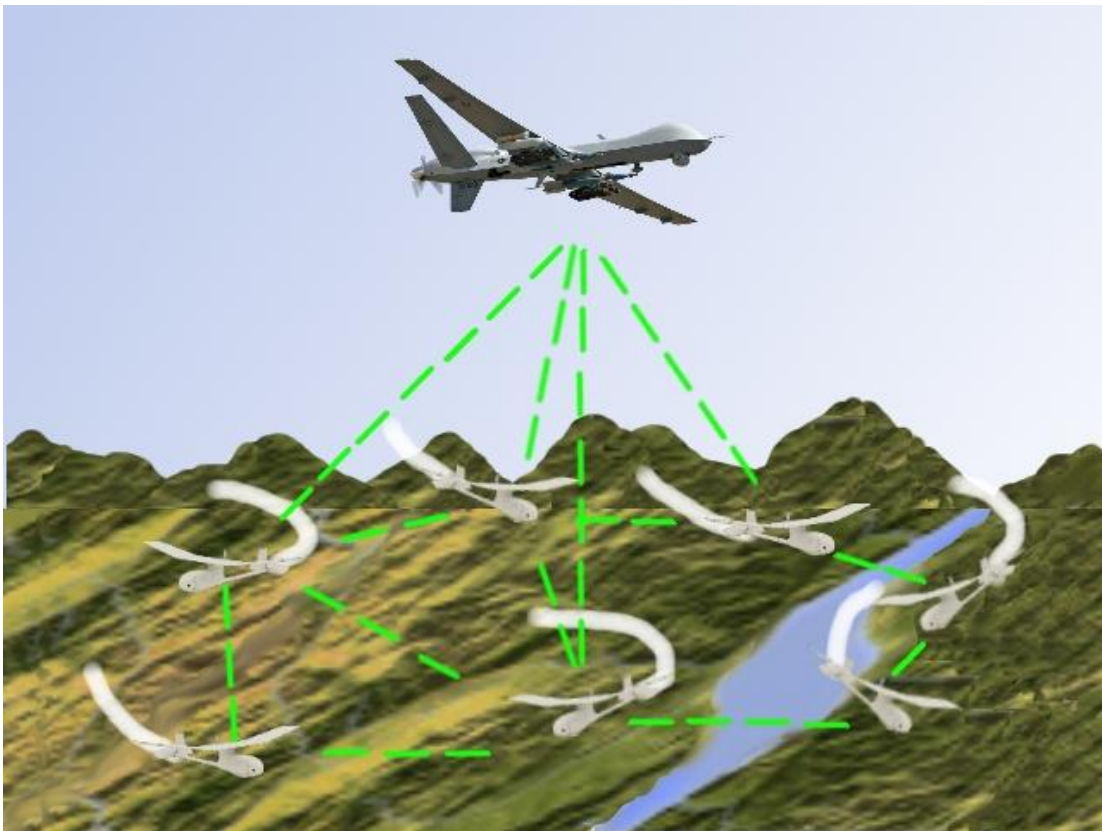
Figure 1: Operational Concept

The proposal calls for a small swarm of Ravens (roughly 5-10, depending on the size of the area and the characteristics of the terrain) to collect the video/IR data at low altitude, pass it along to the RQ-1 overhead, using a nearby E-8C JSTARS as a command and control center. Predators, Shadows, and Hunters had been previously used to

monitor the ground at a conservatively high altitude to avoid the danger of RPG attacks. Small-platform UAVs, such as the Raven are relatively inexpensive in comparison and more expendable, thus allowing them to be flown at lower altitude. Lower altitude platforms provide the opportunity to use lower-fidelity cameras to achieve the same benefit of surveillance. Platforms at 30,000 feet, like the Predator, require very expensive cameras to be able to zoom in on a particular event. Within this role, the overhead Predator can zoom out, view the entire urban area, give appropriate flight taskings to the Ravens below, ferry data to the command center, and forward mission taskings and plan changes from the E-8C JSTARS to the Raven operators as necessary.

Previous C2 centralization at a forward operating base (FOB) had created the requirement for communication over long distances, which resulted in heavy SATCOM usage. A slightly more decentralized approach reduces the volume of traffic traveling over SATCOM links, reducing strain, and increasing potential bandwidth for other services. Of course, not all mission decisions can be made at such a decentralized, tactical level, and some messages will have to travel in between the E-C8 and the FOB, but the expectation is that most decisions can be made in a decentralized manner. Although this decentralized architecture increases the latency of high-importance messages traveling from the area of operations and the FOB, most routine messages will have a decreased latency and the overall responsiveness of the system will increase.

1.5 Goals and Hypothesis

The ultimate goal of this research is to determine a strategy to deploy UASs in a way that takes full advantage of their current surveillance and networking capabilties. In particular, this research focuses on increasing the strategic advantage of small UAVs by

scheduling their movement to areas that weigh their benefit when observing significant

events, such as the activity of HVTs, and when performing a network routing role. The

notional small platform is the RQ-11B Raven (pictured in Figure 2 [2]Figure 12) fixed

with an onboard radio signal repeater. The platform offers the advantage of collecting

higher fidelity imaging than larger, higher-altitude UAS platforms because of their

proximity to the battlespace, but this research focuses on the possiblity of using the

Raven in a dual-role. The key factor about the use of this platform is that the concept of

operations is designed for technology that we have today and is designed to be

implemented in the attainable future.

Figure 2 OWL UAV, converted from RQ-11B Ravens

The experiments will attempt to prove the hypotheses:

1. Using a different balance of information-gathering platforms shifted more
   heavily towards small-platform UAVs allows for higher fidelity imaging of
   significant events.

2. An ad-hoc network of fixed ground nodes and UAVs is capable of supporting C2 and ISR traffic under real operational conditions.

3. Using a polynomial-time algorithm, an ad-hoc network of UAVs can determine routing that provides near-optimally for both their primary surveillance mission and their secondary ad-hoc network formation.

The overall impact of this research will be an alternative to the current CONOPS for both combat C2 and real-time ISR in which network performance is improved and ISR is more streamlined and highly reactive to ongoing events. C2 and ISR network traffic will have an alternative link to SATCOM and video/IR data will have greater potential focus on mission targets.

1.6 Overview and Significance of Work

The existing CONOPS for the usage of UAVs tasked single UAVs with loitering over a given area for a long period of time to observe an event that had a large, uncertain window of opportunity. Often, it would be the case that the services of this UAV were needed a location nearby and it would be re-tasked to temporarily attend to ancillary tasks. In a sense, each UAV had a singular task with the ability to deviate if secondary tasks did not conflict with the primary task.

Changing the model shifts the conceptual approach from one UAV to one task (or, at best: a few), to a relatively large group of UAVs, in concert with ground units, to many tasks. The flexibility of this approach allows for fault tolerance from platform failure and individual refueling stoppage time, as well as its ability to pick up on unexpected events because of its wide geographic coverage. In contrast, the revised approach may cause difficulty in assigning tasks effectively and quickly or in covering a small geographic

11

area with a high level of activity (clustering). The major tradeoff is coverage versus detail.

The tactical plan changes the operational requirements of an urban engagement. Heavy usage of large UAVs introduces reliance on SATCOM and more stakeholders who are outside of the area of responsibility. Large UAVs themselves can be expensive to operate, when considering the cost of a remote user, a ground maintenance crew, and the high cost of fuel. SATCOM communications can be mercurial due to its variability from weather, terrain, and saturation. Large UAV tasking also tends present less flexible tasking. An elaborated approach with specific consideration to assigning small UAVs and tasking information assets helps to utilize all combat and intelligence assets to their full potential, while helping to drive down cost and free up the valuable commodity of bandwidth for other services.

II. Background

This chapter discusses the driving forces behind ad-hoc routing and improved combat data collection. This chapter elaborates on the specific needs of the Air Force and the Department of Defense to improve, not only agency-specific operational policies, but Air Force wide and joint operations. This chapter further discusses currently fielded technologies that support C2ISR and the underlying mechanics. The discussion of alternative solutions expands the current state-of-the-art in ad-hoc UAV routing and discusses philosophies governing their design.

2.1 Air Force Combat Operations and the role of C2ISR

The current unifying theme of research and innovation in command control systems is that net-centric warfare (NCW) and enterprise command and control (EC2) systems [3] [4] are important to creating a common operating picture (COP) to provide situational awareness to the war fighter. A highly capable information system that employs NCW can deliver the COP to troops in a timely manner and increase their mission effectiveness. Current operations of joint military forces depend on this powerful capability to collect, analyze, and act upon information. This battlefield information, as a baseline, must contain movement data for hostile, neutral, and friendly forces, environmental information, battle plans, and processed and raw information and imagery [5]. The requirements that joint C2 centers hold for battle-space awareness, in a realistic sense, place a heavy strain on information infrastructure systems. With the growing need for deployable, expeditionary forces, information gathering and dissemination is becoming a difficult challenge. At a highly conceptual level, high demand for timely

accurate information competes directly with the expeditionary model of war-fighting, which dictates a decrease in forces. Surveillance coverage simply cannot live up to its demand with fewer troops. Novel systems must be put in place to satisfy the demands of NCW.

The Department of Defense's Force Transformation Office released "The Implementation of Network-Centric Warfare" in 2005 [6], stating the importance of information on today's battlefield. The document points out, in the Information Age, the focus is not set entirely on the technology of our information systems, but rather, the interaction that humans have with those information systems. Much of the fast-acting critical thinking is still done by human operators who are trying to gain as much information from our systems as possible.
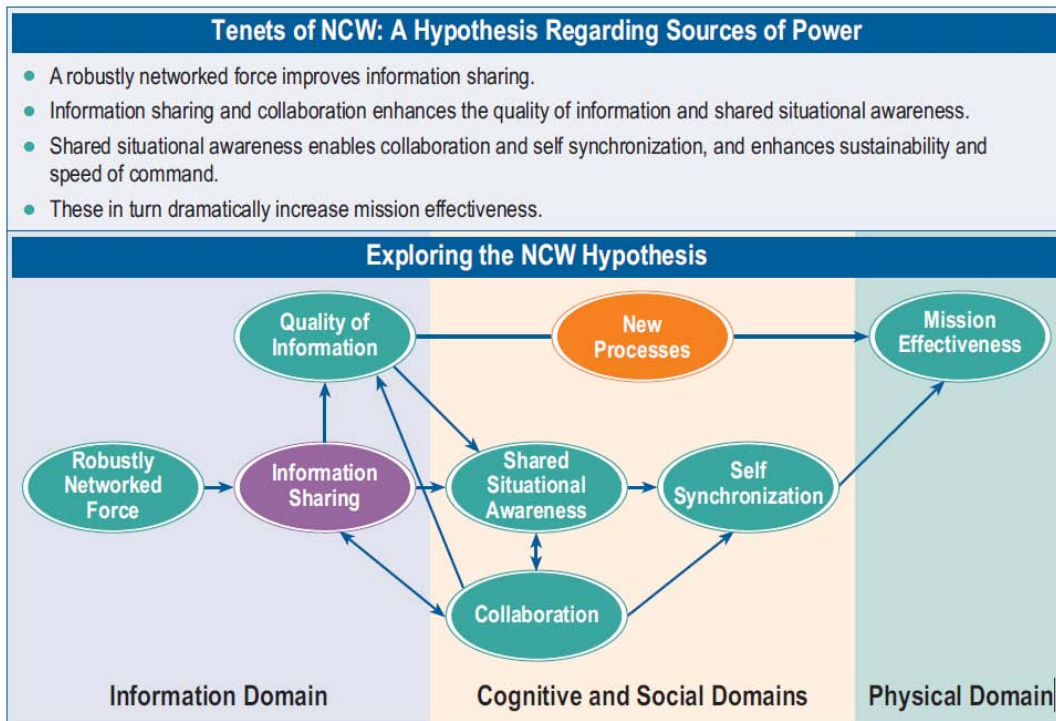
Figure 3: Tenets of Network-Centric Warfare [6]

Net-centric doctrine encapsulates the idea of ad-hoc routing for network stability. Figure 3 shows the relationship between the information domain, the cognitive and social domains, and the physical domain. The ability to effectively share information allows for better collaboration, and increases situational awareness, which ultimately leads to greater mission effectiveness. The techniques developed in this research improve our capabilities in both the information domain and the physical domain to help shorten the gap between the realization of events and our ability to take appropriate courses of action.

2.2 The Global Information Grid

According to Department of Defense (DoD)directive 8001.1 the Global Information Grid (GIG) is the globally interconnected end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating and managing information on demand to warfighters, policy makers, and support personnel [7]. The GIG includes all operating locations, including bases, posts, camps, stations, facilities, mobile platforms, and deployed sites. The overarching policy requires that the GIG support all missions involving information technology (IT) and that it be interoperable. The DoD vision is that a single architecture be capable of managing all information assets in support of information superiority.

The ultimate goal of the GIG is a large undertaking, and currently, there is still improvement t be made. In the architectural vision published in 2007 the DoD outlined specifically the need for robust mobile systems. According to the vision, "the current GIG is static rather than dynamic; it cannot quickly adapt to satisfy the unanticipated needs and users." [8] The main focus of current improvements on the GIG is to "support NCO." [8] The architectural vision describes the shortcomings of currently fielded

systems, and describes capabilities that need to be improved. In the battlespace

environment the GIG facilitates the passage of information between "forces, facilities,

sensors, decision makers (at all levels), weapons, intelligence analysts, [and] support

personnel." The architectural vision places high emphasis on high availability of mission

critical information and performance control, such as bandwidth throttling. The target

GIG is described as the following:

> Users can rapidly access the GIG network and remain connected (i.e., be
> automatically authenticated, recognized, and responded to) as they deploy
> and move. Even at the tactical 'edge,' users have access to sufficient
> bandwidth, that, coupled with network optimization techniques –
> including information caching and performance management – enables
> those users to 'pull' or 'post' important bandwidth intensive information
> such as high-resolution video with acceptable latency. When connections
> are interrupted and resources constrained, the GIG dynamically adapts
> service levels (including data compression) and communication paths on a
> user-priority and precedence basis that optimizes mission assurance.

Though the architectural view is a high-level design and does not describe specific

implementations needed to accomplish the "target GIG," the architecture must

dynamically adapt and support high volumes of traffic, such as high-resolution video.

Not only does the vision call for dynamic bandwidth reallocation, but includes specific

provision for mobile ad hoc networks.

2.3 Mobile-ad Hoc Networks

A mobile ad hoc network (MANET) is a wireless communication system in which the

nodes work together based only on a mutual agreement without knowing about the

network topology around themselves [9]. The general characteristics of MANETs are the

"multi-hop nature of connectivity between the nodes" and "highly time-varying

topology." [10] MANETs do not rely on any infrastructure or static devices, such as base

stations. The driving forces behind the design of MANET protocols include mitigating

high power consumption, using less bandwidth, and lowering error rates [11]. MANET

protocols that vary by adaptive routing or topology control differ to make tradeoffs and

achieve these network performance goals based on particular domain characteristics.

2.3.1 Table-Driven

Table-driven MANET protocols require each node to maintain one or more tables that

contain information about the topology of the network, including routing information,

distance, and link cost. Nodes disseminate messages to each other to announce changes

to the topology. Source-initiated MANET protocols require that nodes create routes as

the messages are being sent. In general, table-driven protocols have higher memory

requirements because of the need for each node to maintain a routing table. Source-

initiated protocols generally use more bandwidth because of the need for more frequent

route discovery.

The archetypical table-driven protocol is Destination-Sequence Distance-Vector

(DSDV) routing, which implements a Bellman-Ford routing mechanism [11]. Each node

maintains a routing table containing the next hop and number of hops to each other node

in the network, using periodic update messages to maintain consistency. Changes to the

topology are broadcasted with either an incremental or a full dump packet. Table-driven

protocols generally require fewer broadcasts than source-initiated protocols, and are thus

more bandwidth and energy efficient, but at the cost of high memory usage of the device,

making the system less scalable, although this memory usage problem can be mitigated

with a clustering scheme. At low topology-change update rates table-driven protocols are

also less resilient to the mobility of nodes. High node mobility invalidates table entries.

2.3.2 Source Initiated

Source-initiated, or on-demand, protocols generally do not require tables, and rely on messages to eventually propagate to their intended destination. The archetypical source-initiated protocol is Ad Hoc On-Demand Distance Vector (AODV) routing. A source attempting to send a packet first broadcasts a route request (RREQ) packet, containing the destination node's IP address, to its neighbors, who then pass the RREQ to their neighbors. Intermediate nodes note the sender of each unique RREQ, ignoring any duplicate RREQs. Once the RREQ reaches its destination, the destination node unicasts a route reply back to its sender. The backwards path is now established for the transmission of the message. Each node keeps a record of successful RREQ/RREP-defined paths, deleting stale entries by a timer [11]. AODV has low overhead, compared to other source-initiated protocols, because request packets need not append path links along the way. Intermediate nodes need only remember each RREQ's sender. AODV is, however, less resilient to mobile nodes because routes quickly become stale and merit RREQ broadcasts. Source-initiated protocols are memory efficient, but have high potential for high bandwidth overhead.

2.3.3 Use of MANETs in a Battle Environment

Command and control communication for forward-operating troops is a difficult task, requiring the use of a wide range of hardware, software, policies, and operational plans. Often, the main pathway of communication for ground troops is a satellite link, which is reliable in many cases, but can be adversely affected by poor weather conditions, extreme terrain, or loss of line of sight with the physical satellite. Current operations in Afghanistan present these communication challenges frequently and create an

18

opportunity for MANETs to improve communication reliability. Common practice dictates that MANETs not be used to cover an entire operational area to maximize spatial reuse and battery conservation [10], and thus, an entire system must exist to allocate the resources necessary to support operations in a specific area. MANET protocols can support these regional operations at the tactical level.

2.4 Currently Fielded C2ISR Tools

The DoD currently fields a variety of C2ISR tools to provide situational awareness (SA), increase the speed of command, and support dynamic planning and redirection. Although the DoD is currently engaged in an effort to streamline the C2ISR process by unifying these tools and imposing strict standards on massage transfer protocols, the current state of information gathering is through disparate, independent systems. In general, these information systems assist the war-fighter by displaying information in a human readable format, adding context by combining data sources, and facilitating the passage of data by implementing a computer-readable message format.

2.4.1. Falcon View

FalconView is a commercial off-the-shelf (COTS) GIS-enabled visualization tool that interfaces with external tools to overlay information about the environment onto a map. When it was created in 1990, it was originally intended to display geographical data to the warfighter using a simple, easy-to-use human interface. It was adopted by the Air Force in 1997 for mission/flight planning and is currently being used by the Army, Navy, and Coast Guard, as well [12]. The GIS standard employs a query-based data retrieval system, thus allowing the warfighter to obtain information such as: "find all the sensitive (non-combat structures) near a target of interest", "find areas where a route passes within

a danger zone", "analyze a drop-zone to ensure the terrain is suitable." [12] Though interoperable web standards exist for GIS communication, the DoD has yet to adopt them, but will likely evolve this tool to be NCO compliant.

Figure 4 shows an example of how GIS tools in FalconView can immediately have more meaning by being coupled with a map overlay. In this case, FalconView displays weather information, which is fairly commonplace. However, FalconView is capable of displaying vehicle overlays for mission planning, satellite imagery, and infrastructure overlays.
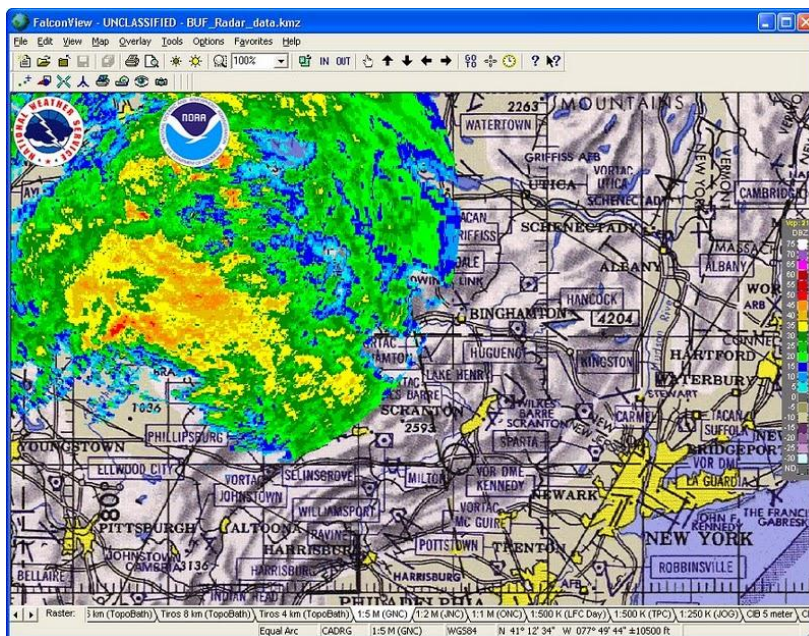
Figure 4: FalconView Interface Supporting GIS overlays [13]

2.4.2 Command Post of the Future

The Command Post of the Future is a US Army project that seeks to decentralize command and control while providing a forum to share information and collaborate on decision making through multiple echelons. That is, to allow commanders at all levels to communicate their intent and their knowledge. It was specifically developed to enable

distributed, collaborative, command and control, rather than simply allowing applications

to share information [14]. The CPOF is currently fielded in approximately 6,000

different units providing situational awareness and a shared COP with shared

workspaces, tools, data, and maps [15]. Where many of the existing information analysis

tools focus on sharing information, this tool focuses on improving command and

allowing actors, who are widely separated geographically, to make fast decisions based

on real-time information from the battlespace.
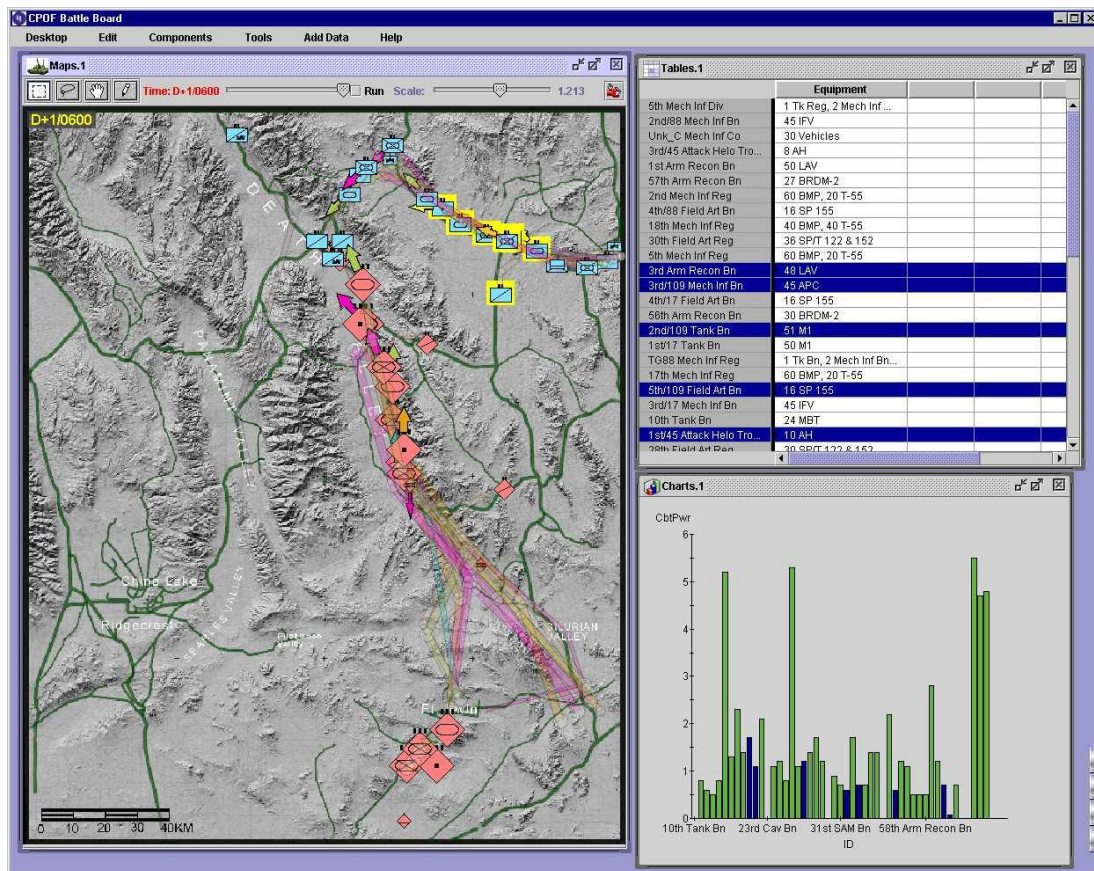
Figure 5: CPoF workspace example
The sophistication of Falcon View and the CPoF exhibit the level of sophistication with

which modern information systems are built. The level of detail given to military

commanders and troops alike is a testament to the large amount of information that is

passed on a normal, everyday basis. The sophistication of the system and the problem of network load are compounded by the fact that these previously mentioned systems are used in concert with Force XXI Battle Command Brigade and Below (FBCB2) [16], Joint Tactical Radio System (JTRS), Advanced Field Artillery Tactical Data System (AFATADS), and Maneuver Control System (MCS). As joint operations become increasingly more common, single vehicles travel with multiple systems in them, in order to share and disseminate information between forces of different military services. The direct implication of this pattern of information-use is that the amount of network traffic is magnified. It is not an uncommon practice for troops to report an event multiple times over multiple C2 systems, in order to coordinate actions with disparate forward-operating groups of blue forces. Until separate military branches and other information-gathering organizations can agree upon a single system that serves the needs of all stakeholders, the trend of large, unwieldy information systems will continue. Communication infrastructure needs to continue to grow to support this burgeoning system.

An operational assessment of a currently deployed multiband communication device confirmed the use of these services and underlined their importance to supporting joint operations [17]. The study also revealed the delays in network communication that result from ad-hoc routing. A task as simple as opening an email from the secret network took over 2 minutes and sending 2.29MB of data took nearly 10 minutes [17]. It is easily conceivable that the inability to obtain critical information within this timeframe could seriously endanger forward-operating forces. The delays in the current state-of-the-art communication infrastructure indicate that the capabilities of software used by military forces is advancing at a rate that is cannot be supported by traditional methods of ad-hoc

communication. A system capable of supporting the high volume of traffic created by modern military operations must take special consideration to the circumstances of the current battlespace environment in order to overcome high queuing delay, dropped packets, and bandwidth overloading.

2.5 Data Collection and Exploitation

The operational activity model shown in Figure 6, taken from [1], describes a high-level view of the processes necessary to maintain surveillance of the operational area. In the case of the Davis and Kabban experiment, airborne ISR assets tracked varying numbers of HVTs in an urban environment. The process, using terminology from the diagram in Figure 6, is as follows: when a UAV "sensed" a potential vehicle of interest, a human track manager generates a "track report" (indicated by the arrow from the "track" box to the "perform command and control" box). A "command and control" staff decides whether the track report has merit and assigns "taskings" to UAV operators to "sense" these HVTs. Based on their observations, the operators generate "sensed data reports," which are passed on to "exploitation" centers. The finished product of the entire process is exploited data reports and assessments.

"Perform Command and Control," "Exploit," and "Track" processes are typically performed at a nearby headquarters or on-board an AWACS or JSTARS. Although the "Sense" process involves track managers that are physically located outside of the battle environment, it also involves the UAVs, located within the battle space, that track the data. A change to the tactical process by which these UAVs collect data ripples throughout the entire data collection/exploitation process, and ultimately speeds the process of effecting stakeholders: the military commanders and combatants.
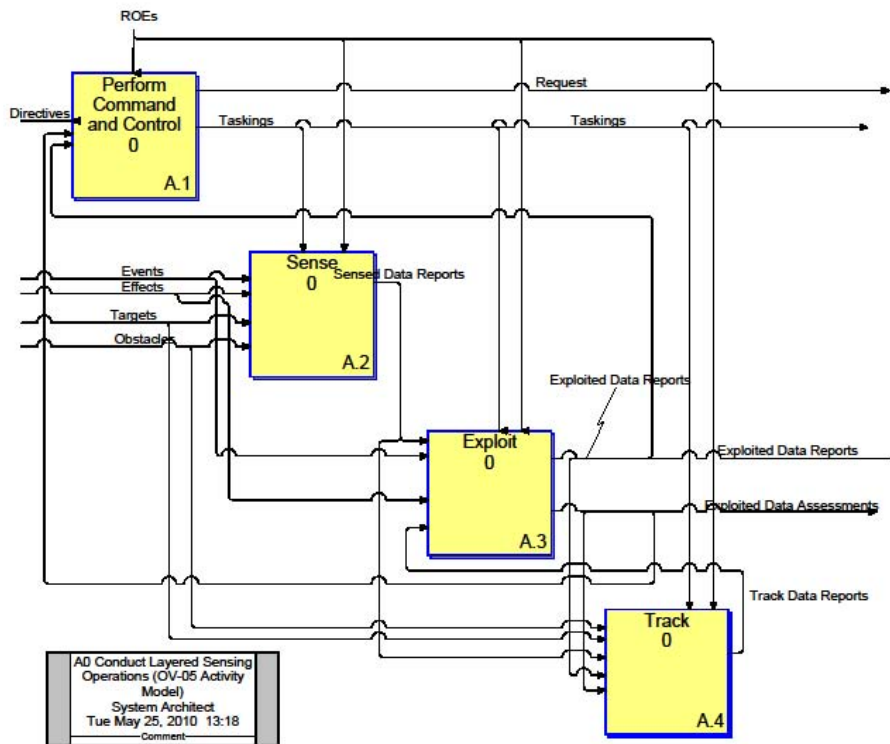
Figure 6: External Systems Diagram Conduct Layered Sensing Operations (OV-5 A0) [1]

Improving the quality and accuracy of data in the "Sense" process improves the ability of C2 authorities to generate accurate taskings. Data reports that rely on multiple data sources can theoretically track targets longer and supply more comprehensive data about the battle environment. An effective data collection process is crucial to mission effectiveness.

Previous research on the information gathering/exploitation process has shown that current standards and CONOPs cause bottlenecks at information analysis centers [1]. Human anaylsts were incapable of correctly tracking HVTs for long periods of time

under the conditions of the scenario. Interviews performed by this study confirmed that

high work loads, an inundation of taskings, caused high latency between the generation

of a surveillance request and the allocation of physical resources to that task. As latency

increases, the geographic area, in which the target-in-question could now be, expands,

increasing the difficulty of spotting the target and distinguishing it from other neutral

vehicles or people.

2.6 Wireless Sensor Networks

Wireless sensor networks (WSNs) are designed to perform a set of high-level

information processing tasks such as detection, tracking, or classification [18]. In the

case of this research, this may not be the case, as the nodes can be fairly sophisticated.

WSNs designs are an important consideration to ad-hoc UAV topology control because

they show the techniques used to improve coverage of the information-gathering

environment. That is, consideration to WSNs will give insight to the mission

effectiveness portion of the problem area.

2.6.1 Common Approaches to Topology Control for WSNs

One of the primary challenges with wireless sensor networks is the need to minimize

power consumption. Power consumption results from sending messages to other nodes.

As expected, protocols that achieve low power consumption are those that are capable of

sending fewer messages. WSN satisfy the needs of their stakeholders and consider

application-specific circumstances to mitigate their potential inefficiencies. For example,

effective topology control can reduce power consumption. While up-to-date topology

information can reduce the number of hops a message must travel to reach its destination,

strict topology control may require broadcasting, which is an expensive operation [18].

Choosing topology control mechanisms can be a double-edged sword.

WSNs consist of three types of devices, sensor nodes, relay nodes, and base stations [19]. Placement algorithms for relay nodes rely on two principles to assist in minimizing cost, Far-Near and Max-Min [19]. The general format for algorithms that place relay nodes consider sensor node location, probabilistic models for the detection of sensor information, and the transmission range of the nodes. Many of these algorithms simplify the map space into a grid to scale the problem space into a discretized domain, including the research in [19]. Common approaches that use this algorithm format include: Nearest-To-Base-Station-First algorithm (NTBF), Max-Residual-Capacity-First algorithm (MRCF), and Best-Effort-Relaying algorithm (BER), but none of these approaches are multi-objective-based.

2.6.2 Mobile Relay Nodes vs. Static Relay Nodes

Energy conservation is the most critical issue in WSNs and static nodes are preferred in most cases. Generally, WSNs refer to large networks of inexpensive, wireless, battery-powered devices that are relatively small [20]. However, within the context of this research, an abundance of static nodes is difficult because the system must be highly deployable. The motivation behind the usage of inexpensive static nodes is that they usually collect simple data, are disposable, and easy to distribute. In the case of intelligence operations, the data collected is usually more complex, such as video surveillance, which tends to increase the cost of the nodes. In addition, the region of interest is not usually known until the time of the operation is close, thus making it difficult to distribute large numbers of sensors quickly. The research in [20] touts the

savings in initial and operational costs for static nodes, but for military applications, the

option is simply not practical. Topology control mechanisms specific to ISR operations

must make the assumption that many of the nodes are mobile.

2.6.3 Routing in Wireless Sensor Networks

The concern in many WSNs is routing messages because, as in most WSNs where the

number of nodes is high and the networks tend to be densely populated, messages must

take many hops in order to arrive at their final destination. It is typically not a problem to

get the message to its destination, but rather, it is a problem to get it there cheaply. If a

message were to simply travel in the direction of its ultimate destination using an open

shortest path first (OSPF) algorithm, the resulting path cost would be much greater than a

routing algorithm that employed a method of estimating a connected dominating set

(CDS). This can be seen in Figure 7: Routing using OSPF versus routing using a CDS
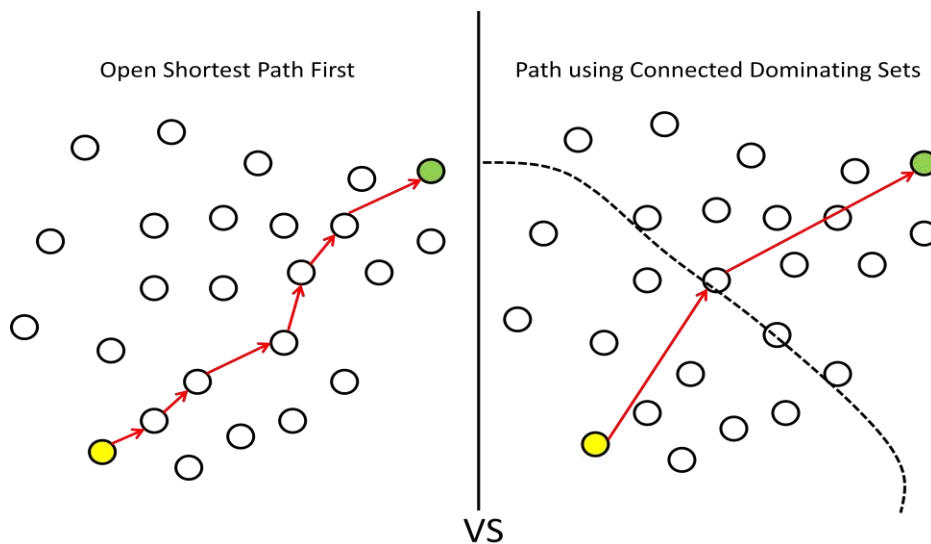
approximation

Figure 7: Routing using OSPF versus routing using a CDS approximation

Analysis from [21] shows that finding a CDS has no optimal solution because the problem space is too large, but by using heuristics that approximate space into polygon-shaped regions, near-optimal solutions can be found and routing in highly dense WSNs can be performed with relatively low setup and operating costs [21].

Alternative solutions to minimizing power include mobile base stations that change their location over time to account for the change in the trend of information origination [22]. With respect to ad-hoc C2ISR, the role of mobile base stations may be fulfilled low-level commanders traveling in tactical armored vehicles. Realistically though, the approach offers difficulty in unity of command and may not alleviate strain on satellite communication. These disparate base stations may need to resort to communication via SATCOM by virtue of their geographic separation. The constraint of this problem environment essentially limits base stations to static placement only.

2.7 Network Tasking Order

The USAF Network Operations Functional Concept from 2006 gives a vision for the framework of Air Force Network Operations (AFNetOps), describing how secret-level C2 will operate, and elaborates on how C2 will operate within the GIG. The framework includes a provision for the NTO, specifically defining it with the following [23]:

> The NTO directs the timely flow of information across the AF-provisioned portion of the GIG. Within the NTO, operational and scheduled events impacting the AF-provisioned portion of the GIG, taskings and additional information will be presented…NTOs are released daily and compliance is mandatory per Air Force policy.

Work by Compton on the NTO process describes a method for collecting information about the environment and assets to develop a plan for placing assets, airborne and ground-based, and governing what communication routes are authorized at what times

[24]. The work created an analog to the Air Tasking Order (ATO) by governing network

assets in a similar manner. Compton further discussed the implications of implementing

security and trust systems into that plan to protect information systems. The major

impact of this work was the concept of allocating assets in advance of operations

specifically for the purpose of supporting the demand for message sending, whether those

messages are simply radio voice messages or high-volume data collection from a wireless

sensor network.

Improving data collection in support of C2ISR operations requires that the

information assets be available and specifically tasked, days in advance, under the

guidance of intelligence reports and everyday activity in the operational area. With

respect to the use of small-platform, hand-launched UAVs, such as the Raven, troops

must be provided the opportunity to prepare for missions that incorporate extra

equipment. It is unrealistic to expect ground operators to draw spontaneous taskings to

HVTs and launch small UAVs. Improving data collection means creating realistic

predictions of significant activity and placing assets accordingly.

These hand-launched UAVs are more cumbersome to carry than not and, if under-

utilized, they would not justify their cost.   A plan must exist to place them only in the

hands of operators who need them. This research focuses on fine-tuning these assets

once they are in the operational area.

An NTO alleviates network traffic within C2ISR, but does not account for traffic that

arises from unforeseen events. An NTO would provide military commanders with a

basic idea of where units and military assets should be placed and to what capacity they

should be utilized to best serve the mission, and is a good tool for managing network

flow, but has a significant weakness in its inability to react to unscheduled spikes from IED detection or reporting of surprise enemy forces. C2ISR supported by a MANET has greater potential to provide the redundancy needed to react to a wide variety of contingencies. Tasking small UAVs to observe HVTs does not replace the need for robust planning via the NTO, but rather, it extends upon the idea by providing strategic guidance after the NTO plan has already been deployed to execute its primary task of surveillance, while also giving provision to its secondary role as a network topology resource.

2.8 Managing Traffic Using a Virtual Circuit Routing Model

An NTO process is clearly a "macro" tool. Using an NTO will help military planners predict areas of interest within a coarse timeframe: blocks of an hour or a few hours. Planning at a coarse temporal level gives an inherent property of permanence to the network topology, in that mobile nodes, such as UAVs, will tend to loiter in the same place for a long period of time. MANET routing protocols provide the service of message delivery "right now." In contrast, NTO-driven planning provides the service of making required resources available for a given period of time.

A virtual circuit (VC) is a connection oriented communication service that is delivered by means of packet mode communication. They are commonly used in problems where messages can be guaranteed the resources to reach their destination, and the difficult relies in using the available bandwidth optimally. A common metaphor to describe a VC is a telephone switchboard. If Bob needs to call Mary, the telephone switcher will connect a wire from Bob's terminal to Mary's terminal, and nothing will disturb that connection until the wire is released. In VCs, links are permanent.

The sensor networks in ISR operations are sparsely populated and routing becomes difficult because of the limitation of the bandwidth on the links between the nodes. Determining routes for messages can be modeled with virtual circuit routing (VCR) problems. Given a set of vertices, edges, edge capacities, and loads, the goal is to minimize the maximum load on any edge at any point in time [25]. As it applies to sensor networks, the goal is to spread network requests over network links as evenly as possible. Although the problem is NP-hard, a solution by Aspenes gives an asymptotically optimal $O(\log n)$-competitive algorithm where n is the number of nodes in the system [25]. Even when the durations, T, of the loads are known, the greedy algorithm yields a solution that is $O(\log nT)$-optimal [25].

2.9 Alternate UAV Movement Optimization Methods

The idea of governing UAVs with automated processes is not a new idea. There has been much work in this field, but the problem formulations have taken on many different forms. The variation between these methods ranges from distributed or centralized control, human controllers versus completely autonomous agents, and networks composed entirely of UAVs versus heterogeneous networks.

Work by Frew and Brown [26] tackles the same problem of setting up ad-hoc networks using small UAVs using COTS tools and the 802.11 wireless standards. The architecture of this system can be found in Figure 8. Much of the research, however, focuses on the physical characteristics of the signal loss and determining the effective range of these platforms. Incidentally, the results showed that the effective range is less than the researchers deemed practical. However, the researchers looked at a case in which they expected 5 radio-controlled airplanes to bridge a communication gap of 7km.
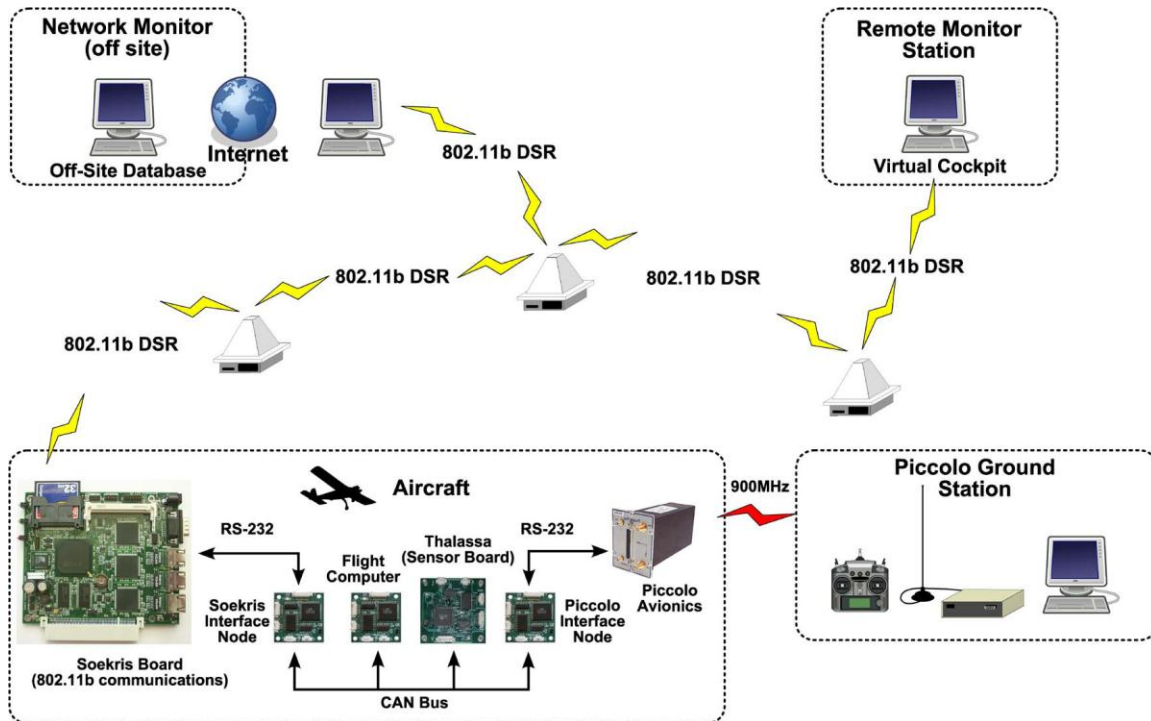
Figure 8: Networked UAS C3 Architecture [26]

The low density of communication led to a high drop rate and low connectivity.

Ultimately, the researchers concluded that a low platform-to-geographic-area ratio lends

itself naturally to a data ferry model, rather than a relay model. The researchers did not

consider a case in which the UAVs would monitor a specific urban area. Given the

assumption of a "bridging the gap" mission, they concluded that the solution would lie in

the ability of the UAVs to cooperate and form a chain or the ability of the UAVs to

coordinate data ferrying.

Building on this work and other work with data-ferrying, Larweck considered

organizing UAVs to perform a data ferrying role, while co-optimizing to view targets, in

a similar fashion to this research [27]. However, the targets remained stationary and the

number of UAVs considered was fairly small, about three or four. The configuration of

the simulation can be seen in Figure 9.

32
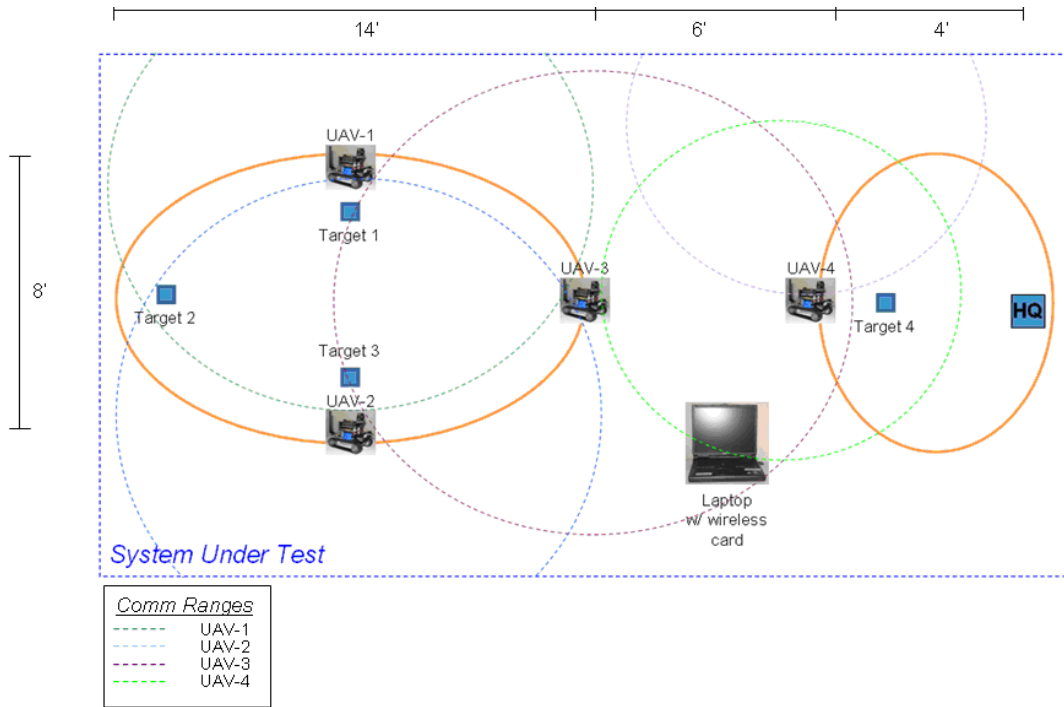
Figure 9: Larweck Simulation Configuration

The small problem space, the use of data ferrying instead of relaying, and the use of

stationary targets limited the problem space to a fairly trivial matter. This research aims

to look at sets up to 15 UAVs, up to 15 moving targets, and using real world network

traffic.

Kwak's research approached the problem from a different perspective, viewing each

of the UAVs as autonomous agents that would act in relation to the other UAVs around

it, using flocking behavior to govern finely tuned movements [28]. The research is able

to leverage different kinds of individual behaviors that work towards group goals, such as

creating a relay network or providing consistent ISR coverage of a particular area. An

example of the behaviors can be seen in Figure 10.
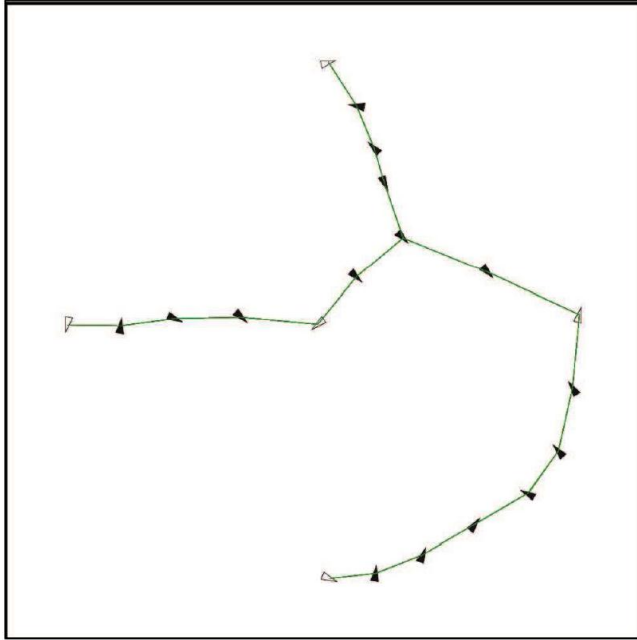
Figure 10 Kwak's Demonstration of a Mobile Agent Relay Network

While the research handles some of the same issues, the problem is a single objective

problem. The problem area does not actually consider any traffic patterns in particular,

but simply provides the means to send messages. While Kwak's work is impressive, the

demands made on the hardware are very strict. The UAVs act autonomously, and

without a controller. They fly from a base station and are left unattended. While the

two-dimensional simulation yields impressive ability to form ad-hoc structures, the

research leaves finely tuned flight controls out of the scope of the research. Much more

research is needed to realize this implementation. The work in this research, focused on

the planning aspect of UAV placement deals with technologies that are available today

and can be implemented with little further investigation.

III. Methodology

3.1 Overview

This chapter outlines the methodology used to determine UAS routing to co-optimize mission effectiveness and network performance. This chapter outlines the goals and hypotheses of this research, elaborates on the problem, generates a mathematical model, applies knowledge of the environment and describes the measures of merit on which the results of the algorithm will be judged. An outline of the experiments to be performed is given. The expected results are given and the expected performance factors are stated.

3.2 Approach

Based on the Layered Sensing CONOPS from [1], an approach to mission planning and execution for implementing a layered sensing architecture in urban environments using small-platform UAVs will be developed. Within that approach, a UAV routing algorithm that co-optimizes between mission performance and network performance is developed. To support that algorithm's mathematical model, a simulation test-bed will be created. That simulation test-bed is given a set of friendly, fixed ground units, network-enabled UAVs, HVTs, and network traffic, and will manage the positions of all entities and present an interface that enables the user to define a UAV tasking scheme. Then a novel tasking scheme, using a dynamic programming algorithm with a co-optimized utility heuristic, schedules UAV movement. At the conclusion, the networking planning problem is mapped onto a virtual circuit routing problem, for which a heuristic determines mission and network performance. To develop input data to the simulation,

notional scenarios from the Davis and Kabban research [1] will be adapted to the mathematical model of this research, appropriate for the inputs of the simulation test-bed.

3.3 Radio Communication Model

Based on the services that a C2ISR system must support, as outlined in the discussion of Joint C2 software tools, a wide range of traffic flows must be supported by the architecture. Choosing a physical-layer protocol can be difficult, as Figure 11indicates. The protocol is a function of the geographic span and the amount of bandwidth required to support the network. To keep the cost of UAVs low and ensure that the parts are widely available, it may be best to assume that they will use the 802.11 protocol.
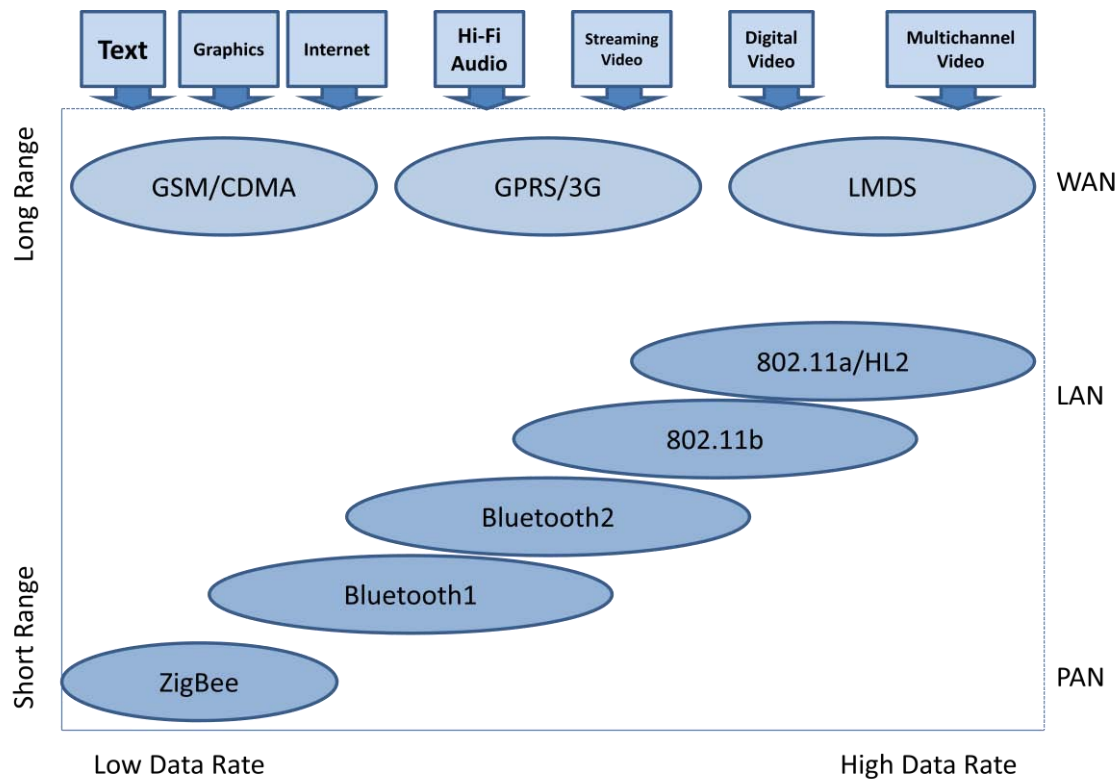
Figure 11: A bird's eye view of wireless technologies, according to data rate and range [18]

The radio communication model in this research will assume that two nodes within a distance of a given threshold will be able to communicate. UAVs are assumed to be able to communicate over a distance of 4.5 kilometers and humvees are able to communicate over a distance of 3 kilometers. Using a basic model of signal attenuation, we know that the amount of signal loss is proportional to the distance between the transmitter and the receiver by an exponential factor, n:

$$(1)\ \overline{PL_d} \propto \left(\frac{d}{d_0}\right)^n$$

In proportionality (1) from [29], $\overline{PL_d}$ is the path loss, n is a path loss exponent that depends on the environment, d_0 is a reference distance, and d is the distance between the transmitter and the receiver. To get a more concrete figure on the amount of loss, the previous proportionality can be stated as the following equation from [29]:

$$(2)\ \overline{PL_d} = \overline{PL_{d_0}} + 10n \log\left(\frac{d}{d_0}\right)$$

$\overline{PL_{d_0}}$ represents the signal loss that we experience at the reference distance, d_0. We generally choose 1m for d_0, which we can estimate will yield about 40dBm for $\overline{PL_{d_0}}$. The general estimate for the attenuation exponent, n, in free space is 2, which is a fair assumption, given that the Raven UAVs will fly above the ground units with little obstruction between them. Signal attenuation may be even more modest yet, given that the area of interest will be Iraq and Afghanistan, where there is less interference from cell phones and other radio devices. The transmitting power, P_{xmit}, of standard equipment in high mobility multipurpose wheeled vehicles (HMMWVs or "humvees"), such as the

Harris AN/PRC-117G radio, can be set as high as 20W [30]. We can assume the minimum receiving power, P_{rcv}, threshold of the UAV to be on par with a cell phone or laptop, at -70dBm, or about 100pW. If we set $\overline{PL_d}$ to our minimum threshold value of $10\log\left(\dfrac{P_{rcv}}{P_{xmit}}\right)$, and use this value to solve for (2), we get ~4.5km. HMMWV-to-HMMWV communication will be a little less because of the obscuration of buildings that are close to the ground. We estimate that they will be able to communicate at an approximate distance of 3km.

The actual communication range depends on many more factors, and distance may introduce noise and may affect data sending rates. These quick calculations are meant to give an estimation of the communication strength of the entities in the proposed architecture. The scope of this research will not delve deeply into the signal propagation aspect of the problem, but rather on the message-level analysis of network traffic. It will be assumed that the nodes will be able to communicate over the statically defined estimates at all times.

3.4 Simulation Test-Bed

A simulation test bed will be created for this research. This problem requires support for mobile networks, map discretization, simple ISR modeling, and network analysis. Many commercial-off-the-shelf software suites offer similar capabilities, such as OpNet, NS2, and SLAMEM (Toyon Corporation), but these tools complex and do not offer the option of discretizing the space, nor do they allow the surveillance and network routing problem to be joined satisfactorily. The simplified simulation test-bed can be developed more easily and allows the process to be more easily streamlined, as opposed to passing

data in between different software components, switching back and forth between data formats. The simulation test-bed will move UAVs using a given algorithm. handle efficient routing of traffic with that given routing scheme and determine results of UAV ISR and network performances.

3.5 Simulation Inputs

The simulation inputs are the geographic map, static ground nodes that generate routine traffic predictably, a set of HVTs, and a set of UAVs capable of detecting these HVTs. The symbolic representations of the inputs to the algorithm are listed below:

- T, time steps in the simulation, where a given time step is identified as a value, $t, t \in T, t \in \mathbb{N}, t < T$

- M, the map that defines possible locations of units. It has a width (east-west span) of x and a height (north-south span) of y. That is, M is a two-dimensional matrix, with dimensions x and y, where any two items within that matrix are considered to be adjacent if neither the x or y indices of the first item differ by more than 1 from the x or y indices of the second. That is:

  $\forall a, b \in M; x, y \in \mathbb{N}; x_1, x_2 < x; y_1, y_2 < y; a = (x_1, y_1); b = (x_2, y_2);$
  $adjacent(a, b) \Leftrightarrow |x_1 - x_2| \leq 1 \wedge |y_1 - y_2| \leq 1$

- F, the fixed nodes, where a single fixed node is denoted by $f, f \in F$

- L_F, the locations of all fixed nodes, where an the location of an individual node, f, is given as $L^f, f \in F, L^f \in M$

- H, the number of high value targets (HVTs)

39

- $L_{T,H}$, the locations of all HVTs at all time steps, where the location of the HVT, h, at time, t, is given as $L_t^h, h \in H, t < T, L_t^h \in M$

- U , the number of small-platform UAVs that can be manipulated

- l, the length of a time step relative to a message interval[1].

- R_F , the radio communication ranges of all fixed nodes, where the radio communication range of a given node, f, is represented as $R_f, f \in F$

- R_U , the radio communication ranges of all UAVs, where the radio communication range of a given UAV, u, is represented as $R^u, u \in U$

- J_U , the volume of traffic generated per time step by all UAVs observing an HVT. An individual traffic generation rate of a given UAV is denoted by $J_u, u \in U$. This is needed to calculate the amount of dynamic traffic being generated from UAVs.

- B_U , the maximum bandwidth limit of all links terminating in a UAV. The maximum bandwidth of a link terminating at UAV, u, is $B^u, u \in U$

- B_F , the maximum bandwidth limit of all links terminating in a fixed node. The maximum bandwidth of a link terminating at node, f, is $B^f, f \in F$

---

[1] A "message interval" is the finest grained amount of time in which a data transmission can be measured. For instance, for a simulation in which each time step represents one minute, a value of 600 for l means that each message interval is 100*ms* long. A point-to-point transmission time of 500*ms* takes 5 message intervals to arrive within the simulation's time-keeping system.

- M, the set of all messages, or flows, sent by ground units. Each message, m, contains a priority, p_m, volume, $size_m$, source and destination, src_m and $dest_m$, and a time, t_m.

The input data used to populate these simulation-specific inputs were adapted from previous research in different areas. The map and target data was adapted from a layered sensing (LS) architecture study from 2010 [1]. The traffic data was adapted from a stress test analysis of a tactical radio from the 82nd Airborne Division at Ft. Bragg, NC [17]. The map is based on an urban area in the Caspian Sea scenario using target movements based on a notional scenario made for the purpose of evaluating sensor collection and information exploitation systems. More information about the original data and its conversion into this research can be found in Appendix B: .

Ultimately, these data will be used to generate a solution: the placement of the UAVs, U, over time, L_U, where each UAV-position at a given time is represented as $L_{t,u}$.

3.6 Overview of the General Approach

There are two phases to the solution that need to be addressed in order to implement a full plan of action for UAV placement. Chapter 2 described both algorithms that place UAVs in real-time and the network tasking order. This solution addresses both planning phases as a unified methodology for governing the placement of UAVs. This solution includes (1) a planning phase consistent with the granularity of an NTO planning cycle, and (2) a fast-response planning phase that governs tactical movements.

For the first of those two phases, the effects of an NTO are modeled. The initial placement of the UAVs is considered. Unlike large UAVs, small UAVs are limited by

41

their radio communication range. Where large UAVs like Predators and global Hawks rely on satellite communication to receive commands from their operators, small UAVs, such as the Raven, communicate with their operators via radio. As it was determined earlier, the approximate threshold for radio communication within the environment of interest is 4.5 km. Urban areas vary greatly in size, but are potentially much larger in scale than this range. Consequently, the initial placement of UAVs becomes extremely important. Improper initial placement eliminates a UAV's ability to even reach tactically important locations. The collection and exploitation of critical information in the planning phase is integral to mission success.

The second of the two phases concerns tactical movements. In this context, tactical movements are not as intricately defined as pitch, yaw, and roll for the UAVs. Rather, tactical tasking dictates loiter-positions in coarse time quanta. In the case of this research, the time quanta will be set at 5 minute periods. The reasoning for this granularity is a compromise between the needs of the network and the needs of the ISR mission. Shorter loiter times favor the ability of the UAV to follow moving targets, but conflict with the stability of the network. Longer loiter times promote reliable network links that take hops through those UAVs, but compromise a UAV's ability to follow targets. For map sector spaces sized at about 1km by 1 km, 5 minutes allows a UAV to stay static enough to relay traffic, while coinciding with the normal speed of vehicles within an urban environment.

3.7 The NTO Model Implementation within the Simulation

The NTO is difficult to model in the simulation because the format of the incoming data is not necessarily formatted in a standard manner. In reality, the sources of

information may be human intelligence sources or the movement of opposing troops. As it relates to this research, the information sources are not computer-readable. Most of the information is interpreted by human planners. In reality, "fuzzy" information generally results in the placement of forces at locations that are near the target of interest, but not necessarily at the exact location. This research makes a point to model the stochastic placement of UAVs because it has serious consequences with respect to the outcome of the mission.

The module that is put in place to show this effect, however, is fairly simple. The HVTs, which move on a pre-determined schedule, each start at a different location. The immediate location of the HVT and the adjacent sectors define a candidate area from which the UAV base stations will draw. The UAVs will randomly select from this candidate area, place a fixed node that represents this UAV's base station at that area, and originate from it to begin. The likelihood that the UAV will start at the same location at which the HVT starts is small, but the UAV will certainly begin close to the target and will rely on its HVT-prediction methods to follow the target thereafter.

Although the model does not necessarily take into account any specific characteristics of intelligence data, it is meant to model the inefficiencies that will result from unreliable information, and show that the tactical model is able to overcome this difficulty.

3.8 Algorithm Overview for the Co-Optimization of Mission and Network Performance

To implement this second phase, an algorithmic approach, using dynamic programming is applied. Although the actual mechanisms needed for this routing algorithm are not implemented in this simulation, the framework is defined. In reality, a

dynamically chosen authority may be needed to dictate the movements of all UAVs, to remove conflict from different command sources. The algorithm may be performed onboard a single UAV and the results disseminated to other UAVs through the ad-hoc network. These implementation details are not specific to the algorithm herein defined, but are a necessary explanation to the reality of the operation of the algorithm

3.8.1 General Approach with Dynamic Programming

Dynamic programming mappings rely on two main mechanisms: breaking the problem down into parts, and the process of re-assembling those parts to form the solution. This problem mapping will also explain how to bound the original problem into a format that is consistent with this approach.

The algorithm that will be used to solve for the UAV schedule that co-optimizes between mission performance and network performance will be a dynamic programming algorithm, which works on the general philosophy that the problem can be solved by combining the solutions of the problem's sub-problems. The design is altered with some adjustments to improve the quality of the output and the speed with which that output is computed. Notably, this specialized algorithm will define a utility function for mission/network performance, a route-selection heuristic, and an alternative-selection threshold (or a "beam" width). This section will outline those algorithmic details and the algorithm design at varying levels.

The starting point for developing the algorithm is the general outline of a dynamic programming algorithm, which is given below [31]:

```
Initialize M[0] = 0 and M[1] = r₁
For j = 2,3,…,n:
      Compute M[j] using the recurrence
Endfor
Return M[n]
```

Figure 12: General Dynamic Programming Algorithm

The key components of this prototype are the function used to `compute M[j]`, the recursive structure, and the value array `M`[2]. The function, `compute M[j]`, is some method of evaluating the value of a given solution. In the case of this design, that method will be a utility function that is a weighted summation of three areas: network support, mission support, and network stability:

$$v_{placement,t} = k_m v_{mission} + k_n v_{network} + k_s v_{stability} \quad (3)$$

This evaluation represents the value of a given placement of UAVs at a given time. To determine the value of a UAV schedule over all time steps, each individual schedule value is summed over time:

$$value_{schedule} = \sum_{i \in t} value_{placement,i} \quad (4)$$

Since a UAV placement at a given time step is dependent upon the UAV placement that was made at the previous time step, the solution space of UAV schedules grows at the rate of $O(M^{UT})$. For large problem sets, it is simply not feasible to consider every possible solution, as a general dynamic programming approach may do, without any domain-specific improvements.

3.8.2 Problem Refinement

---

[2] This value array, `M[]`, should not be confused with the map array, *M[][]*. The naming scheme, `M[]`, is simply standard convention. The actual, defined value function will be renamed later.

As the problem maps to dynamic programming, it actually turns out that this is a dynamic programming problem nested within a dynamic programming problem. It turns out that a UAV-schedule must be solved for each time-step, and within each time-step, a location must be solved for each UAV.

When considering the dependency between two schedules, each within its own time-step, the UAV schedule for the current time-step limits the choices available for UAV-schedules in the next time step. It is important to consider the interaction between UAV schedules at different time-steps.

Then, within a given UAV-schedule, the placement of one UAV does not have an independent utility value—it depends on the configuration of all other UAVs. Network traffic will not route from one side of the map to the other by traveling through only one intermediate UAV, in all likelihood. The interdependency between individual UAV placements within a single schedule, coupled with the interdependency between individual UAV schedules between time-steps lends to a large problem space.

From a data input standpoint, the problem space could realistically be bounded by separating the map into fewer, larger partitions, or the time intervals could be separated into more coarse units, but these simplifications degrade the quality of the result. Instead, adjusting the algorithm will allow it to give more accurate, useful results.

The first elaboration of the algorithm is a simplification. We will bound the algorithm's search space between time-steps by looking no further than the current time-step. We will treat each UAV schedule as an individual problem of the conditions that occur at the current moment: traffic, location of the targets, value of the targets and traffic, etc. This simplification translates to iterating through each time-step, as opposed

46

to considering all overlapping, consecutive groupings of time intervals. This simplification means that UAV placements will not "predict the future" as well as a full search. A single UAV that moves to the best spot for right now may prohibit the option of moving to an even better spot two time-steps from now. It is the expectation that this "greedy" approach is acceptable because reasonably large swarms are capable of viewing the majority of the geographic space, even without erratic movement. UAVs should, realistically, not travel over a wide range, anyway, because it negatively affects the stability of the network and wastes fuel. A greedy approach is simple, effective, and appropriate to the problem domain. The problem is reasonably scaled using this simplification.

The second major elaboration of the algorithm is the definition of the problem decomposition, one of the algorithm's major components. When considering the placement of the UAVs, problem decomposition essentially translates to the number of UAVs to be placed at one time. At one end of the problem-size spectrum, placing only one UAV at a time would be fairly easy—it removes the complexity, by removing interdependency between UAVs. At the other end of the spectrum, placing all UAVs in a given time-step at once is incredibly difficult—all interdependencies exist and, as the number of UAV grows, the number of permutations grows exponentially. . Figure 13 shows how quickly the number of possible solutions can grow with respect to the simplicity of the decision made at one time-step, illustrating how much time it would take to solve non-trivial problem sets. Each icon represents a unique solution, a unique schedule of UAV placement.
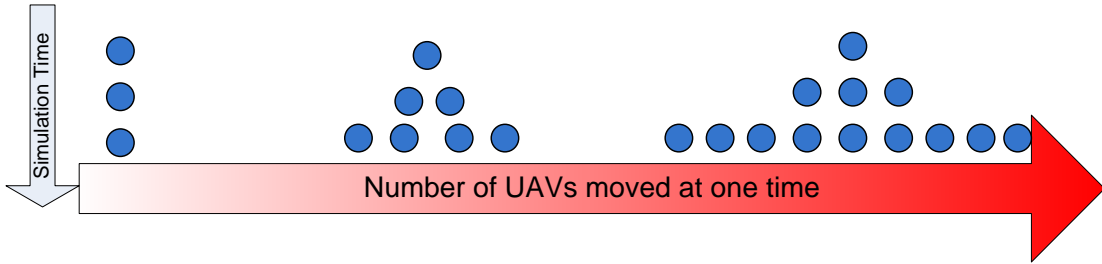
Figure 13 Expansion of solution space with respect to problem simplification

Instead, we parameterize this decomposition, and move p UAVs at a time. In general, the value of p is small, such as 2 or 3. With respect to the problem space, the effect on the solution quality should be greatly increased, while the number of possible solutions should be greatly reduced. A few permutations of 2 are much smaller than a single permutation of 10 or 20. As it affects the problem space however, placing two or three UAVs at a time captures the need for UAVs to depend on each other. Considering all permutations of 2 or 3 UAVs opens the opportunity to "bridge" large gaps with multiple UAV "hops," where single-UAV placements would not venture. With consideration to the real-world application, this bounding characteristic fits well with our mental heuristic of how to place UAVs. If a particular pathway requires three or four or five UAVs to bridge between fixed nodes, that pathway is probably detracting resources from other potential gains. Solutions from permutations of all 10 UAVs likely offer little value-added from multiple permutations of subsets of those UAVs.

The two primary elaborations of time simplification and sub-problem decomposition can be seen in the algorithm pseudo-code in Figure 14.

```
     for each t in T loop
          Solution(t) gets Place_UAVs(U);
     end loop

     Place_UAVs(U) return (VALUE, Placement Schedule) is
          double : V gets 0;
          Placement Schedule : R gets null;

          Remove p candidates from U
          for each permutation, q, of possible placements
          of UAVs in p loop
               W gets
               EvaluateNetwork(q,Nf)+EvaluateHVTObsv(q,H)+
               EvaluateStability(q, Nf)
               if W > V then
                    V gets W;
                    r gets q;
          end loop;
          Add r to R;
          return (V, Place_UAVs(U));
     end Place UAVs;
```

Figure 14 Pseudo-code elaboration of the algorithm formulation

The function, *Remove p candidates from* U, is meant to scale down the

problem space. The complexity is reduced by limiting the cardinality of the permutations

selected in a single time-step, in exchange for some optimality. Instead of looking at all

possible permutations of UAV position assignments, a subset is chosen at each iteration.

For example, suppose we were to have a problem space in which we had 10 UAVs, and

based the speed on those UAVs, each UAV has 9 distinct choices for their position on the

next time-step. If we were to select p = 10, we have 9^{10} possible schedules to choose

from, in this one single time step. If we were to choose p = 2, instead, we would select

between $5(9^2)$ different schedules. The latter choice is smaller by a factor of about 8.6

million.

The mechanism that governs the selection process within *Remove p candidates*

from U can vary between implementations without affecting the size of the problem

space. In this simple implementation, we will simply choose the first two arbitrarily

ordered UAVs in *U*. Some additional heuristics may yield better results, but may add

potentially unnecessary computational complexity. For instance, results may be

improved by selecting the two UAVs that are closest to each other. In this way, the

solution is more likely to contain "double-UAV-bridging" between static nodes, but may

take $O(U^2)$ time.

3.8.3 Raw Measures of Merit

As it was mentioned before in Eq. (3), the main measures of merit are the UAV

schedule's ability to support the network traffic, the mission, and the stability of the

network. The evaluation functions, `EvaluateNetwork (q,N`$_f$`)`, `EvaluateHVTObsv`

`(q,H)`, and `EvaluateStability(q, N`$_f$`)` are used to measure these. Network traffic

is measured by the number of messages that successfully reach their intended destination,

and is an important measure of merit because it is a direct metric for the success of the

network. Mission effectiveness is measured by the amount of time that UAVs loiter over

a target and the value of that target. Target observation is an important measure of merit

because it dramatically increases our ability to respond to significant events if UAVs are

capable of observing them as they develop and as they occur. Stability is important

because it eases the burden on topology control methods. As it was discussed earlier with

MANET routing protocols, broken routes are expensive to fix. Fixing a route means

broadcasting messages, which consumes a lot of power and drains resources away from a

UAVs ability to fly.

The `EvaluateNetwork` function is difficult to state formally because the software-

defined algorithm simply determines what the network topology will be, based on the

positions of the nodes, and uses a largest-message-first algorithm to determine, at the

time-step in question, which traffic messages will arrive and which will not. This process

is pictured in Figure 15 Evaluation of Network Performance
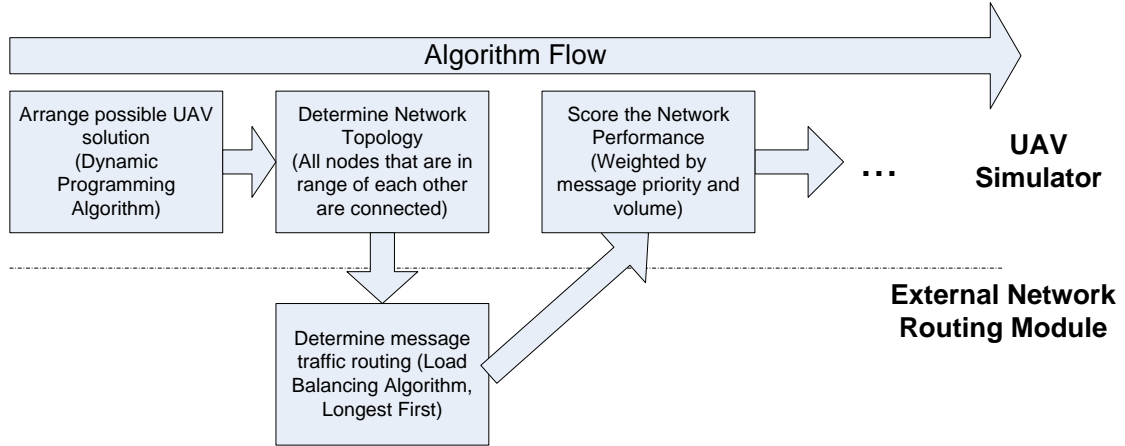
Figure 15 Evaluation of Network Performance

As the figure indicates, most of the actions performed to determine the network

performance occurs within the simulator designed for this research. The determination of

the actual message routing occurs in an external application for load balancing. When

the external module returns the results of the network load balancing, the simulator

assigns a score to the topology by the following equation:

$$v_{network}(raw) = \sum_{m:Messages}^{M} \frac{size_m}{2^{p_m-1}} I_{arrived} \ (5)$$

In this equation, *M* represents the set of all messages in the simulation. These messages

can be scripted from the data set or they can be dynamically created as a result of UAV

video generated from observing an HVT. The score is basically the size of the message

halved for each level of priority reduction. The priority of the message, p_m, ranges from

1 to infinity, where 1 is the highest priority a message can be given. $I_{arrived}$ is an indicator function that returns 1 if the message arrived at its destination or 0 if not. In other words, no points are awarded if the message is not delivered.

The `EvaluateHVTObsv` function is simpler than the network performance piece. Mission performance is measured by the duration of time a UAV loitered over a target, multiplied by the value of that target, and scaled back by a pre-determined schedule if the UAV was close, but not within the "viewing" threshold of the target. This relationship can be expressed using the following equation:

$$v_{mission}(raw) = \sum_{h=0}^{H}(t_{end} - t_{start})v_h S_{viewed}$$
(6)

The default distance-to-cost schedule assigned for this simulation is shown in Figure 16. In reality, this schedule is a set of tuning parameters that may be manipulated by the user, and that schedule can introduce bias into the results. The schedule may be altered to relate more closely to the reality of the quality of the images that are returned by the UAV, but should not be altered after consideration of the data set. Here, there is an unexpected complication. By allowing UAVs to attain residual utility from a relatively long distance, conflicts arise in assigning UAVs to HVTs. Were the rules to dictate that UAVs only attain utility/value by being located directly above a target or extremely close to a target, it would be a trivial problem to break ties between UAVs near a given HVT. Simply pick one of the eligible UAVs and elect the other UAV for another target. To enforce the rule of one-UAV-to-one-HVT and the rule of fractional-points-by-distance simultaneously, breaking ties and assigning UAVs to HVTs becomes yet another NP-hard optimization.

Outside of Visual Range, 0%

100% Visual Range, 40%

66.7% of Visual Range, 60%

Target

10% of Visual Range, 100%

UAV

Target

UAV

Reward = v_{target}•time•100%

Reward = v_{target}•time•40%
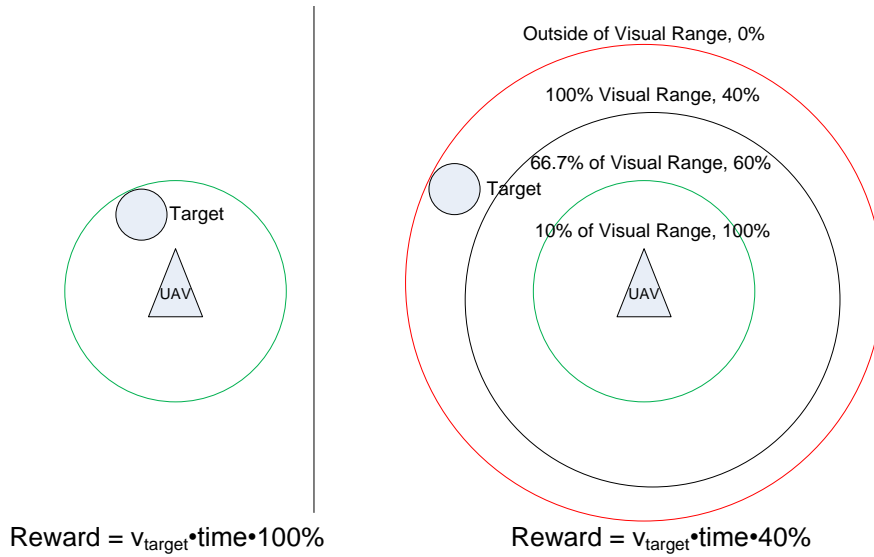
Figure 16 HVT Detection Reward Schedule

To avoid a potentially complex optimization, a simple heuristic is used:

```
Scorecard contains u, List of (HVT, score)
for each u in U loop
      for each h in H loop
            v = determinePercentagebyDistance(u,h)*h.Value
            if (v > 0)
                  Scorecard.u.add(h,v);
      end loop;
end loop;

Sort Scorecard by List size, lowest to highest
for each list in Scorecard
      Sort list by v, highest to lowest
end loop;

for each u.list in Scorecard loop
      Assign u to u.list.first, take list.v
      Remove all occurrences of u.list.first from remaining
      Scorecard
end loop;
```

Figure 17: Breaking Ties with UAV-to-HVT assignments

Essentially, the selection process is a greedy tie-breaker. In short, the process goes, as

follows. For each UAV, determine all HVTs that can be viewed and the value that would

be achieved by doing so. Cycle through all UAVs, starting with the most-constrained,

that is, the UAV with the fewest options. Select that UAV's most valuable HVT and assign them as a pair. Remove that HVT option from the list of all other UAVs. Continue until there are either no more UAVs to select or no more HVTs to select.

Finally, the network stability will be measured in the number of change to the network from time-step to time-step. During the course of the simulation, the simulator will keep track of the network topology and count a change as the loss of an edge that occurred between time-steps. As opposed to the former two metrics, change will be assessed as a penalty, rather than an award. This can be formalized as the following:

$$v_{stability}(raw) = -\sum_{i=0}^{E} loss\_from(E_{t2}[i], E_{t1}[i])\ (7)$$

Here, E is the number of edges in the network from the previous time-step. If the edge from E_1 no longer exists, a penalty is assessed. Note that the difference operator only goes one way. If an edge is added, there is a difference between the networks of the two different time-steps, but no penalty is assessed in this situation. Increasing the robustness of the network over time should be encouraged, or at least it should not be penalized. Only edge losses are penalized.

3.8.4 Refined Measures of Merit

One of the challenges with using a combined utility function as a measure of merit is the consideration of the range of values of the components. If the ranges of values of the individual components vary greatly, component utilities will be disproportionally represented. For example, if the network score were to consist solely of raw bytes delivered and the stability score were to consist solely of total links dropped, the magnitudes of these values would be 10^6 and 10^1, respectively. The resulting score

would discount the significance of the stability score almost entirely, and the aggregate utility value would represent the network score only. It is important to implement a mechanism that will scale the individual utility scores appropriately to their respective numerical ranges.

3.8.4.1 The Bias of the Normalization Factor

To scale the raw measures of merit into appropriately weighted values, a normalization technique is implemented. In a generalized sense, this means that each individual raw score will be divided by its maximum possible value. The resulting value is a proportion of the best possible solution. However, the difficulty with this problem is that the maximum possible value is not known. The problem space is too complex to calculate such a value. Instead, an approximation of the maximum possible value of a given utility score is generated to normalize the raw score. There will be an inherent bias on the utility values when using a normalization factor that is an approximated maximum.

Approximations of the maximum values that err to the side of over-estimation guarantee that no single normalized utility value will exceed 1. To obtain a normalized score greater than 1 allows a given utility value to over-perform by an immeasurable factor. The bias may diminish component values whose domain has a smaller numerical range.

Conversely, to approximate maximum utility values to a value lower than their theoretical maximum over-values the normalized component utility functions. Under-approximating inflates values and potentially over-represents utilities whose numerical range is greater.

In this research, we choose to over-approximate maximum values because the process of obtaining that value is easier, both conceptually and computationally. Without a method of finding an exact maximum value, it is impossible to remove this bias from the utility. It may be possible to approximate an upper and lower bound on the maximum value and average the two, but this method yields more computational complexity and processing time than it would be worth to further remove bias. Rather, we accept the bias and understand that utility functions that yield higher raw values will be slightly under-represented. The resulting bias could be further mitigated by using secondary multiplicative coefficients to adjust the normalized score to the accurate level, but more experiments would be needed to achieve this goal.

3.8.4.2 Network Utility Normalization

The raw network score, calculated by Eq. (5), is assigned a hypothetical maximum score approximation by assuming that all messages are delivered. Effectively, this means that the indicator function always returns true. In all likelihood, the estimation produces a maximum result that is greater than or equal to the theoretical maximum because, although it may be possible to send all scheduled traffic messages, the solution space is bounded by the total possible set of locations to which UAVs can travel, which may not include full-connectivity locations. The normalized utility function is the following:

$$v_{network} = \frac{\sum\limits_{m:Messages}^{M} \frac{size_m}{2^{P_m-1}} I_{arrived}}{\sum\limits_{m:Messages}^{M} \frac{size_m}{2^{P_m-1}}}$$

3.8.4.3 Mission Utility Normalization

56

The raw mission score, calculated by (6) is assigned a hypothetical maximum score by assuming that each HVT is observed for the full duration of the simulation (where $(t_{end} - t_{start}) = T$) and the observers are located directly above each target, resulting in an S_{viewed} value of 100%. The maximum defined here is at least an over-estimation because it may not be possible to view all targets—certainly not in cases where there are fewer UAVs than HVTs—and certainly not when the HVTs are outside the base-station range of the UAVs.

$$v_{mission} = \frac{\sum_{h=0}^{H}(t_{end} - t_{start})v_h S_{viewed}}{\sum_{h=0}^{H} T v_h}$$

3.8.4.4 Stability Utility Normalization

The raw stability score, calculated by Eq. (7), is simply a total number of links dropped. The worst possible case means that all links are dropped. The total number of links dropped cannot be greater than the total number of links total. This estimation is easily an over-estimate:

$$v_{stability} = \frac{-\sum_{i=0}^{E} loss\_from(E_{t2}[i], E_{t1}[i])}{|E_{t1}|}$$

3.8.5 Normalized Total Utility

With the normalizations, the total utility function becomes the following:

$$v_{placement} = \frac{\sum_{m:Messages}^{M} \frac{size_m}{2^{P_m-1}} I_{arrived}}{\sum_{m:Messages}^{M} \frac{size_m}{2^{P_m-1}}} + \frac{\sum_{h=0}^{H}(t_{end} - t_{start})v_h S_{viewed}}{\sum_{h=0}^{H} T v_h} - \frac{\sum_{i=0}^{E} loss\_from(E_{t2}[i], E_{t1}[i])}{|E_{t1}|}$$

The under-valuing biases in the total utility function still exist. Their effect is

theoretically more greatly pronounced in the first two terms and of a lesser effect in the third term.

3.9 Experiment Design

To support the hypotheses of this research, the experiments are designed to estimate the solution space, then within that space, show the improvement of UAV placement, using the utility/heuristic-driven algorithm specific to this research, compared to baseline methods that schedule greedily based upon either network performance or mission performance, but not both.

To define a lower bound to the solution space, a Monte Carlo, or random method, will be implemented. The results will give a general glimpse of the solution space, by allowing many iterations to be run, due to the low computation time. The solutions may give some hints as to what characteristics of a solution will make for a high performing UAV schedule.

The implementation of the full algorithm shows how the heuristic presented in this research can provide near real-time tasking of UAVs for a dual-role co-optimization. It is expected that the dynamic programming algorithm perform much better than the Monte Carlo solutions, even with fewer trials.

All of the experiments will be run with the same geographic urban map on a 10km^2 area, varying the number of UAVs and the saturation of HVTs. Table 1 shows scenarios that will be run with the dynamic programming algorithm. Using four different values for the numbers of UAVs and the numbers of HVTs, there will be 16 different experiments. Network traffic will not vary because the traffic will remain proportional to the number of nodes that are participating in the experiment.

Table 1: Dynamic Programming Experiment Set and Their Iterations

Number of UAVs/ Number of HVTs	1	5	10	15
1	DP 01-01, 30	DP 05-01, 30	DP 10-01, 30	DP 15-01, 30
5	DP 01-05, 30	DP 05-05, 30	DP 10-05, 30	DP 15-05, 30
10	DP 01-10, 30	DP 05-10, 30	DP 10-10, 30	DP 15-10, 30
15	DP 01-15, 30	DP 05-15, 30	DP 10-15, 30	DP 15-15, 30

The unit-amount of traffic that will be generated is based on previous research done on ad-hoc wireless networks in combat environments [17].

The data sets define 16 different configurations that correspond to 16 different sets of operational conditions.   For a given one of those configurations, the UAVs will be placed in a stochastically chosen initial position. A bad placement could potentially doom to the solution by placing the UAVs out of range of the targets. In contrast, a lucky placement could be exceptionally beneficial. To mitigate the consequences of biased placements, 30 iterations of each experimental configuration will be performed.

This experiment will be run over 30 iterations. The expectation is that the results will return a range of values over a normal distribution because of the randomness of the initial placement. To overcome the bias that may occur from this randomness, a sample size of 30 should help to reduce the spread of the resulting data and give reliable results.

3.10 Control Experiments

To establish baselines and give context to the results of these experiments, some control experiments will need to be run. In addition to running the dynamic programming algorithm on the data sets, we will also conduct experiments using a Depth-First-Search-with-Backtracking (DFS-BT) algorithm. The DFS-BT will be able to look at every possible solution. Using this algorithm, we will be able to determine what the

optimal solution is for a given data set. The downside to this algorithm is that it takes too

much time to run. We will only be able to find optimal solutions for "toy problems" that

contain only a small number of UAVs and HVTs. These experiment sets can be found in

Table 2.

Table 2: DFS-BT Optimal Experiment Set and Their Iterations

Number of UAVs/ Number of HVTs	1	5
1	DFS-BT 01-01, 5	DFS-BT 05-01, 5
5	DFS-BT 01-05, 5	DFS-BT 05-05, 5

The DFS-BT experiment set offers an "upper-bound" by defining a limit which no

solution can trump. It is not possible to include the optimal for all the data sets that we

would like, but we can include a rough "lower bound" by including a Monte Carlo

simulation in which UAV positions are chosen at random. By running a large number of

simulations using this method, we expect to generate an acceptable solution. We assume

that if we cannot generate solution that would result from random guessing, then we must

create a better method. These experiment sets can be found in Table 3.

Table 3: Monte Carlo Experiment Sets and Their Iterations

Number of UAVs/ Number of HVTs	1	5	10	15
1	MC 01-01, 500	MC 05-01, 500	MC 10-01, 500	MC 15-01, 500
5	MC 01-05, 500	MC 05-05, 500	MC 10-05, 500	MC 15-05, 500
10	MC 01-10, 500	MC 05-10, 500	MC 10-10, 500	MC 15-10, 500
15	MC 01-15, 500	MC 05-15, 500	MC 10-15, 500	MC 15-15, 500

Again, given a stochastic approach to the initial placement, multiple runs of each of

these control experiments will need to be performed in order to reduce the bias placed on

the results by those very placements. Monte Carlo simulations require little

computational effort and require little time to complete. Each of the test configurations

for the Monte Carlo simulations will be run 500 times. The high number of iterations should create much confidence in the results, and additionally, show the strength of solutions that can be generated with little effort in a comparable amount of time. The DFS-BT simulations will run only 5 times each because of the amount of time that this research will allow. Although the confidence in the solutions will be lower, as much bias as time permits will be removed from these solutions.

3.11 Performance Metrics

The primary measures of success are reliability and performance. If the communication system supporting C2ISR, augmented by the use of MANETs, exhibits higher fault tolerance and maintains connectivity, despite failures, the system is reliable. If the system is capable of delivering large messages quickly, the system is performing well. The ultimate measure of success is the composite v value over the course of the simulation. Previous research had considered the ISR aspect of the problem and settled on intelligence metrics such as the amount of time the target was tracked, how long it took to finally locate a target, or the length of time that a particular UAV was able to continuously view the vehicle [1]. While all of these metrics illustrate how effectively the system "sees" targets, nut not necessarily how well the operator sees the target. These metrics do not account for network performance at all. Tuning parameters for the weight of the importance of one aspect over the other will be kept static throughout the simulations to maintain the consistency between experiments.

3.12 Expected Performance

It is expected that a higher number of UAVs within the same map area should generate better network performance because the UAVs have better coverage of the area,

giving them a higher probability of observing any given HVT, and increasing their

connectivity within the ad-hoc network. The expectation is that the increase in UAVs

improves the performance, but likely with diminishing returns after a certain point. The

increase in performance that results from more UAVs may "plateau." An over-saturation

of UAVs provides no added benefit when compared to a network model that already has

near-perfect surveillance and network coverage.

An increase in the number of HVTs will likely increase mission performance because

there will simply be more targets to watch, as well as a greater likelihood that any given

UAV will be in range of a HVT. Given that UAVs are unable to move outside of a

certain distance from their base station, a UAV may potentially be unable to view any

HVTs at all. Opportunity is increased with more HVTs.

3.12.1 Relative Performance

The experiments should prove the conjecture that the solutions produced by the

dynamic programming algorithm outperform the solutions produced by Monte Carlo

simulations when applied to the same scenarios. The experiments should also prove that

the DFS-BT algorithm produces the best results and serves as the theoretical maximum

performance. The results are expected to be similar to those in Table 4.

The results will be averaged over the total number of iterations for each experimental

condition. The theoretical results shown in Table 4 represent the averaged results for a

set of scenarios that contain the same number of HVTs, however, it is expected that this

trend manifests in all test cases. One of the difficulties with achieving these results will

be the lack of data for large data sets on the DFS-BT algorithm. There will simply be too

little time to obtain results on large data sets. The analysis will rely on comparing small

data sets or by projecting the trend of DFS-BT results onto the larger sets.

Table 4 Expected Results for Total Utility Comparison between Algorithms
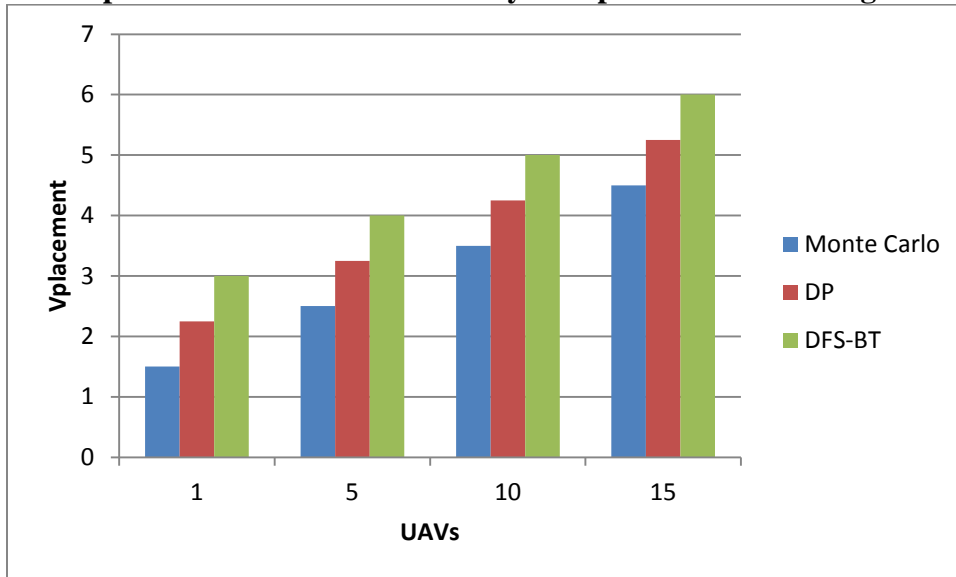
3.13 Summary

This chapter discussed previous work that inspired this research, which outlined the

CONOPS for implementing an LS architecture. The areas of ISR and C2 that could be

improved were presented, and an approach to accomplish those goals was outlined. It

was concluded that a deterministic, simulation-based approach to scheduling UAVs

within that framework to compromise between mission and network performance is a

non-trivial task and requires the implementation of a dynamic programming algorithm

that uses heuristics to make the computation time more manageable. In order to avoid the

complexity of existing ISR and network simulators, a dedicated simulation test-bed was

created for this research.

IV. Results and Analysis

4.1 Introduction

The overall goal of this research is to develop a concrete method for planning UAVs to fill a dual-role in ISR operations and network configuration operations. The two main components of realizing that planning are the simulation test-bed, which implements the mathematical model and dynamics of the system, and the algorithm that determines where to schedule UAV placement. This chapter outlines the implementation of that test bed and the particular implementation details of the algorithm to solve the UAV placement problem. Finally, the results of that algorithm are given and the implications of those results are elaborated.

4.2 Simulation Test Bed

The main component of this research is the simulation test-bed. It consists of two main operating modules, one much larger than the other. The larger of the two modules contains the world model, algorithm interface, and the main driver. The smaller of the two parts is external and routes network traffic. This overview can be seen in Figure 18.

The external module consists of a "dynamic rate queue controller" (DRQC), designed by the researchers in [32]. The DRQC passes off messages to a network tool called CS2, which determines routing and bandwidth usage.

The larger main module contains a model of the world, which defines all entities that can exist and their relationships. The algorithm interface contains the different methods of solving the problem and connects with the main driver to obtain information about the

world, and alter that information as the policies of the algorithm dictate. This high level

overview can be seen in Figure 18.
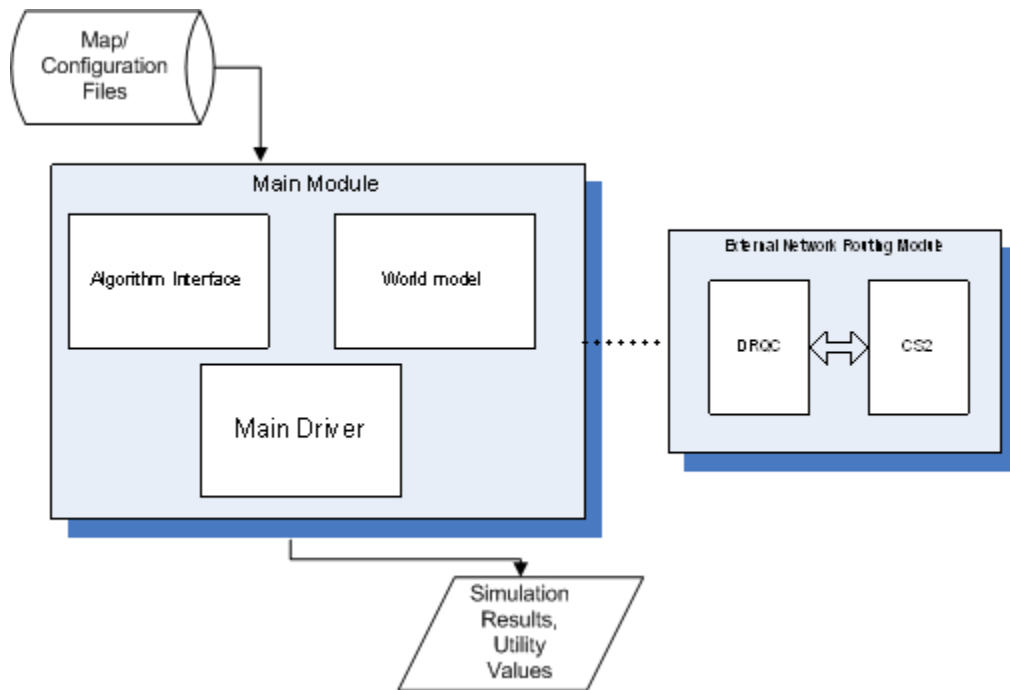
Figure 18 High-Level Overview of the Simulation Test-Bed

The main module reads in the map and configuration files, processes the data, runs the

algorithms to determine the performance of the system and outputs the results of the

simulation, based on the UAV schedule created by those algorithms. More intricate

details of this model can be found in Appendix E: .

4.2.1 Operational Overview

   All of the operations pertaining to the algorithm and the simulation execution occur in

the main driver. A further elaboration of this module can be found in the operational

overview in Figure 19. Configuration files determine the layout of the base stations and

other fixed nodes, the starting position and attributes of the UAVs and HVTs, and the

network traffic that will occur during the course of the simulation. Beyond the creation

of the software-defined model of the world, the simulation runs in two main steps: pre-order determination of the UAV scheduling, and the simulation of that schedule under the conditions defined in the configuration. After the algorithm-determined UAV schedule is run in simulation, a score is assigned and output to a simulation log. The two main steps require some further elaboration.
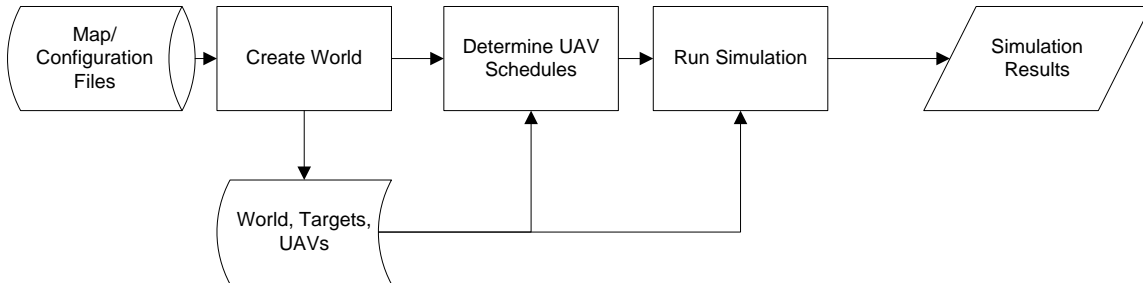
Figure 19 Operational Overview of the Main Driver

4.2.1.1 The Run Simulation Component

The "run simulation" step can be represented by the functional decomposition in Figure 20. The process moves fairly sequentially: move UAVs according to their pre-determined schedule, determine the network overlay, line up network traffic based on pre-scripted messages and dynamic messages resulting from UAV video data, then determine the scoring based on the utility functions, and iterate through each time-step until all time-steps have been performed.
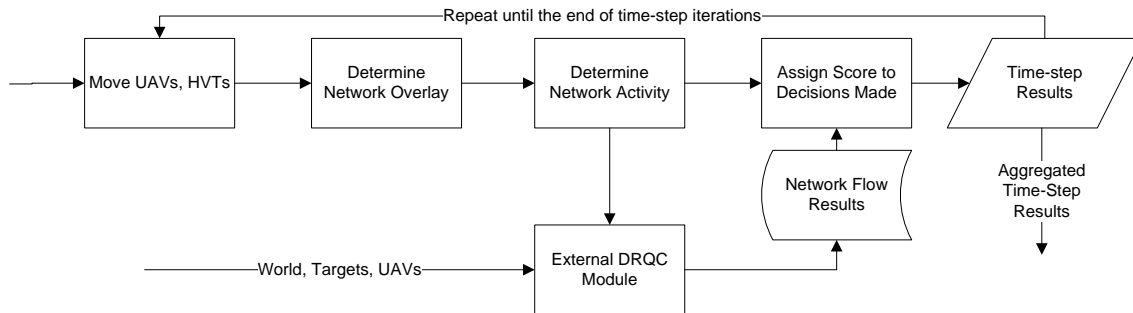
Figure 20 Functional Decomposition of the "Run Simulation" Module

To describe the "Run Simulation" module in more detail, we will briefly describe each of the module's components and explain the mechanisms used for their implementation.

"Move UAVs and HVTs" uses the schedules assigned to the units to determine their next location. For the HVTs, these schedules were scripted. For the UAVs, these schedules were assigned using the prescribed algorithm in the previous step.

To "determine network overlay" every distance between every node pair is calculated. For any two nodes, if the nodes are within the given threshold distance, a link is created between them. This threshold distance is the lower of the two communication ranges. This relates back to the radio communication model described in Chapter 3, and a visual representation of this relationship can be seen in Figure 21.
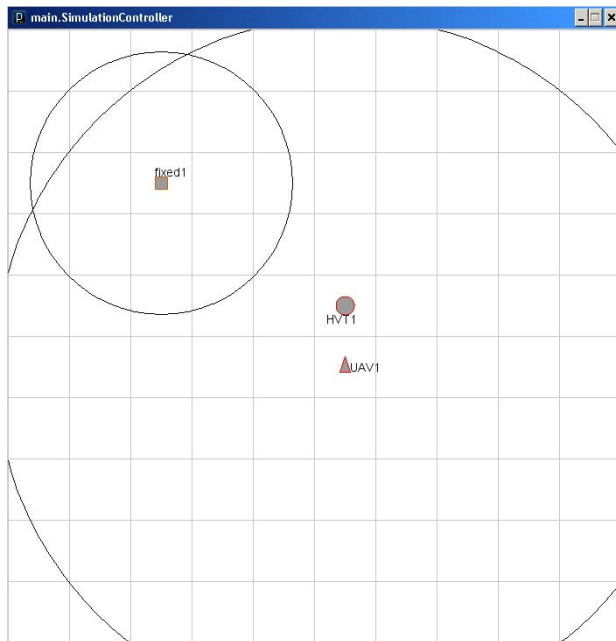
Figure 21 Visualization of Communication Range in the Simulation Test Bed

Figure 21 shows UAV1 and FixedNode1 and their respective communication ranges. A link between nodes implies that each of the nodes in question can transmit to the other, even beyond the simple need for one node to be able to transmit to the other.

Communication range is a simplification from the need to consider both transmitting range and receiving range. The implication is not that these two distances are the same, but rather that, when a message is sent from the sender, the receiver must have the ability to acknowledge receipt of a message for effective communication. A reliable link cannot be made between two nodes unless they are both "in range" of each other. To satisfy the depiction in Figure 21, these two nodes would *not* link. FixedNode1 is within range of UAV1, but UAV1 is too far outside the range of FixedNode1.

The "determine network activity" component draws from a large pool of scripted traffic that is defined by the input files. For the experiments conducted in this research, traffic was based on stress-testing of tactical radios for Army operations [17]. More information on this data can be found in Appendix C: . The other part of the data comes from dynamically created traffic generated from UAV video-feeds generated from HVT observations. These streams are not always present because UAVs may not always be currently tracking a target. More information about the generation of this video traffic can also be found in Appendix C: .

The export of the network routing from the "DRQC Module" was explained in Chapter 3. Within the context of this research, the model used within is essentially a black box. The general idea is that the module performs near optimal routing (and in this case, that optimal routing is within a factor of $O(\log n)$ of the optimal solution). The assumption is that the real-world implementation of this system would employ an appropriate system that would route traffic efficiently. In the case of the DRQC module, the network flows are routing using a load balancing algorithm that employs a largest-flow-first heuristic.

This module returns the results of the flow routing, from which, the utility function can determine the scoring that the corresponding UAV placement will earn. The process is repeated for all time-steps, collecting and aggregating the utility score at each time step. At the end, the aggregated utility score is reported.

4.2.1.2 The Determine UAV Schedule Component

Still in reference to the Operational Overview in Figure 19, the first step of the two main operational parts in the main driver determines the UAV scheduling based on the prescribed algorithm. For the control experiments, this prescribed algorithm will be the Monte Carlo simulation and the depth-first search with backtracking. For the main experiment, the prescribed algorithm will be the dynamic programming algorithm that is the object of this research. The "prescribed algorithm" roughly encapsulates this entire module. An algorithm should incorporate all of these components, or least all of the decision points. For example, the Monte Carlo simulation performs all of the decision actions described in these modules, but does not perform any kind of network simulation and does not repeat within a given time-step until all candidates are considered. It simply selects a placement for a UAV, untested and using no prediction methods, and sets the decision permanently. Both the DFS-BT and the DP algorithms perform every step defined in Figure 22.

Many of these steps are similar to those described in the previous module, "Run Simulation," so they will be mentioned only briefly, but the unique steps will be elaborated further. In essence, this module will consider prospective decisions and determine their worth based on a predicted outcome using the same external module that the "Run Simulation" module would use.
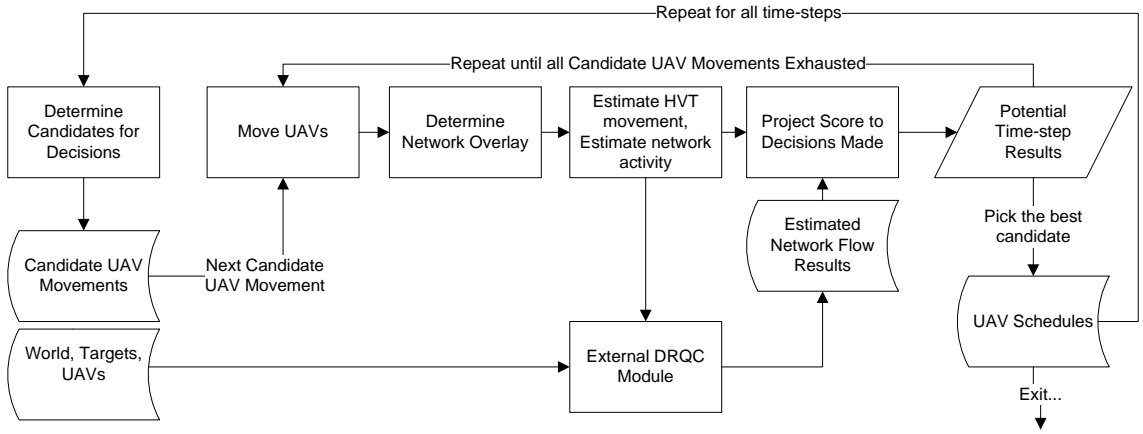
Figure 22 The "Determine UAV Schedules" Component

The "Determine Candidates for Decisions" module uses the speed of the UAV to determine all of the spaces that the UAV can reach within this time-step. Given that this is a grid world, the candidates will be adjacent sectors, calculated outward from current position of that UAV. A visualization of this candidacy is given in Figure 23.
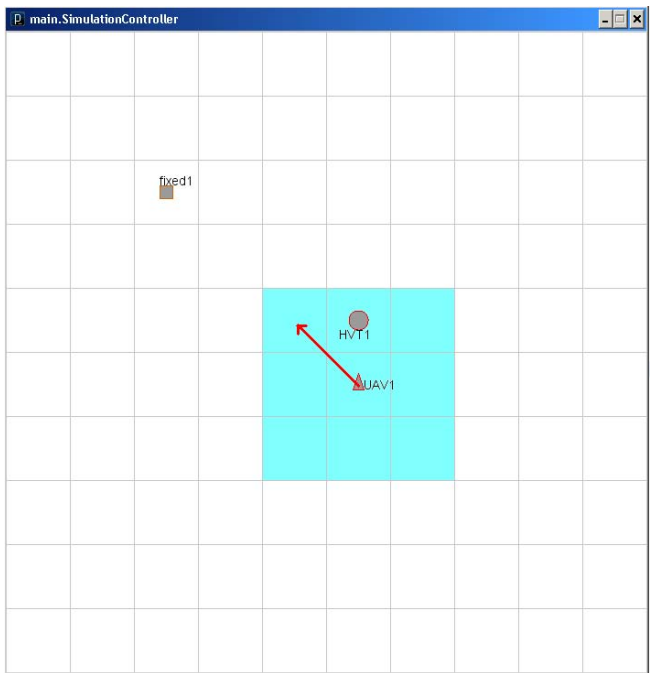
Figure 23 UAV Movement Candidacy Based on Speed

Figure 23 shows the candidate spaces for UAV1 at its current location. The speed of the UAV is indicated by the length of the red arrow vector pointing out of it. Based on this distance, the highlighted spaces make up the candidate locations for the next time-step. The UAV can either continue to loiter in its current location or it can travel to any location that its speed will permit.

For each of these candidate spaces, the World is altered by moving the UAV to that space, and, as in the "Run Simulation" module, a network prediction is made. Unlike the "Run Simulation" module, the algorithm must rely on a prediction. For a given UAV movement, the prediction of the network overlay will work on perfect information. The network connectivity will have no uncertainty because the positions of all friendly units will be known. However, the UAVs have no knowledge of the traffic that will be sent, and can therefore have no knowledge of the score they will obtain from positioning themselves in a given location. Instead they will work on a network traffic model. For this iteration of the research, however, the prediction model that is implemented gives the UAVs perfect information. It will be left to future work to implement a more robust prediction model that brings in elements of the NTO to provide this level of tactical planning. By using a "perfect information" model, this research will determine a baseline for an effective algorithm, independent of the underlying mechanism for network prediction, which is another large area of research.

Furthermore, the UAV system will rely on an HVT-movement prediction model to similarly determine the projected mission effectiveness score. As was the case for the network prediction model, the HVT prediction model will have perfect information. Movement models can be very complex and introduce a high level of complexity that

71

may confound the results of the simulations. By leaving this module at a level of perfect

information, a baseline can be established for the format of an effective algorithm.

The "prescribed algorithm" uses these modules as tools to determine the outcome of

the decisions it makes. Ultimately, the tools will allow the algorithm to determine which

UAV placements will benefit the system in the best way. Whether the algorithm is

deterministic or stochastic, whether it performs network tests once or performs multiple

predictions on the same candidate solution, these tools define the underlying structure to

any algorithm that is applied.

4.3 Analyzing Utility

The dynamic programming (DP) algorithm defined in Chapter 3, a basic depth-first

with backtracking (DFS-BT), and a Monte Carlo simulation were implemented in the

framework defined in this research using the methodology defined in Chapter 3. The

expected result was for the Monte Carlo solution results to bound the DP algorithm on

the lower end and for the DFS-BT solution results to bound the DP algorithm on the

upper end. The results are discussed here.

4.3.1 Analyzing the Total Utility

The total utility, $v_{placement}$, is the primary measure of merit to assess the success of the

algorithm. When the $v_{placement}$ is measured from the DP algorithm to the Monte Carlo and

DFS-BT, and we stack the results adjacent to each other, from Monte Carlo to DP to

DFS-BT, we should see an upward trend at every level. The results however, do not

reflect this trend, as Table 5, Table 6, Table 7, and Table 8 show. These tables are

displayed on the next few pages.

One of the issues that came up in experimentation was that DFS-BT was so computationally complex that it was not possible to run multiple simulations on it. It had been previously thought that the experiment sets DFS-01-01, DFS-01-05, DFS-05-01, and DFS-05-05 would be able to complete in a reasonable amount of time and the results could be extrapolated to have results that could be compared even at the larger scale. The DFS-BT must make a decision for each UAV and at a given time-step, the number of permutations of UAV placements is very large. Then, for each of those single-time-step-single-permutation options, an entire set of options for future UAV-schedules exists. It is not possible within the amount of time allotted for this research to obtain results for DFS-BT to gain enough confidence in the averages to report on its performance. In fact, we were only able to collect 5 results for DFS-01-01, 5 results for DFS-01-05, and only 2 results for DFS-05-05. The next few pages of results will only show a few of the results for DFS-BT simulations.

Table 5 Total Utility with 1 HVT

73
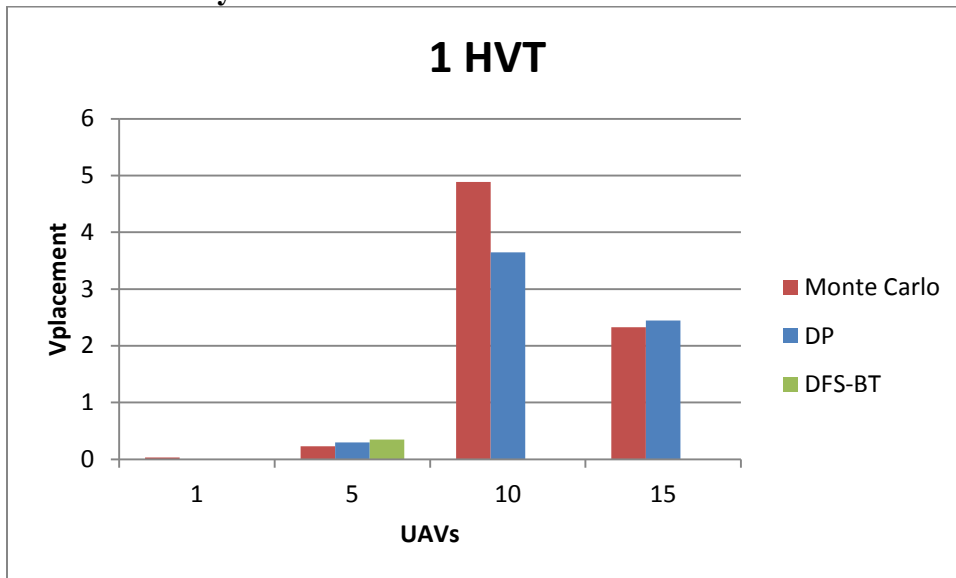
Table 6 Total Utility with 5 HVTs
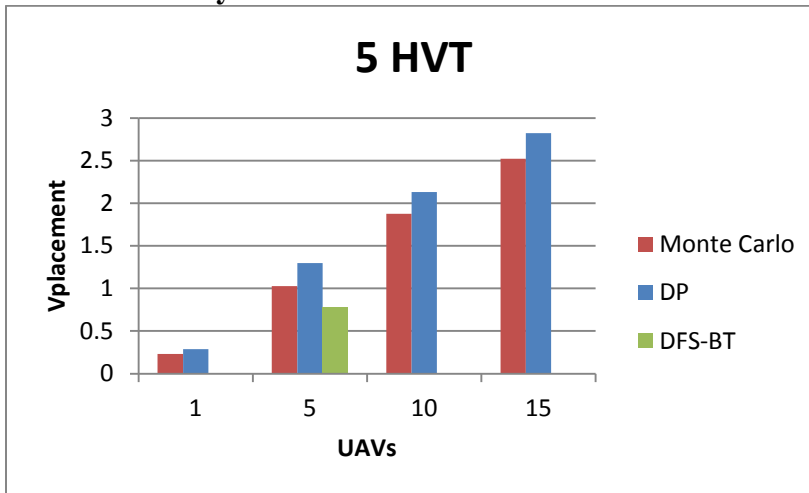
Table 7 Total Utility with 10 HVTs
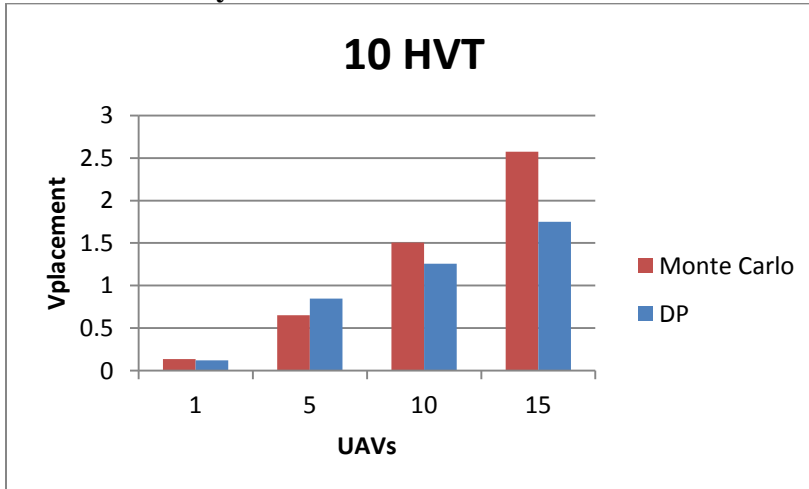
Table 8 Total Utility with 15 HVTs
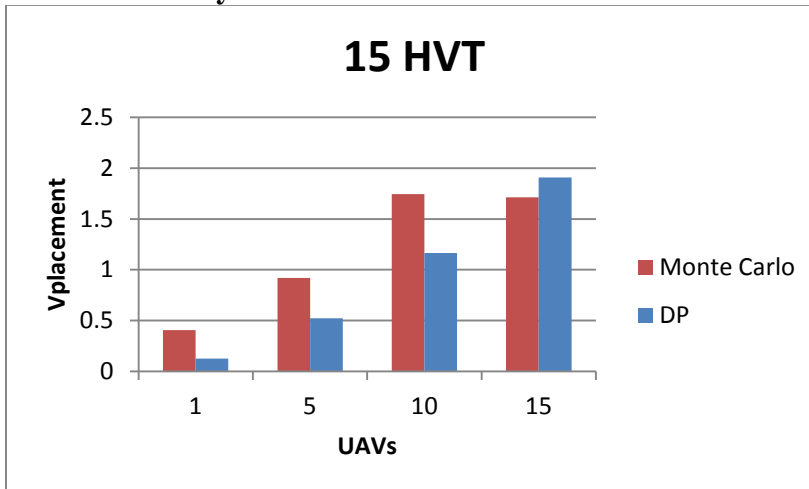
4.3.2 Conjecture on the Effect of Mission Effectiveness of Small Numbers of UAVs or Small Numbers of HVTs

The first unexpected characteristic is that the Monte Carlo utilities are greater than the DP utilities for many of the scenarios. In the scenarios that offer 1 HVT we see that the comparisons can be sporadic. The DP algorithm is marginally better at 5 and 15 UAVs. The Monte Carlo simulation performs much better at the 10 UAV level. In fact, across the spectrum, when there is only 1 UAV or only 1 HVT, the results are nearly unpredictable. We conjecture that with scarce UAVs, bad initial placement may have a very large effect on the outcome, causing sporadic results despite any other conditions. If this conjecture is true, it underlines the importance of good intelligence and its importance on the outcome of military operations.

We may also draw the conclusion that with a small number of UAVs it is difficult to achieve video surveillance coverage because of the scarce opportunities to view it. With limitations on the traveling distance of UAVs, a video track from one UAV may need to "pass off" responsibly to a UAV in the next zone. If no UAV exists in that zone or the UAV in the adjacent zone is busy, the target will be missed.

For small numbers of UAVs or small numbers of HVTs it is difficult to draw conclusions about the simulation because it is difficult to determine where the loss of performance occurred. It may have been the case that the targets were missed because there never existed opportunity or it may be the case that the targets were missed because the algorithm was designed poorly, but these cases are indistinguishable at the lowest level. The scenarios with 5, 10, and 15 UAVs and HVTs should provide more information.

4.3.3 Conjecture for Poor DP Algorithm Performance

One key characteristic that differs between the data sets is indicative of an underlying

problem in the data. As expected, the Monte Carlo simulations trend upward at a very

consistent rate, both at the number of HVTs increase and as the number of UAVs

increases. However, this is not true of the DP algorithm results. At the 5 HVT level, the

DP utility increases steadily, but under-performs heavily throughout the 10 HVT and 15

HVT levels. The Monte Carlo results plateau at the 15 UAV/15HVT level, as expected,

because of over-saturation of the map with respect to both UAVs and HVTs. When we

take a closer look at the data, beyond what is displayed in these total utility graphs, we

see that the DP-algorithm results are missing an entire component that the Monte Carlo

data includes. The trend exists in all data sets, but we will examine the 15 UAV/ 15 HVT

levels to show the pronounced difference.

Table 9 Component Utilities for Monte Carlo and DP at the 15-15 Level

Monte Carlo				Dynamic Programming			
Utility	Bytes Delivered	HVT Score	Links Dropped	Utility	Bytes Delivered	HVT Score	Links Dropped
1.509940989	5.59E+07	874	0	1.93125	0	1236	0
1.418625507	4.20E+07	824	0	1.959375	0	1254	0
1.758007008	1.83E+08	822	0	1.875	0	1200	0
1.852885981	1.42E+08	908	0	1.8375	0	1176	0
1.341809324	5.54E+07	772	0	1.93125	0	1236	0
1.368125461	4.84E+07	796	0	1.89375	0	1212	0
1.467775641	7.60E+07	812	0	1.89375	0	1212	0
1.557326919	6.91E+07	870	0	1.81875	0	1164	0
1.878444039	1.66E+08	856	0	1.95	0	1248	0
1.679824365	8.31E+07	890	0	1.875	0	1200	0

Table 9 shows that both algorithms report that no links were dropped, which likely means

that the stability score is not being recorded properly. Although both methods

successfully report the HVT score, only the Monte Carlo results report a network score (bytes delivered).

Despite the trend that none of the DP algorithm results included bytes delivered and all of the Monte Carlo simulations reported a score for two component utility values, the data shows that the DP algorithm stayed competitive and even beat the Monte Carlo in every case at the 5 HVT level. We conjecture that the DP algorithm, were it to report network score, would easily beat the Monte Carlo results. The data may not show the result directly, but the strong likelihood is that the DP algorithm performed so well in a single aspect of the problem that it beat the benchmark by a large margin. It will be difficult to support this conjecture, but future results could prove this point.

4.4 The Spread and Reliability of the Data

Finally, we examine the reliability of the data. In the methodology, it was suggested that a large number of simulations would help to mitigate the bias that results from initial placement. It was decided that 30 iterations would be a reasonable number for the DP algorithm because it would allow the utility data to reduce its spread while still allowing results to be obtained in a manageable amount of time. For Monte Carlo simulations which operated very quickly, 500 iterations were run for each test case. It is expected that the standard deviations would be very small for the Monte Carlo simulations and the standard deviations for the DP algorithm simulations would be a little bit larger, while still reliable. The standard deviations are reported in Table 10.

Table 10 Standard Deviation of Total Utility for Monte Carlo and DP

Monte Carlo

		UAVs			
		1	5	10	15
HVTS	1	0.0700	0.1590	1.2579	1.2579
	5	0.1548	0.2748	0.3942	0.4487
	10	0.1386	0.1764	0.2690	0.4174
	15	0.3828	0.1874	0.3780	0.2709

Dynamic Programming

		UAVs			
		1	5	10	15
HVTS	1	0.0000	0.0616	0.2803	0.4610
	5	0.0467	0.0821	0.1556	0.2319
	10	0.0078	0.0332	0.0674	0.0818
	15	0.0112	0.0208	0.0253	0.0600

The standard deviations are relatively small in comparison to the utility scores that they describe, but this is difficult to ascertain from the standard deviations themselves. To give more context, we derived 95% confidence intervals, using Student's t-Distribution, calculated from the sample mean $\overline{X}$, sample standard deviation S and sample size n [33]:

$$\overline{X} \pm t_{\alpha/2} S / \sqrt{n}$$

The t-distribution assumes that the population is normally distributed, using the variable $t_{\alpha/2}$ to describe the likelihood that the sample mean represents the true mean. The 95% confidence intervals show the range of values in which the true mean value could fall at the 95% confidence level. In order words, there is only a 5% chance that the true mean value for the total utility function falls outside of the range of values indicated in the following graphs. The numerical ranges are shown in tables below the graphs.

Table 11 95% Confidence Intervals on DP Algorithm Total Utility at 1 HVT

Dynammic Programming Total Utility with 1 HVT, 95% C.I.

	1	5	10	15
Upper Limit	0	0.264241009	3.581833911	2.362028187
Lower Limit	0	0.326097155	3.713818263	2.531305146
Utility	0	0.295169082	3.647826087	2.446666667
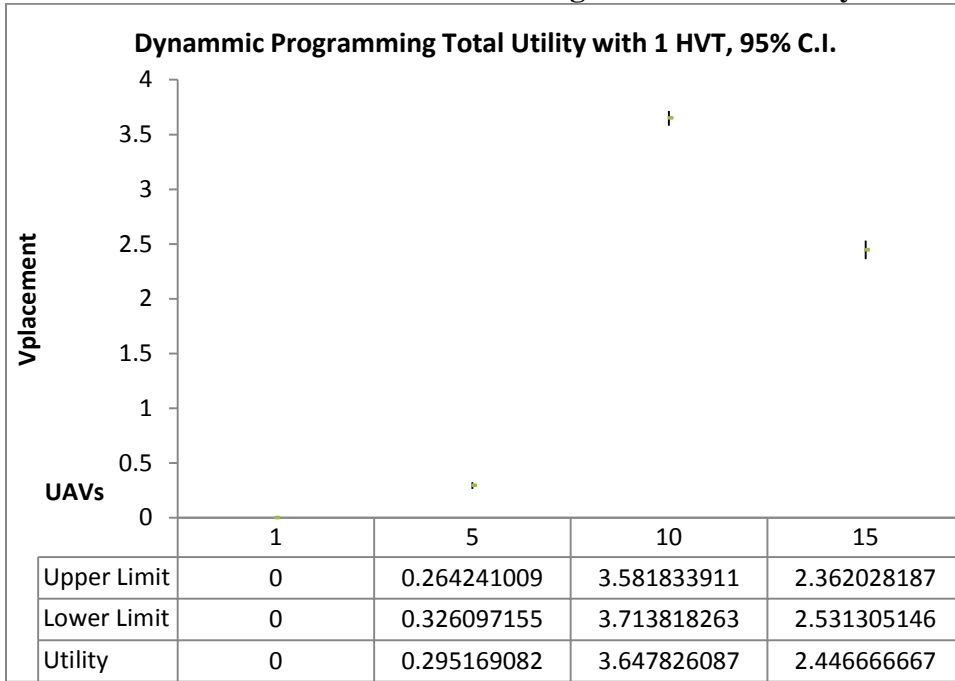
Table 12 95% Confidence Intervals on Monte Carlo Algorithm Total Utility at 1 HVT

Monte Carlo Total Utility with 1 HVT, 95% C.I.

	1	5	10	15
Upper Limit	0.025752937	0.217757334	4.852063083	2.293971065
Lower Limit	0.041714541	0.24181567	4.919724502	2.357652859
Utility	0.033733739	0.229786502	4.885893793	2.325811962
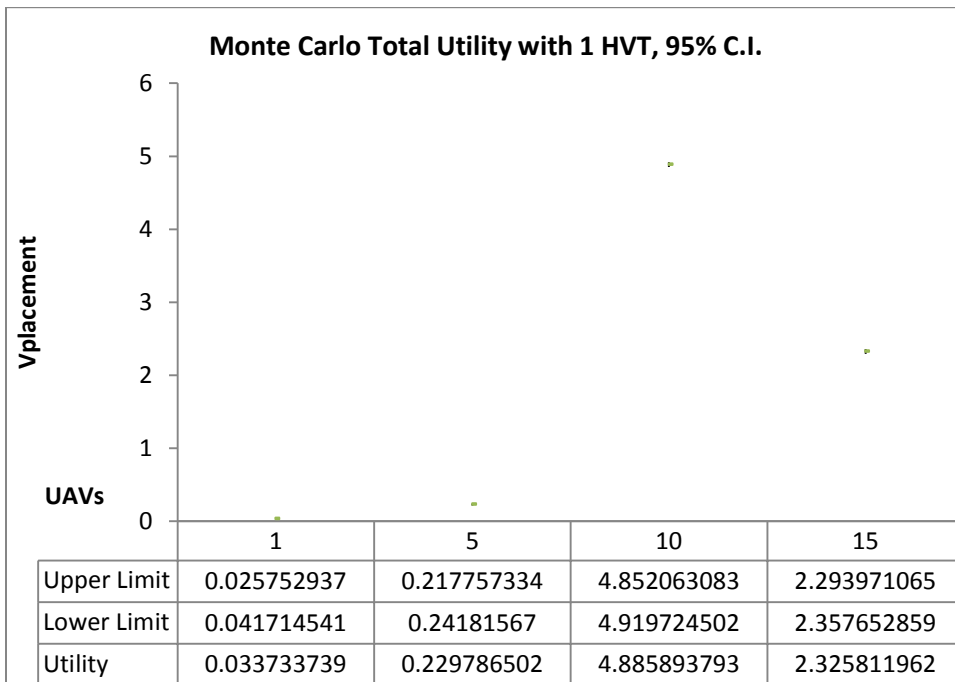
Table 13 95% Confidence Intervals on DP Algorithm Total Utility at 5 HVTs

Dynammic Programming Total Utility with 5 HVT, 95% C.I.

UAVs	1	5	10	15
Upper Limit	0.258807599	1.261141157	2.083189654	2.761229596
Lower Limit	0.312689985	1.33257865	2.181544646	2.881282481
Utility	0.285748792	1.296859903	2.13236715	2.821256039
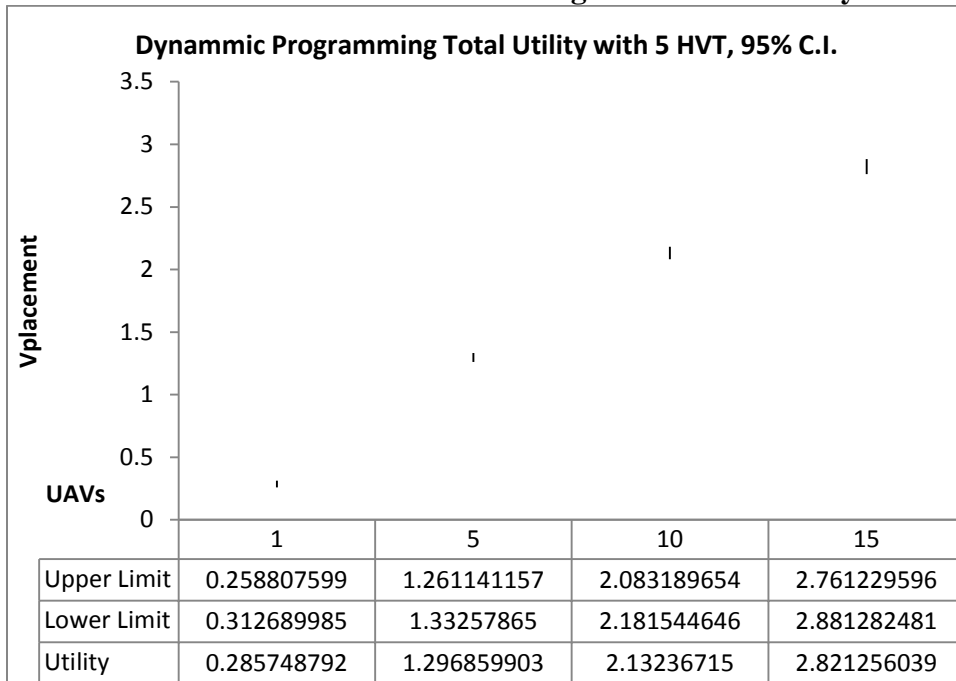
Table 14 95% Confidence Intervals on Monte Carlo Algorithm Total Utility at 5 HVTs

Monte Carlo Total Utility with 5 HVT, 95% C.I.

UAVs	1	5	10	15
Upper Limit	0.217860275	1.009585317	1.857679955	2.501522312
Lower Limit	0.241597803	1.041209058	1.895555006	2.541932531
Utility	0.229729039	1.025397188	1.876617481	2.521727422
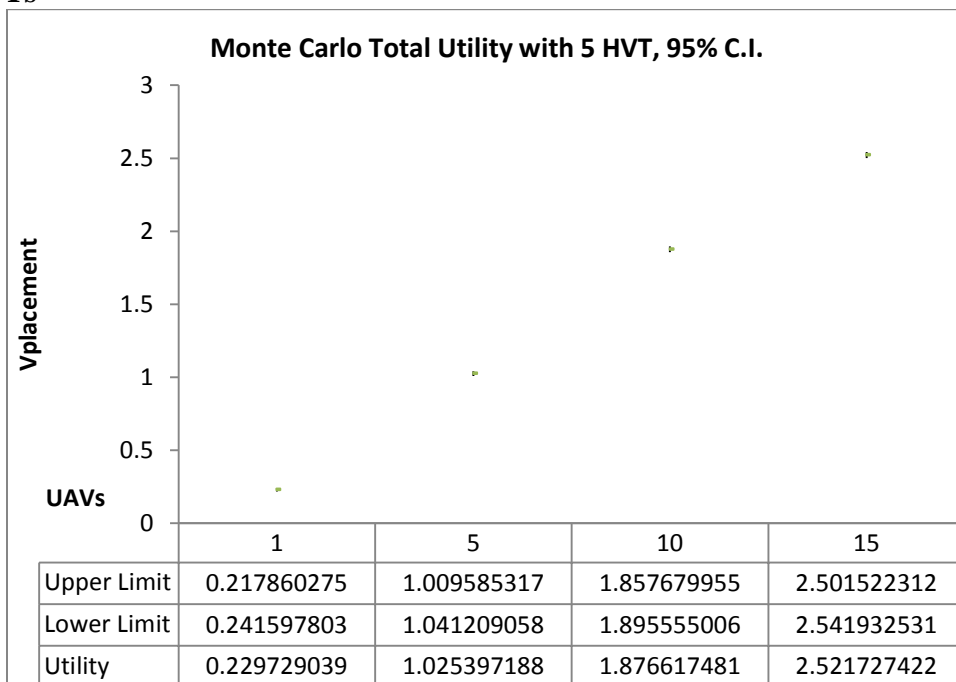
Table 15 95% Confidence Intervals on DP Algorithm Total Utility at 10 HVTs

Dynammic Programming Total Utility with 10 HVT, 95% C.I.				
UAVs	1	2	3	4
Upper Limit	0.108754956	0.823781469	1.224452539	1.714589917
Lower Limit	0.130745044	0.869200987	1.289198255	1.785886274
Utility	0.11975	0.846491228	1.256825397	1.750238095
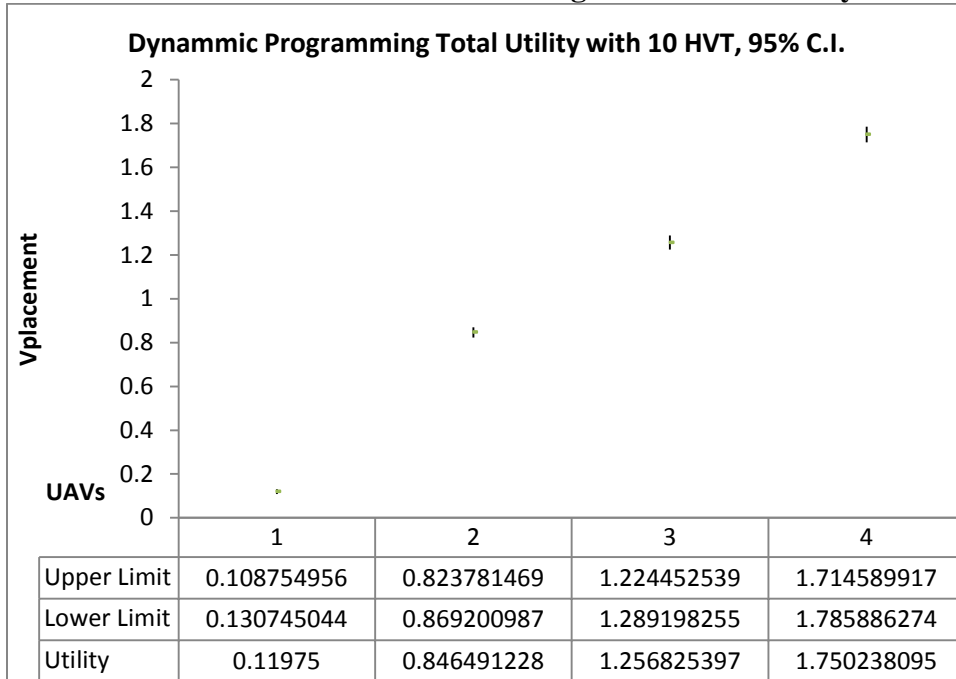
Table 16 95% Confidence Intervals on Monte Carlo Algorithm Total Utility at 10 HVTs

Monte Carlo Total Utility with 10 HVT, 95% C.I.				
UAVs	1	2	3	4
Upper Limit	0.124198996	0.640464993	1.49047679	2.554926532
Lower Limit	0.146657182	0.66580194	1.521768412	2.593900989
Utility	0.135428089	0.653133467	1.506122601	2.574413761
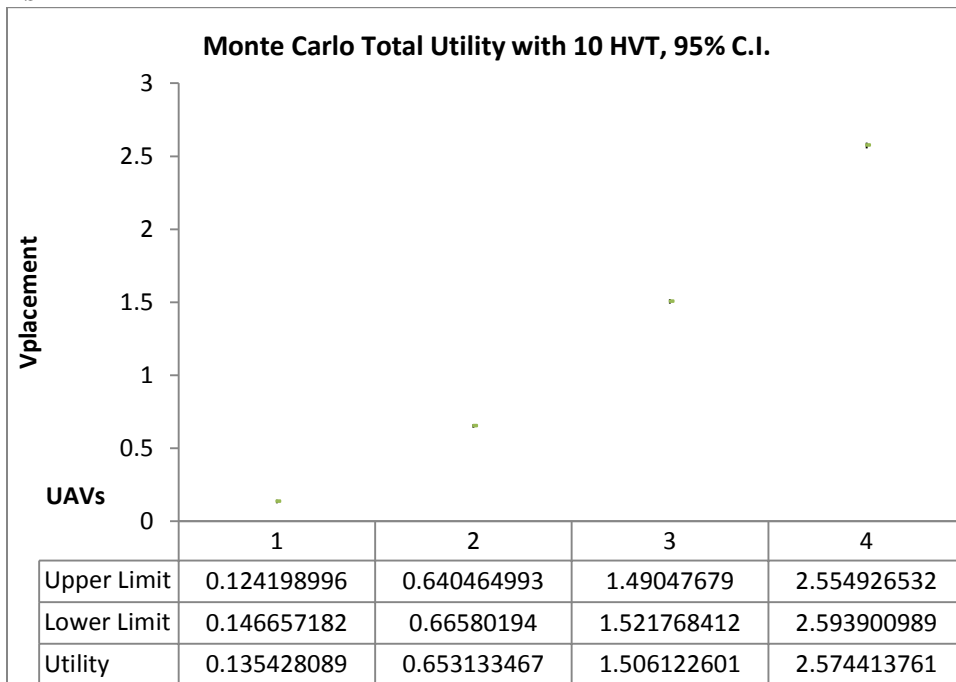
Table 17 95% Confidence Intervals on DP Algorithm Total Utility at 15 HVTs

Dynammic Programming Total Utility with 15 HVT, 95% C.I.

	1	2	3	4
Upper Limit	0.114119031	0.506282722	1.145953323	1.876703756
Lower Limit	0.140545692	0.542259552	1.185602232	1.937796244
Utility	0.127332362	0.524271137	1.165777778	1.90725
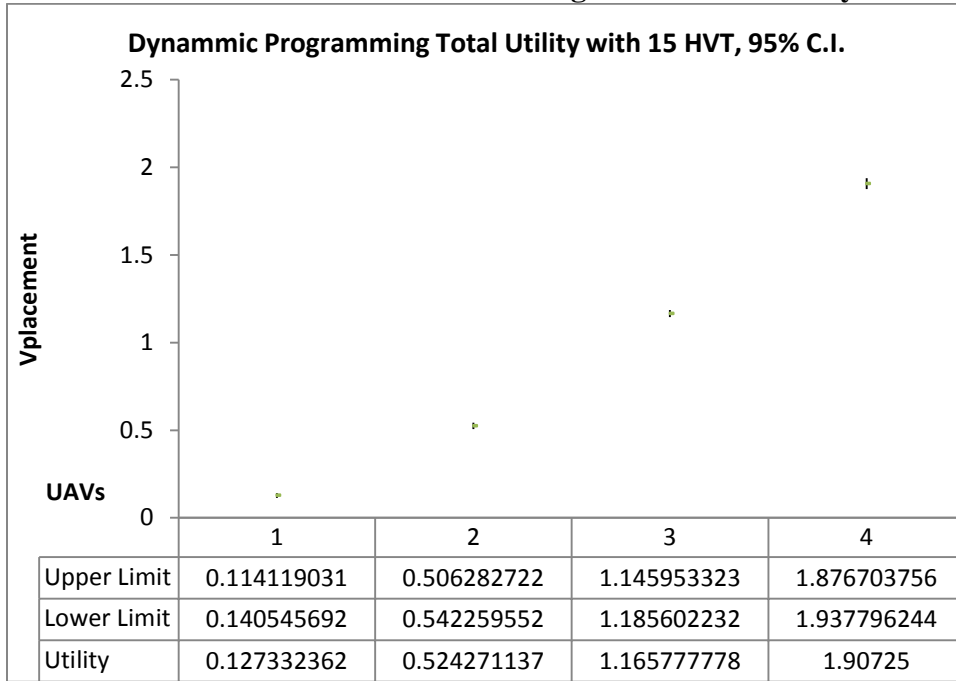
Table 18 95% Confidence Intervals on Monte Carlo Algorithm Total Utility at 15 HVTs

Monte Carlo Total Utility with 15 HVT, 95% C.I.

	1	2	3	4
Upper Limit	0.387206122	0.905983705	1.726022439	1.697910853
Lower Limit	0.424532763	0.932101142	1.763113303	1.729310396
Utility	0.405869443	0.919042423	1.744567871	1.713610625
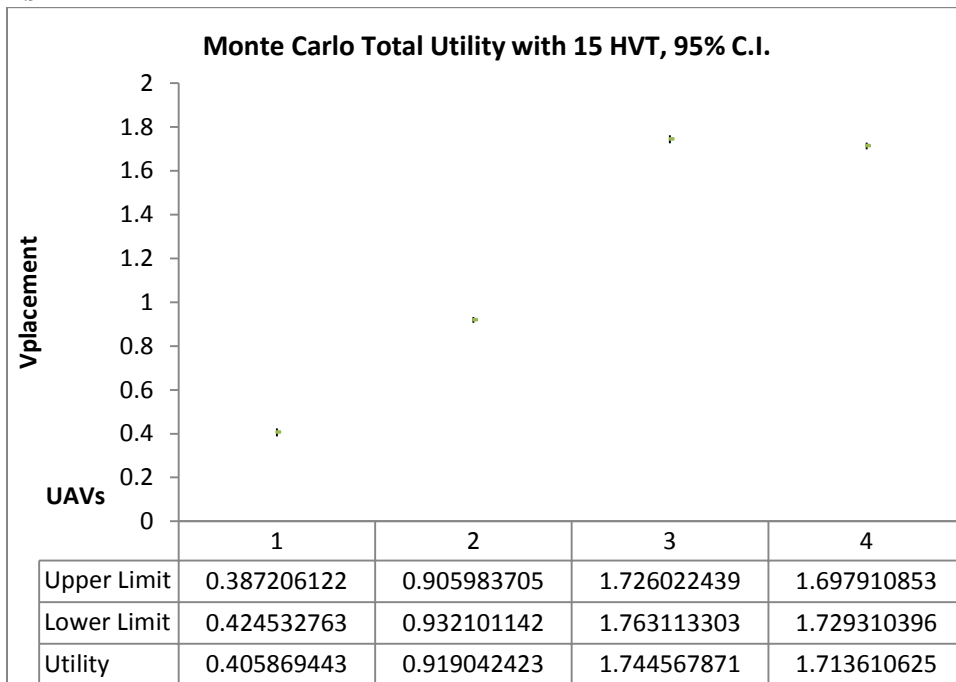
What we notice is that the confidence intervals for the Monte Carlo simulations are extremely small meaning that there is a very high confidence in the mean values that

were reported by the simulations. As expected, the 500 iterations helped to mitigate the bias that may have resulted from the stochastic initial placement. Were there a less reliable initial state mechanism, it may have lead to even larger ranges of values, but using the baseline initial placement mechanism we have high confidence in our results.

The dynamic programming solutions, even with only 30 iterations were able to give more stable outputs than the Monte Carlo simulations. This was expected because the decision making ability of the algorithm tends to give high consistency to the results. It was for this reason of solution consistency that a deterministic method was chosen. Since the spread of the sample data is small, the resulting 95% confidence interval for the spread of the data is also small. Even at the most rigorous level with the most uncertainty, at 15 UAVs with 15 HVTs, we observe a spread that is only 3.2% of the mean value. The spread of data supports the conjecture that a deterministic algorithm returns predictable, reliable results.

4.5 Summary

The simulation test bed designed for this research is robust and capable of supporting a plethora of different algorithms. For this research, we employed a dynamic programming algorithm, a depth first search with backtracking, and a Monte Carlo, using the tools that are the core of this test bed.

Using realistic data, based on operational conditions, the results showed promise that the deterministic DP algorithm would strongly outperform its benchmark. It was capable of following HVTs to a high degree, and pending the introduction of network data, the algorithm would likely report high utility values, as well.

From the data that was collected, there is high confidence that the deterministic algorithm is a suitable choice for real world operational missions in which UAVs are needed. Small confidence intervals in the dynamic programming results show that the results the algorithm gives have a high level of consistency, which is a strongly desired trait in a planning tool. The utility values show that it is potential to be effective at performing the dual-role.

V. Conclusions and Future Work

This research focused on the prospect of enhancing the use of UAVs by placing them in a network-oriented/mission-oriented dual role. It was suggested that one of the greatest obstacles to realizing this potential for highly utilized UAVs is having the operational and tactical planning necessary to allow large numbers of UAVs to work cooperatively. To build on previous research that developed the idea of a Network Tasking Order (NTO), concrete planning methods are needed. To fulfill the need for concrete planning methods that accomplish a dual-role for UAVs, an operational plan was suggested, along with a mathematical model to support it.

That mathematical model discretized the geographic space to scale back to computational complexity, modeled the environment with high-value targets, UAV entities, and a network model, and introduced dynamics that would govern that environment. Furthermore, a simulation test-bed was developed that implemented that mathematical model and created a method of implementing new algorithms.

Furthermore, this research suggested a deterministic dynamic programming algorithm to compromise between network performance and stability and mission effectiveness. Novel heuristics and problem simplifications implemented in this algorithm allowed it to operate in a relatively short amount of time, while still generating excellent UAV schedules. The algorithm's simplicity, speed, and accuracy make extremely operationally relevant.

We suggest that, with only a few more improvements, the algorithm should be used in forward deployed locations, alongside the expertise of human operational planners, to

present this novel potential for dual-role UAVs. The problem formulation, simulation test-bed, and algorithm serve as good first steps towards realizing a tool that will dramatically improve mission effectiveness, and potentially change the way we fight.

5.1 Future Work

The next iteration of algorithms for UAV placement must consider the consequences of short flight time. The reality of small UAVs is that they have a capacity for significantly less loiter time, and it can be a difficult challenge to maintain coverage of a particular target in the absence of a sensor's continuity. The current utilization of small UAVs is generally confined to "over the berm" surveillance, at a highly tactical level. Much of the motivation for using large UAVs, such as the Predator and Global Hawk, is that they can have flight times as high as 40 hours. The amazing flexibility of this capability naturally broadens its utilization. Improving hardware has the potential to mitigate this problem, but realistically, we may never see a Raven that can loiter for 40 hours, even if the need were to exist. A Raven can fly up to 80 minutes, and the U.S. Army's Shadows and Hunters can easily achieve over 3 hours of flight time. Though the flight times may be modest, their logistical challenges are far fewer, and refueling via "touch-and-goes" is a distinct possibility. An algorithm that can account for flight time as a parameter, and can provide a mechanism for planning to refuel, will create more realistic results for routing in swarms of small UAVs.

Also, although the simulation test-bed was originally intended to support both discretized and continuous space, it currently only supports the former. By opening the simulation-test bed to a continuous space, it may make the test bed lean a little closer to more mainstream, general purpose simulators, but it may also increase the breadth of the

problem and present challenges that are truer to life. One of the limitations of the discretized space is that there some sampling error that occurs with positioning. In the grid world, it is assumed that a UAV will be hovering over a particular area for a period of time. But by the assumption that the UAV is not moving at all in the simulation, it may violate the agreement that the UAV would be in a certain position to support the network. Continuous space would open the problem space and reduce some of the sampling error.

Research on improved algorithms to place UAVs should continue. This research presents a deterministic algorithm, but perhaps some stochastic methods could improve the results. Evolutionary algorithms or simulated annealing techniques may do a better job of exploring the search space and avoiding the pitfalls of greedy choices that sacrifice long-term gains. As is the case with all algorithms, the results may vary based on the domain. The assumptions made, and the methods that resulted from those assumptions, during this research were based on the environment in Afghanistan and Iraq where the areas of interest are sparsely populated and there is little radio interference. Perhaps in a physical environment like China or India, where the population is much denser and the amount of interference is higher, different planning techniques or different algorithms may be better suited. The downside to a stochastic algorithm is that the characteristics of the resulting solution may be harder to predict, and thus more difficult to enforce consistency of quality, but the solution may be produced more quickly or with a greater breadth of the problem space.

Finally, Incremental changes to the current simulator are needed. For instance, the utility normalization can be improved with more accurate bounding on the maximum

possible value. This will help to remove bias in the utility function. The decision-making currently uses perfect information about the movement of HVTs and the movement of network traffic to determine where to place UAVs. It may be a large area of research to introduce more uncertainty while preserving the quality of the results, but it will be necessary to introduce another level of reality to the simulation. Furthermore, it may reduce computation times significantly because the generation of the hypothetical next-state for a potential solution will be less computationally rigorous. The time quanta are set at 5 minutes with fixed schedules for the HVTs. A tool that can generate new scenarios based on desired characteristics of the battle environment will produce more accurate performance results by testing the algorithm under more varied conditions. And finally, the environment and signal propagation model is simple. It is assumed that there is no terrain and there are no buildings to obstruct signals. The level of planning addressed in this problem may not necessarily require such intricate planning, but consideration to the terrain may have great consequences for the routing of the UAVs. There are a plethora of ways to improve upon the simulation test bed and the algorithm itself, but the preceding areas name only a few.

Appendix A: List of Acronyms

AODV, Ad Hoc On-Demand Distance Vector
AOR, Area of Responsibility
ATO, Air Tasking Order
AWACS, Airborne Warning and Control System
BLOS, Beyond Line of Sight
C2, Command and Control
C2ISR, Command, Control, Intelligence, Surveillance and Reconnaissance
CAOC, Combat Air Operations Center
CDS, Connected Dominating Set
CONOPS, Concept of Operations
COP, Common Operating Picture
DSDV, Destination-Sequence Distance-Vector
EC2, Enterprise Command and Control
FBCB2-BFT, Force XXI Battle Command-Brigade and Below-Blue Force Tracker
FBCB2-EPLRS, Force XXI Battle Command-Brigade and Below-Enhanced Position
Location Reporting System
JSTARS, Joint Surveillance and Target Attack Radar System
FOB, Forward Operating Base
GIG, The Global Information Grid
HMMWV or "Humvee", High Mobility Multipurpose Wheeled Vehicle
IED, Improvised Explosive Device
IRC, Internet Relay Chat
LOS, Line of Sight
LS, Layered Sensing
MANET, mobile ad hoc network
MRAP, Mine Resistant Ambush Protected
MTTP, multi-Service Tactics, Techniques and Procedures
NCW, Network-Centric Warfare
NTO, Network Tasking Order
OSPF, Open Shortest Path First
RPG, Rocket Propelled Grenade
SA, situational awareness
SATCOM, Satellite Communication
TOC, Tactical Operations Center
UAS, Unmanned Aircraft Systems
UGS, Unattended Ground Sensor
VC, Virtual Circuit
VCR, Virtual Circuit Routing
WSN, Wireless Sensor Networks

Appendix B: Proofs

B-1 Solution Space Size

Given U UAVs, M locations, and T time steps, the solution space can be enumerated as follows:

- At any given time step, a UAV can be placed in M different locations. The placement of a UAV in a given spot does not prohibit the placement of another UAV in that same location because bandwidth constraints do not guarantee that the single UAV will be able to satisfy the network demand of a given set of nodes. The solution space must account for assigning a given area with more than one UAV. Thus, for one time step, there are $O(M^U)$ different UAV schedules.

- From any given schedule, all UAVs could potentially end up at a different location for the next time step. The distance that a UAV can travel depends on the parameter of their speed, but because this variable is a parameter, and not an input, it is difficult to account for this limiting factor in the solution space. A conservative estimate of the solution space is $O(M^{UT})$.

Appendix C: Data Sources

C-1 Environment and Target Data

The Caspian Sea scenario was previously used to demonstrate dynamic tactical targeting capabilities using an architecture to task a sensor network. The research was conducted by Maj. Michael Davis and Maj. Reginald Kabban to develop metrics that could be used to measure the performance of layered sensing, an information collection and exploitation system [1]. A map of the geographic area can be seen in Figure 24 Geographic Area of the Notional Scenarios used in Layered Sensing Study
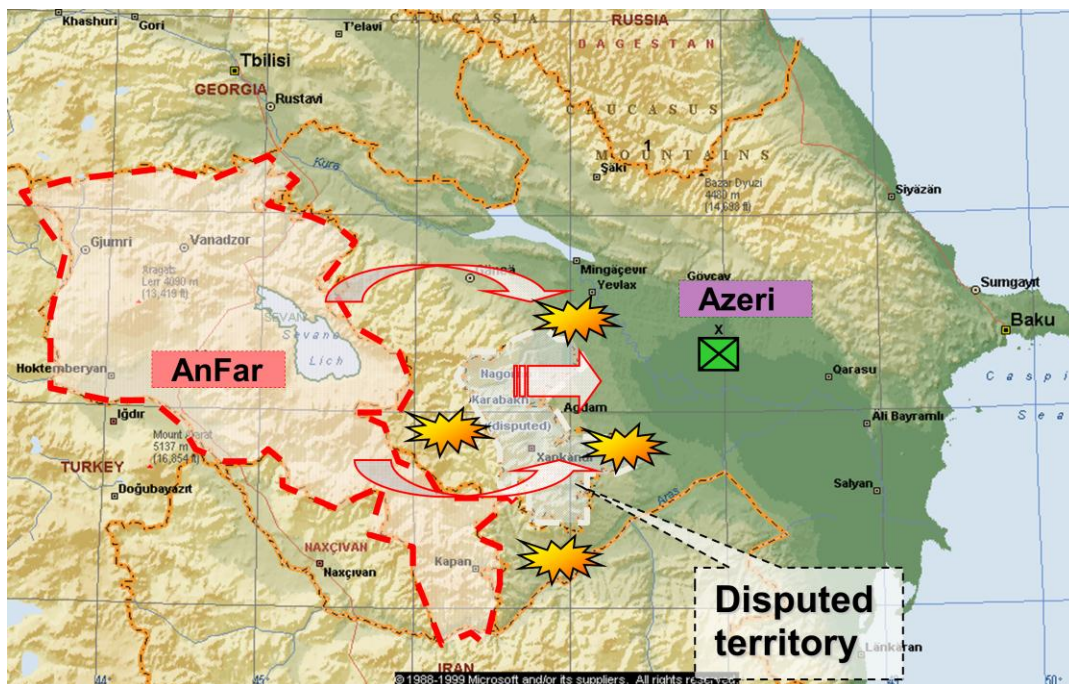

Figure 24 Geographic Area of the Notional Scenarios used in Layered Sensing Study

The researchers defined an entire notional conflict to give context to the events that occur and to add an element of realism, in terms of U.S. response. In the scenario, two hypothetical countries, AnFar and Azeri begin conflict over a disputed region. AnFar

military forces occupy the disputed region and Azeri forces attempt to limit further incursion into Azeri territory. The U.S. responds to ease tension and squelch the conflict.

The U.S. response that is modeled in the simulation is a deployment of intelligence assets, such as an assortment of UAVs, aerial C2 platforms, and unattended ground sensors (UGSs). As Figure 25 Layered Sensing Full Scenario ISR Deployment shows, the different platforms correspond to different levels of ISR responsibility. Where the JSTARS and Global Hawk are responsible for the entire region, the Predator, Shadows, and Hunters are assigned specifically to a particular urban area.
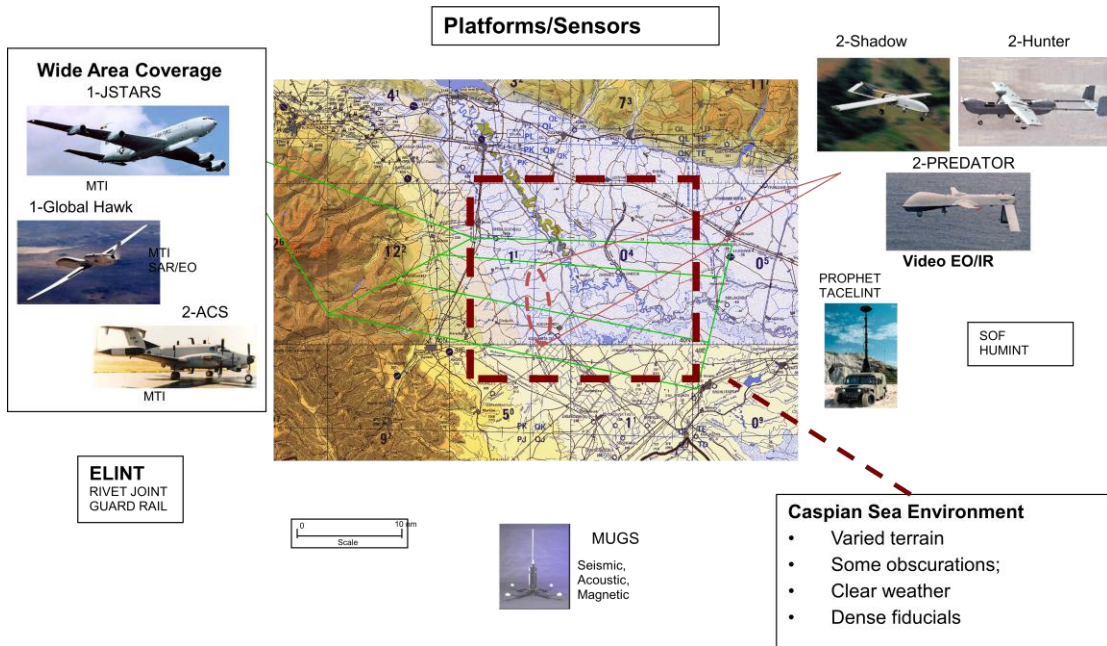

Figure 25 Layered Sensing Full Scenario ISR Deployment
The deployed ISR assets are meant to support ground forces, whose mission is to neutralize offensive threats, detect and terminate insurgent activity, and develop and maintain lines of communication. The objectives and the enemy targets can be seen in Figure 26 Layered Sensing Mission Objectives and Enemy Forces. The stated mission of the ISR assets is to develop situational awareness. Specifically, the ISR assets will

locates static and mobile IADs threats positions and conduct continuous weather reconnaissance. The ISR assets simulated in this environment are supplemented with electronic intelligence (ELINT) and human intelligence (HUMINT).
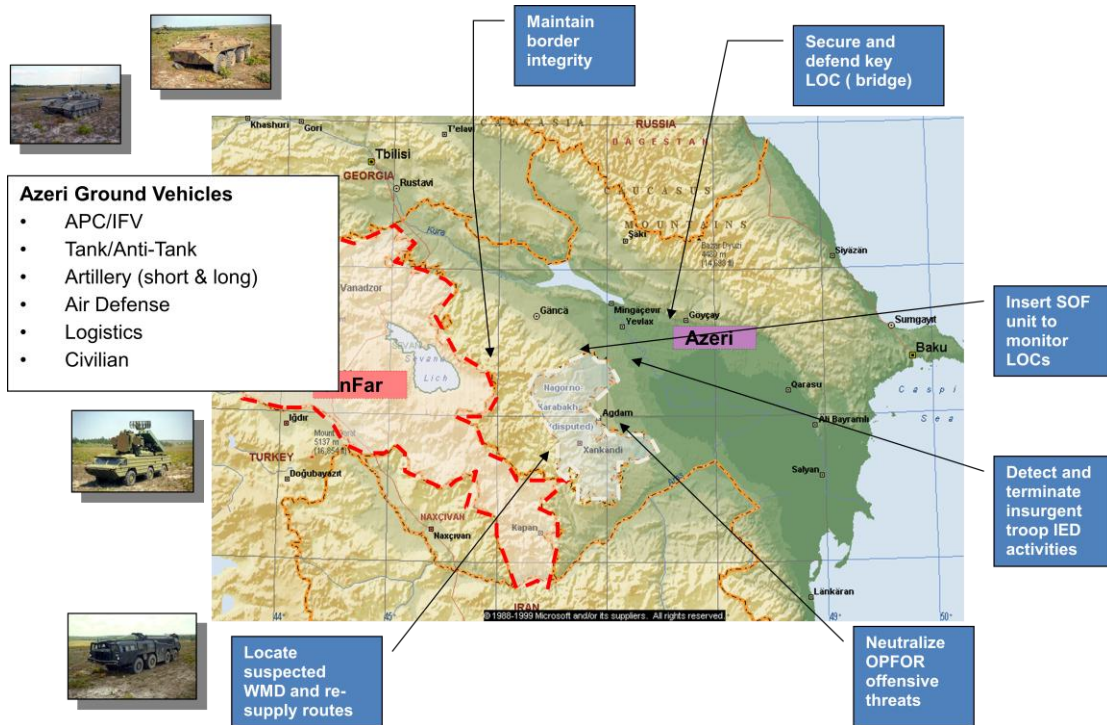

Figure 26 Layered Sensing Mission Objectives and Enemy Forces

C-2 Modeling Network Traffic

The data sets that are used to populate the network traffic are not direct data dumps from operational campaigns. Often it is the case that this data is sensitive and the contents are not able to be released. However, there are many studies on equipment and procedure that emulate operational network usage pretty accurately. Recently, in 2010, the 82nd Airborne Division at Ft. Bragg, NC performed a study on the Harris 117G tactical multi-band radio [17]. The research was meant to be a stress test on the hardware and software included in the package. To do this, the researchers included software tools

that are currently used in military operations, such as the mIRC chat client, Webmail,

UDP file transfer, Sharepoint Portal Traffic, Biometrics enrollment transfer using

TACCHAT IP, and TIGRNet. Although the network traffic was heavily Army focused,

the intent does not change. The nature of the data is not fundamentally different, nor is

the schedule on which it occurs. Included in this appendix are the traffic volume results

that were used to generate the input to the simulations in this research. Each of these data

sets shows traffic over a four minute period. It was simply assumed that this traffic

pattern would continue to loop in the same fashion over the course of the simulation, and

the data sets that were used repeated the load for the duration of the simulation.
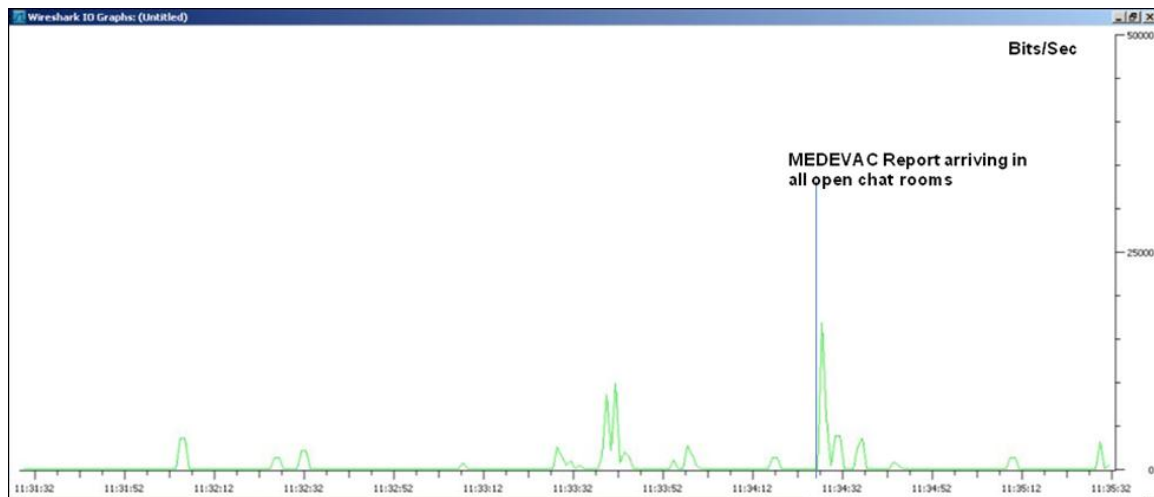
Figure 27 mIRC Traffic Analysis

The mIRC application is a chat client. It is commonly used by tactical operators who

are physically on the battlefield, commanders stationed at nearby command posts, and

information analysts also located near the battlefield. The tool allows these stakeholders

to discuss information about the environment and coordinate plans of action. This traffic

is typically low volume because it is simply text-based.

The SIPR web-mail access entails application-level key and password authentication, as well as the transfer of messages. The translation of this logged information into the data set entailed single login attempts by ground users, following by intermittent e-mail accesses by those users. It was assumed that only half of the ground users would need e-mail access within the given period, but that those users would access multiple e-mails over the course of the simulation. The amount of traffic generated from these specific actions can be seen in Figure 28.
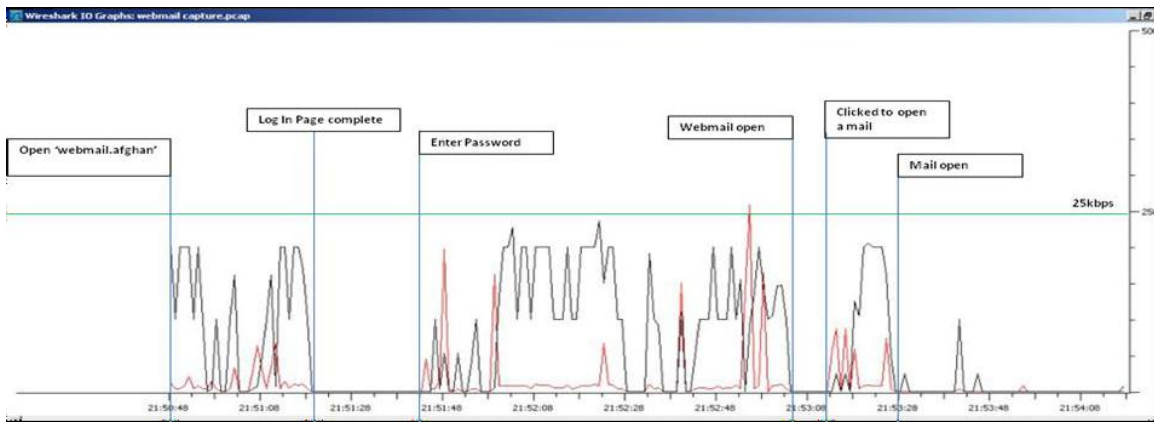

Figure 28 SIPR Webmail Traffic Analysis

TACCHAT IP is an Army chat client that enables UDP file transfer, as depicted in Figure 29. Only single file transfer was included over the course of the simulation.


Figure 29 Map File Transfer (TACCHAT IP) Traffic Analysis

SharePoint access shows normal web traffic that soldiers may need to access to receive shared data. The pattern of this traffic over the course of a 4 minute period is shown in Figure 30. This volume of traffic was repeated continuously, looping throughout the entire simulation.
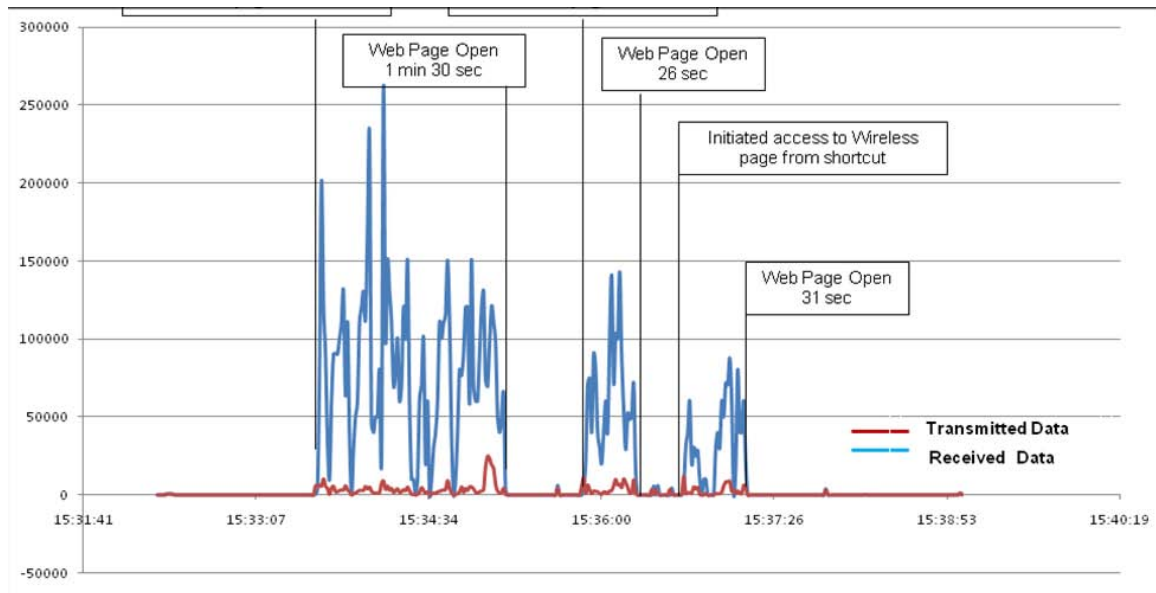

Figure 30 Microsoft SharePoint (Portal) Traffic Analysis

TIGRNet is an Army tool for sharing battlefield information is shown in Figure 31. This traffic is repeated continuously.
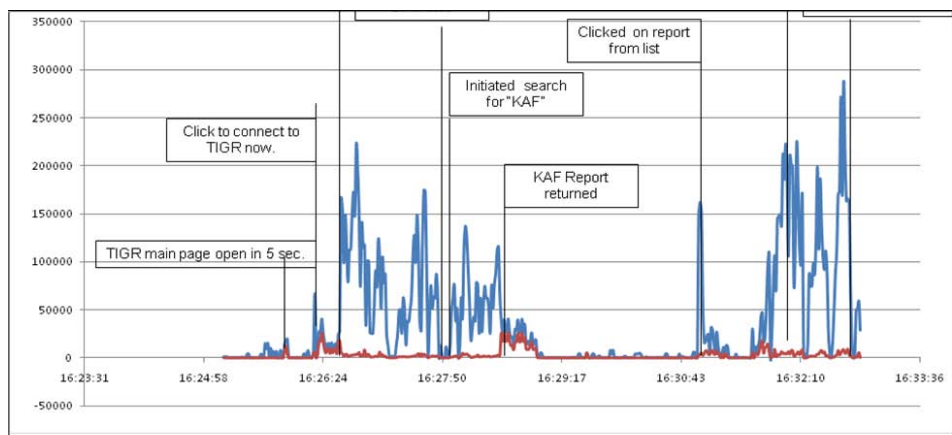
Figure 31 TIGRNet Traffic Analysis

The previous sources describe web traffic that originates from ground nodes during the simulation. Figure 32 shows a traffic pattern resulting from full motion video that is captured by a Predator UAV. In the course of the exercise performed in [17] the video quality is up-converted halfway through. In this simulation, it was simply assumed that the video would stay at one level of quality. For every HVT that was observed, the assigned UAV would produce 96kbps of data, as shown in Figure 32.
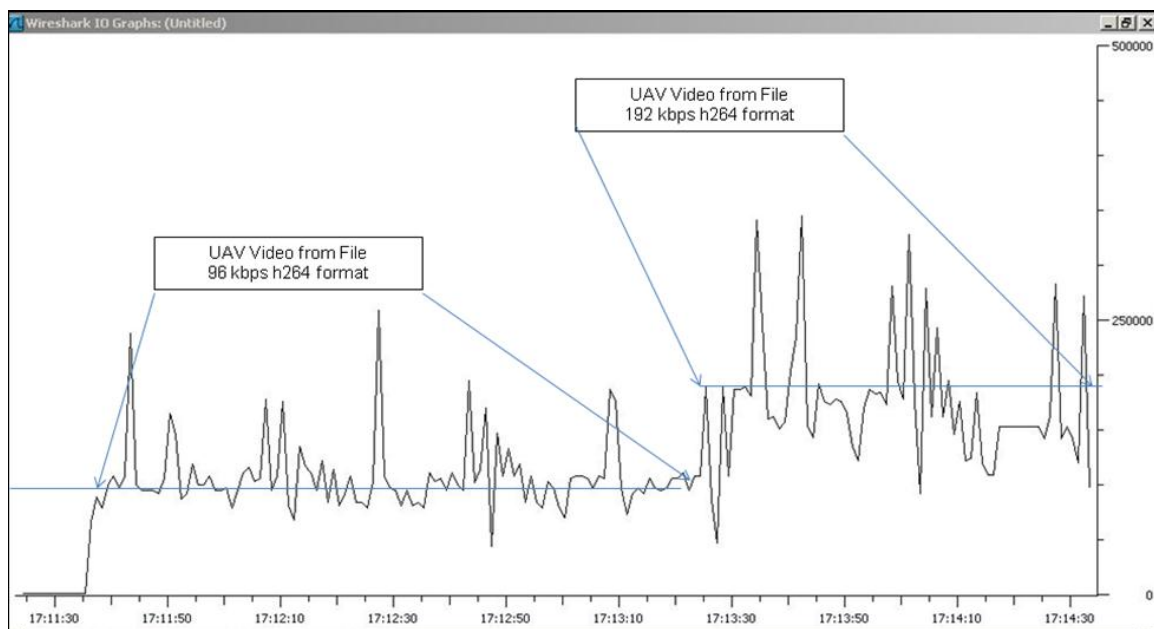
Figure 32 Streaming Video Traffic Analysis

To align with the parameters of the problem space, the traffic was arbitrarily assigned priority by traffic type. UAV video traffic was assigned priority 1, TIGRNet traffic was given priority 1, mIRC traffic was given priority 2, webmail wag given priority 3, TACCHAT file transfer was given priority 4, and SharePoint access was given priority 5.

Appendix D: Greedy Algorithms

According to [32], a *greedy algorithm* always makes the choice that looks best at the moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution. Greedy algorithms do not always yield optimal solutions, but for many problems they do.

The authors in [32] go on to define a generalized sequence of steps that can be used to determine whether or not a problem can be solved using a greedy algorithm and how to generate that algorithm. The steps are the following:

1. Cast the optimization problem as one in which we make a choice and are left with one sub-problem to solve.

2. Prove that there is always an optimal solution to the original problem that makes the greedy choice, so that the greedy choice is always safe.

3. Demonstrate that, having made the greedy choice, what remains is a sub-problem with the property that if we combine an optimal solution to the sub-problem with the greedy choice we have made, we arrive at an optimal solution to the original problem.

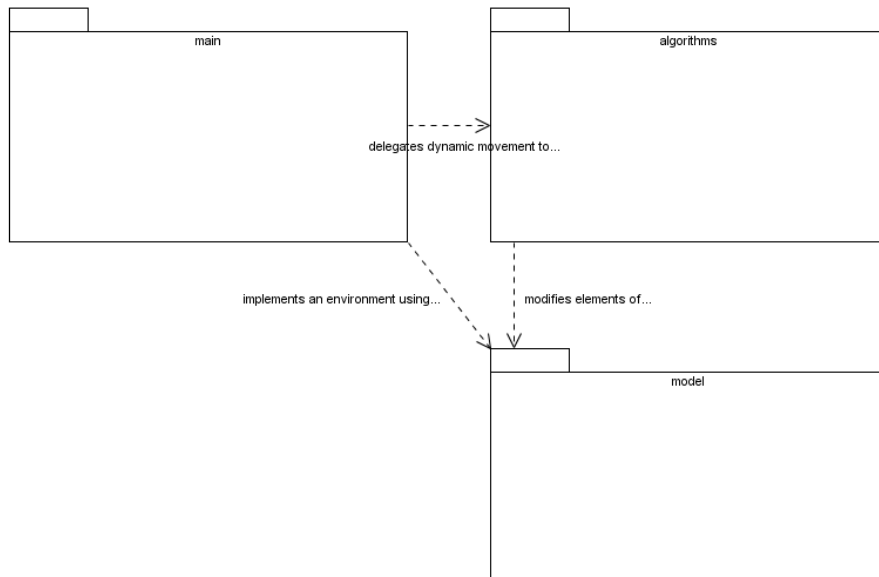It should be noted that this definition and these steps apply to optimization problems in which a greedy algorithm can achieve "the best" solution. In the case of this research, a greedy algorithm cannot provide the best solution, but the hope is that a greedy algorithm gets close.
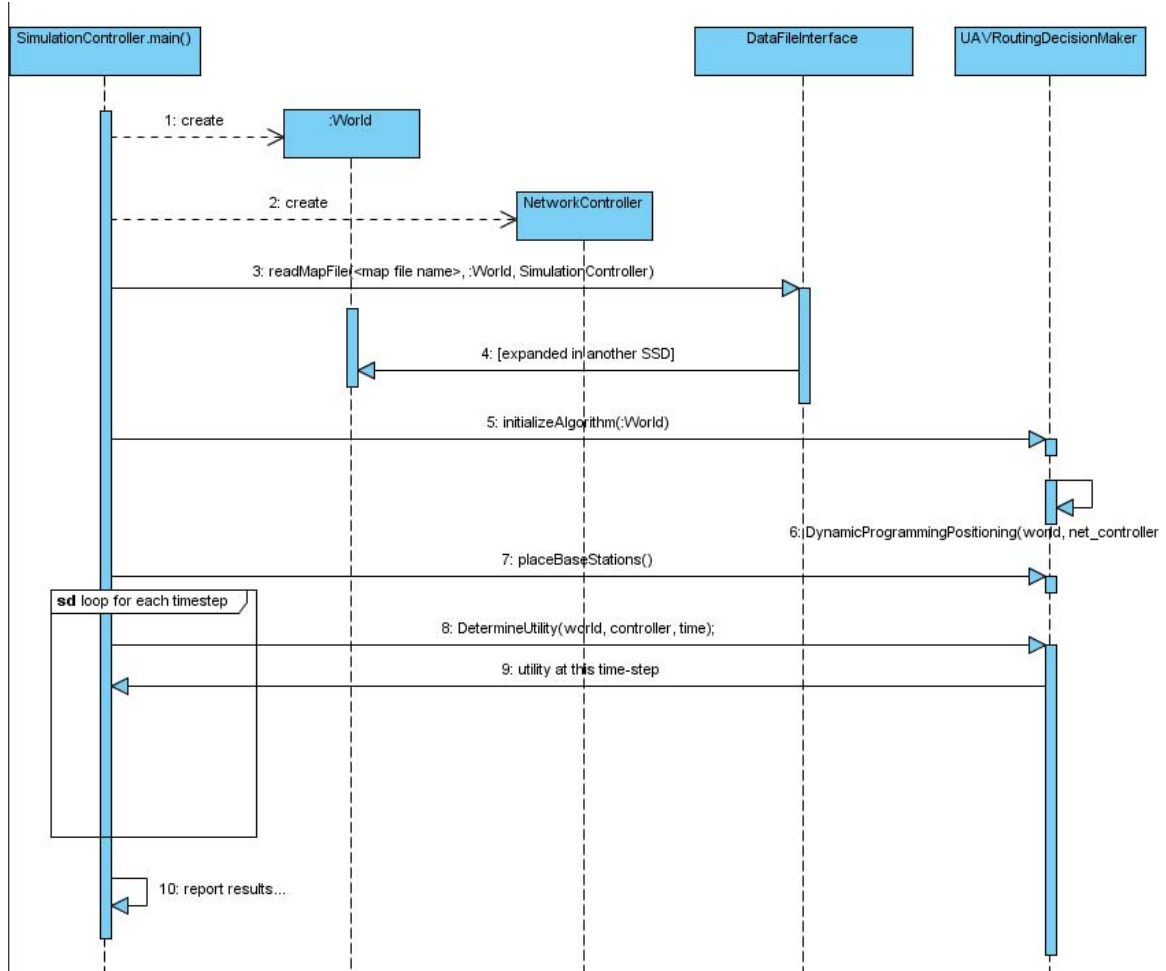
Appendix E: Simulation Test Bed

E.1 World Model
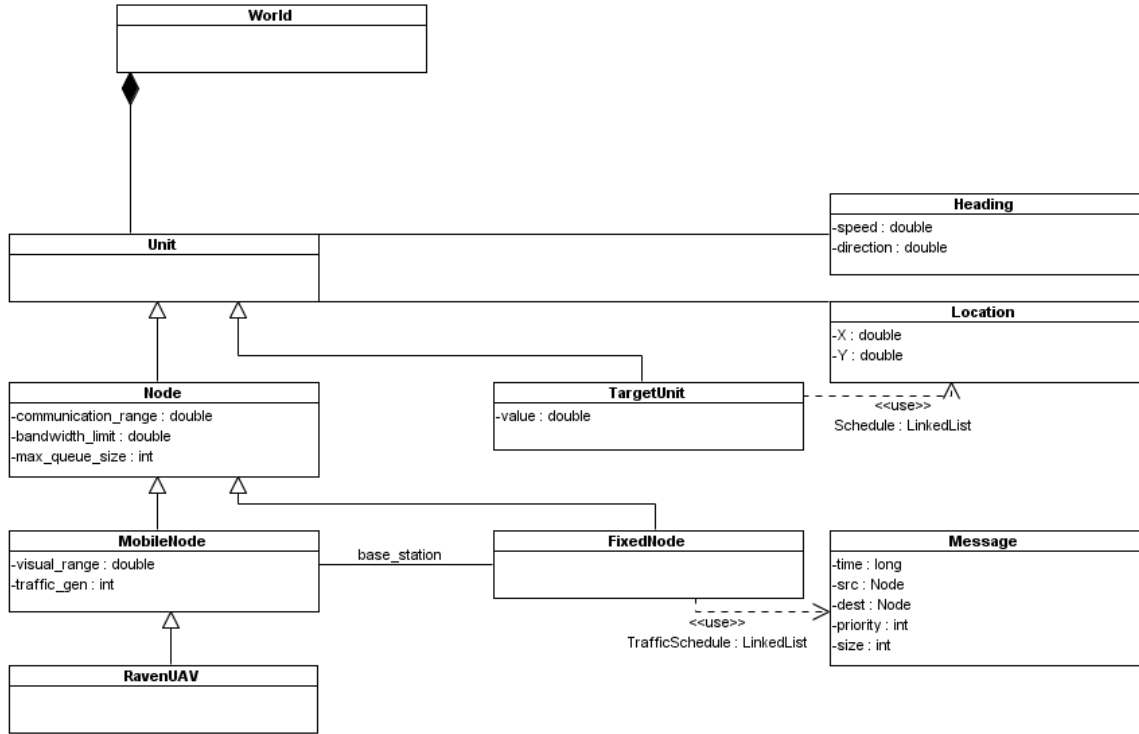
E.2 Package Overview

E.3 High Level Sequence Diagram of the Program Flow

VI. Bibliography

[1] Michael Davis and Reginald Kabban, "Developing Metrics for Evaluating Competing Layered Sensing Architectures," Wright-Patterson AFB, OH, Graduate Research Paper 2010.

[2] Matthew T. Seibert, Andrew J. Stryker, Jill R. Ward, and Chris T. Wellbaum, "System Analysis and Prototyping For Single Operator Management of Multiple Unmanned Aerial Vehicles Operating Beyond Line of Sight," Air Force Institute of Technology, Wright-Patterson AFB, OH, Master's Thesis AFIT/GSE/ENV/10-M01, 2010.

[3] Jr, Paul W. Phister and John D. Cherry, "Command and Control Concepts within the Network-Centric Operations Construct," in *Aerospace Conference, 2006 IEEE*, Big Sky, MT, 2003, pp. 1-9.

[4] Jay S. Bayne, "A Theory of Enterprise Command and Control," in *Military Communications Conference, 2006. MILCOM 2006. IEEE* , Washington, DC, 2006, pp. 1-8.

[5] W. E. Maj. Gen. USMC, Vice Director, Joint Staff GASKIN, "GLOBAL COMMAND AND CONTROL SYSTEM COMMON OPERATIONAL PICTURE REPORTING REQUIREMENTS," Washington, DC, 2008.

[6] Office of the Secretary of Defense, "The Implementation of Network-Centric Warfare," U.S. Government Printing Office, Washington, D.C., Report 2005.

[7] Department of Defense Directive, Global Information Grid (GIG) Overarching Policy, September 19, 2002.

[8] John G. Grimes, "Department of Defense Global Information Grid Architectural Vision," U.S. Department of Defense, Washington, D.C., 2007.

[9] Masoumeh Karimi and Pan Deng, "Challenges for Quality of Service (QoS) in Mobile Ad-Hoc Networks (MANETs)," in *Wireless and Microwave Technology Conference, 2009. WAMICON '09. IEEE 10th Annual* , Clearwater, FL, 2009, pp. 1-5.

[10] Anurag Kumar, D. Manunath, and Joy Kuri, *Communication Networking: An Analytical Approach*. San Fransisco: Morgan Kaufman Publishers, 2004.

[11] Elizabeth M. Royer and Chai-Keong Toh, "A Review of current Routing Protocols for Ad Hoc Mobile Wireless Netwoeks," *IEEE Personal Communications*, pp. 46-55, April 1999.

[12] Chris Bailey, "Department Of Defense Usage of FalconView," Atlanta, GA, 2007.

[13] Georgia Tech Research Institute. (2007) KML Support Image.

[14] Harry Greene, Larry Stotts, Ryan Patterson, and Janet Greenberg, "♣ Command Post of the Future: Successful Transition Of a Science and Technology Initiative To a Program of Record ," 2010.

[15] Amy Walker. (2009, July) www.army.mil. [Online]. http://www.army.mil/-news/2009/07/13/24299-command-post-of-the-future-program-nears-fielding-of-

6000th-unit/

[16] Deborah Tharp and Laird Wallace. (2010, December) Enhanced Position Location Reporting System: Legacy System Provides New Technology for Warfighters. [Online]. http://www.spawar.navy.mil/sti/publications/pubs/td/3155/5a_S4papers/EPLRS.pdf

[17] Michael Maj. USA Dean, "4/82 Brigade Combat Team Operational Assessment of the Harris AN/PRC-117G Falcon III Multiband Manpack Radio In OEF X," Fort Bragg, NC, 2010.

[18] Feng Zhao and Leonidas Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Fransisco, USA: Morgan Kauffman Publishers, 2004.

[19] P. Sasikumar and N. Jagadeesan, "Heuristic Approach with Probabilistic Management for Node Placement in Wireless Sensor Networks," in *International conference on Advances in Computing, Control, and Telecommunication Technologies*, Trivandrum, Kerala, 2009, pp. 734-736.

[20] Natalija Vlajic and Nelson Moniz, "Self-Healing Wireless Sensor Networks: Results That May Surprise," in *Globecom Workshops, 2007 IEEE* , Washington, DC , 2007, p. 1.

[21] Maria Striki and Tony McAuley, "Using Novel Distributed Heuristics on Hexagonal Connected Dominating Sets to Model Routing Dissemination," in *Communication Theory, Reliability, and Quality of Service (CTRQ), 2010 Third International Conference on*, Athens, Greece , 2010, p. 22.

[22] Waleed Alsalih, Selim Akl, and Hossam Hassanein, "Placement of multiple mobile base stations in wireless sensor networks," in *Signal Processing and Information Technology, 2007 IEEE International Symposium on* , Giza, 2007, p. 229.

[23] David B. Pistilli, Robert J. Elder Jr., and Ronald E. Keys, "United States Air Force Network Operations Functional Concept," 2006.

[24] Matthew Capt. USAF Compton, Improving the Quality of Service and Security of Military Networks with a Network Tasking Order Process, 2010.

[25] Dorit S. Hochbaum, Ed., *Approximation Algorithms for NP-Hard Problems*. Boston, USA: PWS Publishing Company, 1997.

[26] Eric W. Frew and Timothy X. Brown, "Mobile Agent Relay Networks for Small Unmanned Aircraft Systems," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2008-2027, December 2008.

[27] Anthony L. Larweck, "Using a Multiobjective Approach to balance Mission and Network Goals Within a Delay Toleratn Network Topology," Air Force Institute of Technology, Wright-Patterson AFB, OH, Master's Thesis AFIT/GE/ENG/09-26, 2009.

[28] Hyon H. Kwak, "Toward a Mobile Agent Relay Network," Air Force Institute of Technology, Wright-Patterson AFB, OH, Master's Thesis AFIT/GCS/ENG/10-04, 2010.

[29] Theodore S. Rappaport, *Wireless Communication: Principles and Practice*, Camille Trentacoste, Ed. Upper Saddle River, NJ, USA: Prentice Hall, 1996.

[30] Harris RF Communications, "AN/PRC-117G(V)1(C) Applications Handbook," Rochester, NY, User Handbook 2009.

[31] Jon Kleinberg and Eva Tardos, *Algorithm Design*. Boston: Addison Wesley, 2006.

[32] James Haught, "Adaptive Quality of Service Engine with Dynamic Queue Control," Air Force Institute of Technology, Wright-Patterson AFB, OH, Master's Thesis AFIT/GCS/ENG/11-03, 2011.

[33] Susan J. Milton and Jesse C. Arnold, *Introduction to Probability and Statistics*, 4th ed. Boston, United States: McGraw Hill, 2003.

[34] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 2nd ed. Boston: Massachusetts Institute of Technology Press, 2001.

[35] Azzedine Boukerche, Anis Zarrad, and Araujo Araujo, "A Cross-Layer Approach-Based Gnutella for Collaborative Virtual Environments over Mobile Ad Hoc Networks," *Parallel and Distributed Systems, IEEE Transactions on*, pp. 911-924, 2010.

[36] Brian Drabble, Tim Black, Chris Kinzig, and Gary Whitted, "Ontology Based Dependency Analysis: Understanding the Impacts of Decisions in a Collaborative Environment," in *Collaborative Technologies and Systems, 2009. CTS '09. International Symposium on* , Baltimore, MD, 2009, pp. 10-17.

[37] Janth English and Roy George, "Database support for collaborative battle planning," in *Automation Congress, 2002 Proceedings of the 5th Biannual World*, vol. 14, 2002, pp. 429-434.

[38] Jeffrey Heissler, Dave Kaplan, Stan Manoski, and Jenny Wu, "Collaborative Planning Over Low Bandwidth Tactical Networks," in *Military Communications Conference, 2004. MILCOM 2004. IEEE*, vol. 2, 2004, pp. 630-634.

[39] Panos Louvieris, Catherine Collins, and Natasha Mashanovich, "Investigating the Use and Effectiveness of Virtual Collaboration Desks for Collaborative Military Planning," in *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on* , Big Island, HI , 2009, pp. 1-10.

[40] Joseph B. Lyons, Stephanie D. Swindler, and Julie A. White, "Network Centric Warfare: Organizational Collaboration as a Key Enabler," in *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on* , Irvine, CA, 2008, pp. 367 - 374.

[41] Ali Bahrani, Jun Yuan, David Emele, Daniele Masato, and Timothy J. Norman, "Collaborative and context-aware planning," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, San Diego, CA, 2008, pp. 1-7.

[42] Dirk R. Klose, Israel Mayk, Andrew Chan, Mike Mai, and Hobbie Negaran, "Cognitive interoperability and decision making in network-centric coalition force environments," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*, Atlantic City, NJ , 2005, pp. 2406-2412.

[43] Harvey Reed and Fred Stein, "Net-Centric Conversations: The Unit of Work for Network Centric Warfare and Network Centric Operations," in *Military*

Communications Conference, 2006. MILCOM 2006. IEEE , Washington, DC , 2006, pp. 1-8.

[44] Kevin Dela Rosa and Jeffrey Ellen, "Text Classification Methodologies Applied to Micro-text in Military Chat," in *Machine Learning and Applications, 2009. ICMLA '09. International Conference on*, Miami Beach, FL , 2009, pp. 710-714.

[45] LorRaine T. Duffy, "DoD Collaboration and Chat Systems: Current Status and Way Ahead," in *Semantic Computing, 2008 IEEE International Conference on*, Santa Clara, CA, 2008, pp. 573-576.

[46] Roger Trafton and Steven V. Pizzi, "The Joint Airborne Network Services Suite," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, Washington, DC, 2006, pp. 1-5.

[47] John Brand and George Hartwig, "Ad Hoc Network Management With C2 Data Models ," in *SoutheastCon 2001. Proceedings. IEEE*, Clemson, SC, 2001, pp. 100-104.

[48] Roy Ladner, Elizabeth Warner, Fred Petry, , Uday Katikaneni, and Kevin Shaw, "Web Services: Evolving Techniques in Net-Centric Operations," in *OCEANS, 2005. Proceedings of MTS/IEEE*, 2005, pp. 441-448.

[49] Stephen E. Cross, Edward Walker, Harry Forsdick, Ted C. Kral, and Bolt Beranek, "Collaboration technology for crisis management," in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1994. Proceedings., Third Workshop on*, Morgantown, WV, 1994, pp. 48-52.

[50] William K. McQuay, "Distributed Collaborative Environments for Systems Engineering," *Aerospace and Electronic Systems Magazine, IEEE* , pp. 7-12, 2005.

[51] Zhu Haibin and MengChu Zhou, "Role-based collaboration and its kernel mechanisms," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, pp. 578-589, 2006.

[52] James Marsh, Mashhuda Glencross, Steve Pettifer, and Roger Hubbold, "A network architecture supporting consistent rich behavior in collaborative interactive applications," *Visualization and Computer Graphics, IEEE Transactions on* , pp. 405-416, 2006.

[53] Pieter Jorissen, Maarten Wijnants, and Wim Lamotte, "Dynamic Interactions in Physically Realistic Collaborative Virtual Environments," *Visualization and Computer Graphics, IEEE Transactions on*, pp. 649-660, 2005.

[54] Helmut Hoyer, Andreas Jochheim, Christof Röhrig, and Andreas Bischoff, "A multiuser virtual-reality environment for a tele-operated laboratory," *Education, IEEE Transactions on*, pp. 121-126, 2004.

[55] Jason Leigh and Andrew E. Johnson, "Supporting transcontinental collaborative work in persistent virtual environments," *Computer Graphics and Applications, IEEE*, pp. 47-51, 1996.

[56] Latha Kant et al., "Network science based approaches to design and analyze MANETs for military applications," *Communications Magazine, IEEE*, pp. 55-61, November 2008.

[57] Jack L. Burbank, Philip F. Chimento, Brian K. Haberman, and William T. Kasch, "Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 39-45, November 2006.

[58] Elizabeth M. Royer and Chai-Keong Toh, "A Review of current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications*, pp. 46-55, April 1999.

[59] Tracy. Camp, Jeff Boleng, and Vanessa Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, p. 483–502, August 2002.

[60] Christian Bettstetter and Giovanni Resta, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks," *Mobile Computing, IEEE Transactions on*, pp. 257-269, July 2003.

[61] Joseph E. Martz, Joseph E. Jr. O'Donnell, Wedni B. Carpenter, and Stephen J. Miller, "Multi-Service Tactics, Techniques, and Procedures for Internet Relay Chat for Command And Control," 2009.

REPORT DOCUMENTATION PAGE		*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 24-03-2011	2. REPORT TYPE Master's Thesis	3. DATES COVERED *(From – To)* Sep 2009 – Mar 2011
4. TITLE AND SUBTITLE Routing UAVs to Co-Optimize Mission Effectiveness and Network Performance with Dynamic Programming		**5a. CONTRACT NUMBER**
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Spenser D. Lee, 2nd Lt, USAF		**5d. PROJECT NUMBER** 11G256
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCS/ENG/11-04
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Bruce Preiss Chair, AFRL Modeling and Simulation Working Group (RYAA) Air Force Research Laboratory (AFRL) 2241 Avionics Circle, Area B Bldg 620, Wright-Patterson AFB, OH, 45433 (937) 528-8308 Bruce.Preiss@wpafb.af.mil	10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RYAA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In support of the Air Force Research Laboratory's (AFRL) vision of the layered sensing operations center, command and control intelligence surveillance and reconnaissance (C2ISR) more focus must be placed on architectures that support information systems, rather than just the information systems themselves. By extending the role of UAVs beyond simply intelligence, surveillance, and reconnaissance (ISR) operations and into a dual-role with networking operations we can better utilize our information assets. To achieve the goal of dual-role UAVs, a concrete approach to planning must be taken. This research defines a mathematical model and a non-trivial deterministic algorithmic approach to determining UAV placement to support ad-hoc network capability, while maintaining the valuable service of surveillance activities.

15. SUBJECT TERMS

co-optimization, UAV, ISR, MANET, ad-hoc routing, topology control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dr. Kenneth M. Hopkinson, ENG
REPORT U	**ABSTRACT** U	**c. THIS PAGE** U	UU	117	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 x4579; Kenneth.Hopkinson@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18