3-17-2009

Using a Multiobjective Approach to Balance Mission and Network Goals within a Delay Tolerant Network Topology

Anthony L. Larweck

USING A MULTIOBJECTIVE APPROACH TO BALANCE MISSION AND NETWORK GOALS WITHIN A DELAY TOLERANT NETWORK TOPOLOGY

THESIS

Anthony L. Larweck, Capt, USAF

AFIT/GE/ENG/09-26

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/09-26

USING A MULTIOBJECTIVE APPROACH TO BALANCE MISSION AND NETWORK GOALS WITHIN A DELAY TOLERANT NETWORK TOPOLOGY

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 in Electrical Engineering

Anthony L. Larweck, B.S.E.E.

Capt, USAF

March 2009

AFIT/GE/ENG/09-26

USING A MULTIOBJECTIVE APPROACH TO BALANCE MISSION AND NETWORK GOALS WITHIN A DELAY TOLERANT NETWORK TOPOLOGY

Anthony L. Larweck, B.S.E.E.

Captain, USAF

Approved:

Capt. Ryan W. Thomas, PhD (Chairman)	17MAR09 Date
Lt. Col Stuart H. Kurkowski, PhD (Member)	17Mar09 Date
Dr. Kenneth M. Hopkinson (Member)	17MAR09 Date

Abstract

Current Air Force operations are dictated by the mission, with planning and requirements for these missions detailed in an Air Tasking Order (ATO). An analogous document exists for the communications (link) requirements in the Communications Tasking Order (CTO). Unfortunately, the CTO has very little networking focus, meaning the network requirements and plans are not adequately specified or tied into the mission. This means network performance often suffers. One approach currently being pursued to address this issue is the Network Tasking Order (NTO), which identifies these network requirements and plans, allowing a commander to manage his battlespace in a manner that incorporates mission and network requirements. Unfortunately, while the ATO and NTO provide important data for the commander, using it for management requires manual intervention by network and mission operators, causing inefficiencies due to delayed response or inconsistent problem resolution. Particularly for autonomous weapon systems such as unmanned aerial vehicles (UAVs), an improved approach would be to automate this management. One way to accomplish this would be to incorporate the NTO and ATO data into cognitive network framework. Since cognitive networks have already been proposed as a way of intelligently adapting to end-to-end *network* goals, extending the cognitive process to incorporate global *mission* goals is a natural extension. This paper describes a simple cognitive network process designed to solve the multi-objective optimization problem of balancing *both* network and mission goals.

This process consists of two components: a multi-move look-ahead component, in which the future outcome of decisions are estimated, and a subsumption decision making

architecture (from the field of behavior based robotic control), in which these decision-outcome pairs are selected so that they co-optimize the dual goals. To test this process, it was applied to a sample Air Force mission scenario consisting of an UAV surveillance mission within a delay tolerant network (DTN) topology. This scenario used a team of UAVs (operating as a team but each running the cognitive process independently) to balance the mission goal of maintaining maximum overall UAV time-on-target and the network goal of minimizing the packet end-to-end delays experienced in the DTN. In this scenario, the cognitive process could control three possible parameters of the UAV operation: the current orbit of the UAV, whether to hold or forward packets, and, if forwarding packets, what UAV to forward the packets to. The future outcomes of each of these decisions was evaluated by the subsumption process to determine which set of actions best optimized both the mission and network goals in the long term.

The performance of the cognitive process under this scenario was evaluated thru simulation by comparing it against a baseline, non-cognitive DTN approach. Two scenarios were investigated: optimizing just the network goal and balancing both the network and mission goals. The results indicated that the pseudo-cognitive approach improved the mission goal of increasing the average percent loiter time by approximately 6 and 6.75 percent and the network goal of end-to-end delay approximately 48.6 and 52 percent over the baseline DTN and look-ahead approaches for the associated workloads of 1 and 2 percent of total image opportunities.

Acknowledgments

I would like to thank my thesis advisor, Capt Ryan Thomas, for providing the proper level of guidance to keep me on track, while allowing me the freedom to grow technically and fully pursue my ideas. Lastly, I'd like to thank LtCol Stuart Kurkowski and Dr. Ken Hopkinson for their support as well.

Anthony L. Larweck

Table of Contents

Page

List of Figures

List of Tables

USING A MULTIOBJECTIVE APPROACH TO BALANCE MISSION AND NETWORK GOALS WITHIN A DELAY TOLERANT NETWORK TOPOLOGY

I. Introduction

The mission of the United States Air Force is to "*fly, fight* and *win* ... in air, space and cyberspace." To achieve this domain supremacy, the Air Tasking Order (ATO) and Communications Tasking Order (CTO) are used to specify the requirements necessary to support these vital missions. The ATO and CTO identify mission plans, involved units, desired aircraft and forces, communication requirements, as well as the details regarding the particular targets of interest [1].

To elaborate further, the ATO highlights specific aspects related to the mission (such as flight paths, call signs, targets, controlling agencies), as well as general mission instructions. In contrast, the CTO addresses the communication aspects related to the allocation of the useable electromagnetic spectrum to the various units in the Area of Responsibility (AOR), prescribes networks and provides contact information for each network manager. The CTO also lists the various data links, describes their capacity, itemizes maintenance actions, and provides commanders with up-to-the-minute availability and downtime visibility [1].

With the evolution of the cyberspace domain and net-centric warfare, computer networks have come to the forefront in the communications realm. As a result, the network architecture (links and components) have been included within the CTO, but the actual desired performance objectives have been neglected. Unfortunately, neither the ATO nor CTO adequately describe the requirements for the network performance. As a result, the network succumbs to the priority of the mission goals dictated in the ATO and the overall network performance suffers.

Recent efforts have initiated the development of a network tasking order (NTO) [1, 2] to specify these desired performance objectives in an attempt to directly control these attributes and provide the commander a way to manually fine tune the computer network to better align its objectives with the mission objectives. However, this would still require continuous oversight and intervention by the commander and therefore would not produce an optimal approach. Creating an autonomous dynamic ability to adjust the mission and network parameters would greatly benefit the Air Force and will be the focus of this endeavor.

1.1 Background

The main premise of the NTO is to identify the communication assets/nodes and capabilities within the battlespace in such a way as to create an interconnected picture of the network topology. This information could then be used to determine and/or exploit potential routing opportunities to improve network performance. This aspect could prove to be extremely beneficial within Air Force missions since routes are typically predictable as they are based upon a priori knowledge [2].

Of particular interest are small unmanned aerial vehicle (UAV) missions conducted in urban or sparsely connected environments. Current operations abroad (i.e., Iraq, Afghanistan) have proven that these assets are extremely useful since they can provide soldiers the ability to gain situational awareness without endangering lives unnecessarily. Typical operations are over-the-hill reconnaissance and convoy-following tasks. Essentially these units are very portable (e.g., can weigh pounds), launched by hand, and can be controlled via a small ground station and receive images via a laptop or personnel electronic device [3]. This can prove useful in the military context of dynamic environments with little or no infrastructure.

Additionally, UAVs can be utilized to extend communication, either through multihop ad-hoc networks or via a delay tolerant network (DTN) construct. A DTN is one in which the link for the next hop or relay node is disrupted or delayed for a finite period of time. This topology configuration does not conform to typical routing protocols (because there is no end-to-end path). This deficiency is typically overcome by either moving a node into communication range or by using a data ferry to carry information within communication range of the intended destination. However, since this is an eventual delivery approach for data transfer, long delays are typically incurred. This situation could be extremely detrimental in military operations where information is typically time sensitive and failure to respond can be catastrophic.

1.2 Research Problem

In current Air Force operations, mission goals dominate network goals thus causing network inefficiencies. This is due partly to the exclusion of network parameters within the mission planning process. As was described earlier in Section 1, this is because neither the ATO nor CTO adequately capture these parameters. Development of an NTO will aid in closing this gap but will not eliminate the need for administrator or commander intervention for direct resolution.

An approach is needed to address these deficiencies dynamically within the context of a common framework in an effort to achieve optimal performance by balancing mission and network objectives without intervention. Research into an effective scheme or methodology is needed to accomplish this requirement.

1.3 Approach

This thesis presents a novel approach to address the aforementioned suboptimal network performance issues. This methodology builds upon concepts of the NTO, behavior based robotics, multi-objective optimization, and cognitive networks.

This work was based upon an Air Force UAV surveillance mission using a DTN topology. A pseudo-cognitive approach was demonstrated by using a team of UAVs that autonomously attempted to balance the mission and network goals by employing a subsumption multi-objective optimization (MOO) architecture within the Unified Behavioral Framework (UBF) [4] controller construct. The mission goal in this case was to maximize the average percent loiter time over targets of interest whereas the network goal was to minimize the end-to-end delay of the network traffic.

This work builds upon the applicability of the NTO, implementation of cognitive networks, and use of multi-objective optimization within DTN environments. The objective of this thesis is to incorporate aspects of an NTO within a pseudo-cognitive approach in an attempt to balance the mission and network goals to achieve an overall improved network performance. For the purposes of this work, pseudo-cognitive implies that all cognitive aspects are incorporated with the exclusion of the learning aspect.

Since UAVs are not readily available for this research, this problem was assumed to be comparable to a ground based, mobile robotics problem and was modeled as such. This assumption was made because mobility patterns dominate the overall network performance and robotic movement can be made similar to UAV orbits (with the exception of fewer degrees of freedom). As such, approaches from robotics were used to form the basis of the framework and methodology used in creating the solution. Behavior based robot control was used as a means to implement the overall framework.

Since the Unified Behavioral Framework (UBF) was an example of this, it was used as a basis to form the overall framework. This framework was then incorporated with a subsumption multi-objective approach in an effort to capitalize on its simplicity, ease of implementation, and reactive nature. This approach has been used in the past for behavior-based robotic control methods for control of autonomous robot agents. This was used in an attempt to overcome the delay shortcomings of the DTN. This approach will attempt this while also attempting to maximize surveillance time over target.

The ideas presented in this thesis were tested using a discrete event MATLAB simulation that was created to implement, test and perform validation. The validation aspect was accomplished using the MATLAB debugger and overall structure construct to ensure movements were synchronized correctly and the results were consistent.

1.4 Scope

The scope of this thesis examined the system performance impact as a result of the inclusion of ATO, CTO, and NTO requirements within a pseudo-cognitive approach used to manage data routing and orbit selection in a DTN topology.

The design of the MATLAB discrete event simulation was a full factorial design with three DTN configurations which are tested under two workloads (5 and 10 images/min). The configurations were a baseline DTN, multi-step look-ahead DTN (pseudo-cognitive), and multi-step look-ahead with MOO. The approaches were distinguished by increasing levels of cognition. The metrics observed consisted of the average percent loiter time, number of images transmitted and received, and end-to-end delay. There were 25 iterations per experiment to achieve a 95 percent confidence interval. This resulted in 150 simulated trials.

II. Background and Literature Review

2.1 Introduction

This chapter provides basic definitions and an overview of research regarding the associated concepts related to the major aspects of this thesis. The areas of NTO, Small Scale UAVs, Mobile Ad Hoc Networks (MANETs), delay tolerant networks (DTNs), cognitive networks, multi-objective optimization techniques, and robot control principles will be discussed.

2.2 Network Tasking Order

With the advancements in technology and increased emphasis on computer network availability and capacity, net-centric operations and pursuit of the Global Information Grid (GIG) have come to the forefront in modern warfare. Department of Defense Directive 8000.01 [5] and Air Force Joint Vision 2020 [6] highlight the tenets of net-centric warfare (NCW) and layout the roadmap for military operations within this Information Age. Of primary importance is the fact that "interconnectedness" will become a necessary global requirement and the need for information will continue to dictate all aspects of the mission. Terms like shared situational awareness, information advantage, and information superiority have become mainstays within current policies.

As a result of this emphasis, network availability has become a critical part of daily operations within the military. This has in turn led to the realization that cyberspace is a distinct domain that must be incorporated within the mission planning process. If neglected, inefficiencies and vulnerabilities to attack leave the military ill prepared to conduct the task at hand.

As alluded to in the introduction, the NTO concept is a way to capture and document these critical requirements in such a way that this additional information can be used to enhance the mission planning process. Along with the ATO and CTO, the complete picture is then conveyed to the mission planners and combatant commanders. This allows network aspects to be considered which normally would not. It makes network performance improvement possible by providing a means to consider trade offs which may currently be neglected. For example, if the mission is the only consideration during planning (with all resources are allocated to mission accomplishment) then the mission objective by defacto will dominate the network objectives and network performance can suffer.

This concept is not entirely new but it has received more attention recently. Stookey introduces the idea of a notional battlespace in which communication assets that could be used to route dynamic traffic are clearly identified. Also, the expected position and movement of these assets are captured, identifying potential connectedness and thereby allowing predictions of where links can or will be established. This can greatly enhance routing protocols and allow opportunistic connections that otherwise couldn't be identified. Lastly, he emphasized how understanding the overall communication needs could provide better resolution and more efficient use of theater bandwidth [1].

Pecarina interpreted the purpose of an NTO as providing the commander the ability to allocate bandwidth and link availability by strategically positioning communication resources within a theater of operations by assigning weights to differing priorities. This would in turn provide the commander the ability to dominate cyberspace and hold information superiority over the enemy [7].

Compton presented a slightly different approach, stating that the NTO should be structured similar format to the ATO and include each mission type (i.e. type of network traffic expected), node configuration and capabilities (i.e. a relay, router, etc…), and the priority of traffic flows. Also, he highlighted that the additional information could identify single points of failure, gaps in connectivity, or possible bottlenecks [2].

2.3 Small Scale UAVs

In 2007, the Department of Defense released a report titled "The United States Department of Defense Unmanned Systems Roadmap for 2007-2032 [8]". This extensive report described in detail the DoDs desire to aggressively pursue the development and deployment of unmanned systems. Within the plan all aspects of unmanned systems were described with emphasis on creating a "sophisticated unmanned force to entail vehicles of the three domains of air, land, and sea [8]. The report went on to highlight the top DoD priorities within specific battlefield applications for UAVs within military. These applications comprised of the areas of reconnaissance and surveillance, target identification/designation, counter mine warfare, and chemical, biological, radiological, nuclear, explosive (CBRNE) reconnaissance [8]. Specifically, within the reconnaissance/surveillance and target identification/designation areas, the main focus was regarding the abilities to maintain covertness, positive identification of enemy targets, and reducing latency and precision of GPS guided weapons.

Emerging technologies and development challenges were also highlighted within the report. Autonomy and cognition were among the top concerns. Autonomy referred to a UAV possessing the ability to make autonomous decisions for either collision avoidance or collaborative/cooperative communication between multiple vehicles for object sensing and surveillance. Cognition is the capability of an unmanned system to extend human perception and action capabilities, with perception being a way to enhance understanding, reasoning, and decision making in a mission environment and action instilled within computer algorithms used to systematically solve problems, in a thought invoking decision making process. These are a few of the key concepts that are highlighted within this thesis and will be explored in more detail later in this chapter.

Of particular interest within this document was the area regarding small scale UAVs. Small scale UAVs are defined as having a gross takeoff weight of less than 55 pounds. The ceiling altitude can reach up to 15,000 ft and the operating range is up to 40 nautical miles. This is the type of UAV that the scenario will be depicting.

JUAS Categories	Current System Attributes							
	Operational Altitude (ft)	Typical Payload	Launch Method	Weight (lbs)	Airspeed (kts)	Endurance (hrs)	Radius (nm)	Current Systems (Projected by 2014)
T1 - Tactical 1 Special Operations Forces (SOF) Team Small Unit Company & below	≤ 1,000	Primarily EO/IR or Comm Relay	Hand launched	≤ 20	≤ 60	< 4	< 10	Hornet, BATCAM, Raven, Dragon Eye, FPASS, Pointer, Wasp, BUSTER (rail-launched), MAV
T2 - Tactical 2 Battalion/Brigade Regiment SOF Group/Flight	≤ 5,000		Mobile launched	20 - 450	≤ 100	< 24	< 100	Neptune, Tern, Mako, **OAV-II,** Shadow, Silver Fox, ScanEagle, Aerosonde

Figure 1. Types of Small Scale UAVs [8]

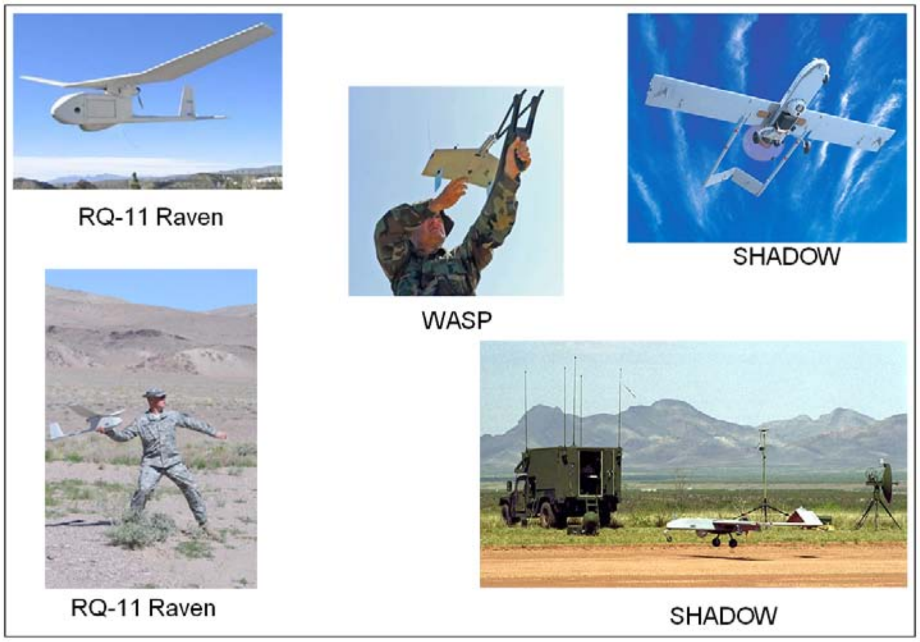

Figure 2. Small Scale UAVs [9-11]

In addition to military application, UAVs are used throughout government and civilian sectors. Applications range from law enforcement, to wildlife management, and environmental studies. However, since the operating environment is not in as strict control as the military airspace there are many regulatory issues (FAA issues) relating to controlled airspace and how to best manage these assets within the airspace, namely safety of flight and air traffic control issues.

Frew and Brown [12] went on to highlight the aforementioned regulatory issues and also address issues with networking small scale UAV systems. In particular they emphasized how operational requirements dictate network requirements. In the Air Force, this is analogous to mission requirements leading to networking requirements. They went on to state how these demands can then in turn directly affect network connectivity data delivery and service discovery [12]. Additionally, they state that a delay tolerant mobile ad hoc network architecture offers the best option in terms of flexibility, reliability, robustness, and performance compared to other network possibilities. This type of network configuration provides the opportunity to exploit controlled mobility to improve performance when the network becomes fractured [12].

Many universities are currently conducting research regarding UAV systems [13-15]. Their works focus mainly on control system implementations incorporating autonomous actions, but none appear to incorporate a cognitive aspect used to balance multiple objectives as is the focus of this work. There are however, some examples that have contributed to forming the ideas within this thesis.

Of interest, the University of Colorado created the Ad-hoc UAV Ground Network (AUGNet) in an effort to study the performance of airborne mobile ad-hoc networks. This experimental platform was comprised of both UAVs and ground nodes. Communication paths were created from UAV to UAV, UAV to ground node, or UAV to UAV via a ground node used as a relay. The protocol used was the Dynamic Source Routing (DSR) ad hoc protocol. This is an on demand routing protocol which means it creates a route when there is a packet to send. As a result of experimentation, it was determined that the significant factors affecting the network performance were the path link (number of hops), quality of the link, whether the nodes were fixed or mobile, and whether a UAV was used for routing information. Tests were conducted to measure the network performance (throughput, connectivity, and congestion). Their results identified that adding an airborne mobile node (i.e. the UAV) within the network doubled the communication range of the baseline ground to ground nodes and also increased the throughput (with greater than two hops between nodes) but with a higher variance (possibly due to node maneuvering) [12].

This work formed the basis for Brown and Frew's future effort on the Heterogeneous Unmanned Aircraft System (HUAS) [16]. The HUAS was created to study airborne communication networks and specifically multivehicle control. Within their work they focused on the design of an intelligent flight management system for UAVs. This system was designed to provide operator control of system parameters while allowing the vehicle to make autonomous mission level decisions based on network metrics. It essentially augmented the AUGNet framework with their advanced communication, command, and control system creating an intelligent node.

This intelligent node combined network metrics, vehicle status, and mission parameters to perform data-centric tasks while remaining within mission parameters specified by an operator. Messages were disseminated via a broadcast to all nodes to avoid the complexity associated with addressing requirements. The temperature and ping packet parameter levels were established via the 802.11 ad-hoc link by the operator and when a condition was encountered (meaning a parameter was met), the vehicle autonomously made a decision to return to base. This construct demonstrated that high level decisions could be made (i.e. using a combination of parameters) based upon network metrics (ping packet parameter) and specified operating conditions [16]. However, this work is distinctly different from this thesis in two ways. First, it doesn't attempt to maintain a threshold level (maintain a constant level) it just meets a designated level which triggers a conditioned response. Secondly, it doesn't pursue competing objectives which requires a multi-objective process.

2.4 Mobile Ad Hoc Networks

In wireless networks, communication is conducted via RF links between wireless hosts typically tied into the overall network infrastructure at the network edge (but not always) via a wireless access point (base station) [17]. These wireless hosts can be laptops, PDAs, phones, or even desktop computers and must be within range of an access point to receive a signal and establish/maintain a connection. Wireless communication poses distinct challenges different than conventional wired computer networks in the form of path loss due to decreased signal strength, interference from other sources, and multi-path propagation which occurs from additional signal information as a result from a reflection from the ground or other objects thus causing the received signal to differ from the one sent [17].

An Ad-Hoc Network is basically a wireless network that can be quickly established as an autonomous network operating either in isolation or as a "stub network" that connects to a fixed network [17]. This type of wireless network lacks infrastructure (e.g., there is no need for a base station) and therefore has to have each host provide the necessary services required for routing, address assignment, DNS-like name translation and more [17]. This creates a way to quickly establish a wireless network by rapidly configuring links without the need for infrastructure planning, reducing time and cost.

Mobile Ad Hoc networks (MANETs) add movement to the wireless network topology which results in additional complexities, however it appeals to a broad new audience of applications. MANETs are of particular interest for military and disaster response applications since they address the concept of battlefield survivability. This concept pertains to a computer network's ability to avoid single points of failure (decentralized distribution with redundancy), ability to operate independent of existing communications infrastructure, rapidly deployable, self organizing, and uses multi-hop packet routing for communication to users who are not within line-of-site. Overall this type of network provides a mobile, deployable, wireless, multi-hop network [17].

Therefore, MANETs are characterized by their ability to multi-hop for communication, possess a dynamic topology, are bandwidth constrained, contain variable capacity links based on each unique node that enters the network, are energy-constrained, have limited physical security, scalability, self-organizing capabilities, and the ability to communicate and move at the same time. Since these networks are distributed (with no centralized control point), they require completely distributed algorithms that have each node potentially a router.

In general, in a computer network routing must be done efficiently and therefore the optimal route must be determined. There are two standard algorithms used to determine this "best" path—distance vector and link state. The distance vector algorithm uses the Bellman Ford algorithm to determine the shortest distance paths based on the information received from its neighboring nodes. This method requires each node in the network to build and maintain a routing table containing the distance between itself and all possible destination nodes. The link state algorithm uses the Dijkstra algorithm to calculate the cost to take a particular path. This process entails each node (router) construct a link state packet consisting of the names of and cost for each of its neighbors, disperse the packet by flooding to all nodes in the network, then use the received information from all nodes (global information) to build and maintain a routing table for the entire network [18].

In general, there are routing protocols to specify how routers communicate with each other and how they use the above mentioned algorithms (or slight variations) to determine the best path to reach the intended destination. In Ad Hoc wireless networks, there are several protocols that have been developed and these are shown in Figure 3. However, the focus of this effort is dependent on a routing protocol based on using a routing information update mechanism. The three approaches regarding this are the proactive, reactive, or hybrid [18].

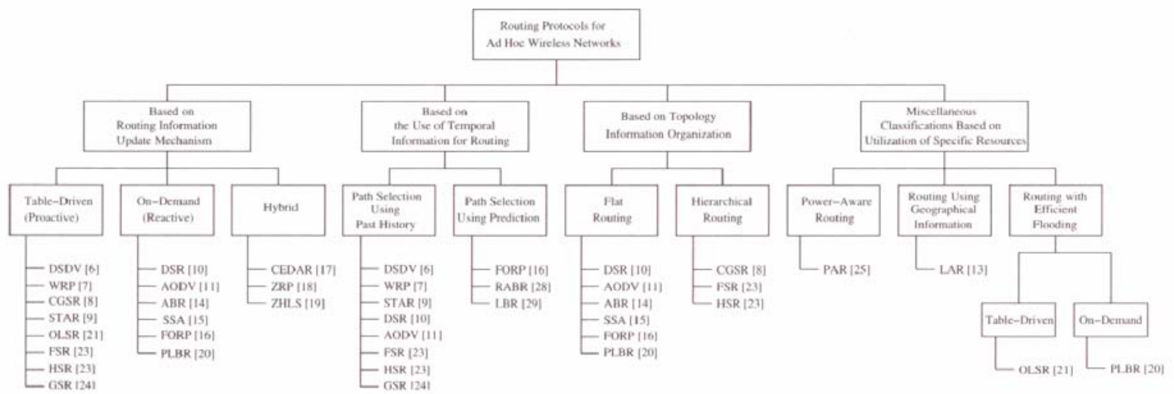

Figure 3. Routing Protocols for Ad Hoc Networks [19]

Proactive protocols establish the route paths in advance and then maintain and update routing tables (Table-driven) as the network dynamically changes. Reactive protocols establish routes as needed and only maintain the route while needed. These routes must be discovered since they are not known in advance. Hybrid protocols are a combination of the proactive and reactive protocols. They are proactive within a geographic area and reactive if the packet must travel outside of the defined area [18]. Therefore, use of a protocol is dependent on the intended application since there are tradeoffs that must be considered in each case. For example, if the delay required for route discovery is not acceptable, a proactive protocol should be used whereas if one wanted to only maintain the information regarding the active routes a reactive protocol should be used [18].

2.5 Delay Tolerant Network

In normal computer network operations, connectivity is established between nodes within and possibly across communication regions. This can be accomplished by either wired or wireless means to create the interconnected grid required for communications and data delivery. Within this architecture, internet protocols have been used to accomplish networking and data transfer tasks, particularly Transmission Communication Protocol (TCP). However, TCP requires an end-to-end connection to exist long enough to verify the connection (send an acknowledgement and receive confirmation), send the data, then receive a confirmation that the data was sent. In order to accomplish this task the connection must exist for a defined set of time. This time has been called “time-to-live” [17,19] and is established in the packet header of the packet to be sent. Therefore, if there are disruptions or the delay exceeds the packet “time-to-live”, the protocol will not work. As a result, a new architecture has been designed to handle these situations when they exist, a delay tolerant network (DTN) framework [19].

A DTN is a type of computer network that consists of geographically separated communication regions that are characterized by long delays and intermittent disruptions. The framework was first designed for an interplanetary internet communication system to conduct deep space exploration and was recently captured within RFC 4838 by the Internet Engineering Task Force (IETF) [20,23]. This foundation has now led to many practical applications that were not possible before.

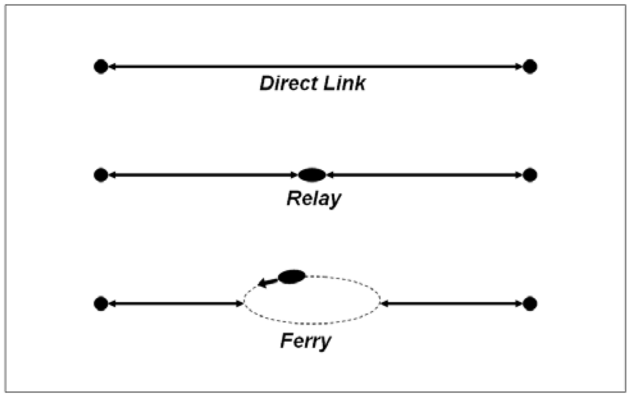

Figure 4. Comparison between different communication link connections

Here is a simplified example to highlight the main points. Imagine living in a rural area and have a written a letter that needs to be delivered to a friend that lives down the road (geographically separated regions). A decision was made to send it to them by giving it to a neighbor along the path within a limited distance (transmission range) and then they give to the next neighbor and so on until the letter reaches the friend (store-and-forward approach). However, it is undesirable to wait too long for the letter to get to them and as such do not want confirmation at each transfer to the next person. Instead the person who is transferring the message is held accountable to ensure it was sent. Any number of disruptions could occur along the route, preventing the transfer of the letter between subsequent neighbors.

A particularly long delay could result if a next neighbor in the path is never home causing the sender to hold on to the letter for an extended period of time. One possible solution to this problem would be to have the waiting neighbor act as a data “ferry” driving past this neighbor to the next in order to bypass a missing neighbor, delivering the letter to the next neighbor in line. The data ferry would be limited in distance it could travel and who it could contact along the way. This can be further complicated if each neighbor is only home for a scheduled timeframe and therefore causing the delivery to be hit or miss. In this scenario, neighbor schedules would need to be known in advance in order to correctly sequence the transfer.

This analogy capture a scenario in which an end-to-end path for data delivery may or may not exist at any given time. This is a typical DTN configuration and illustrates why TCP would fail given this situation – without an static end-to-end path, ACK messages would never be delivered. To overcome this limitation, data delivery is reliant upon a data ferry for data delivery. As in the analogy, a data ferry is used to extend the transmission range by physically moving closer to the destination. However, since the intermediate and final destination node locations vary in time and variations in propagation delays and end-to-end paths, routing can be very difficult. Intermediary nodes are used for storage until the next hop or destination is within range for transmission. This type of routing is based on an eventual delivery approach in which connections can be in stored (queued) which in turn can result in long delays [21]. This problem is the focus of much research in the DTN field.

2.5.1 DTN vs MANET

As was described above, DTNs have unique aspects that are both similar and dissimilar from MANETs. As is similar to a MANET, in a DTN network normal computer networking conventions do not hold. Node connections are time varying and the network topology changes dynamically. However, within MANETs routing protocols aim at establishing end-to-end paths between communicating nodes and thus support the end-to-end semantics of existing transports and applications [22]. In contrast, DTN schemes imply asynchronous store and forward communication.

2.5.2 DTN Architecture

RFCs 4838 [24] and 5050 [25] detail the DTN architecture and bundle protocol specification respectively. These RFCs were created by the Internet Research Task Force (IRTF) [26] in an effort to standardize protocol specifications for development but not to act as an overall internet standard.

The DTN architecture was originally designed for interplanetary communications in which long delays are encountered. This was in an effort to provide internet type services for deep space exploration. This framework addresses concerns with occasionally connected or disrupted network connections (scheduled or nonscheduled).

A “bundle layer” [24] is required to run as a layer above the transport layer in which information is passed to endpoints or nodes. This layer is used for persistent storage that is needed for the store and forward approach that is employed. This is a critical requirement since long queuing is needed to store the message until a contact becomes available. These contacts may be persistent, on-demand, intermittently opportunistic, intermittently scheduled, or intermittently predictive. Persistent is always available and on-demand is as required remaining persistent until terminated. The intermittent connections are more typical of a DTN in which the connection is only available at intermittent times.

Custody transfers are used as a means for reliable data transfers. This is done to ensure a message has been successfully transferred to the next hop in the routing path. The node transferring the message performs a custody transfer in which the node receiving the message now accepts responsibility to ensure the data reaches its desired destination. The messages are encapsulated in “bundles” and use endpoint identifiers to identify the source and destination.

2.5.3 Bundle Protocol

The bundle protocol was designed for implementation within the DTN architecture to act as an overlay network to run on top of the current Internet Protocol (IP) to account for deficiencies with connectivity. This protocol is considered an overlay network store-and-forward protocol [23]. The packets used in normal IP are formed into bundles with the necessary control information to bridge the gap between the application and TCP layers of the OSI model.

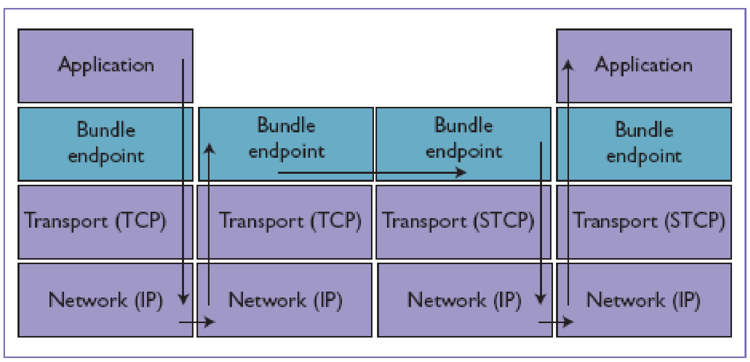

Figure 5. Overlay Network Approach: Bundle Protocol [19]

The bundles possess the required packet format to support end-to-end messaging within the bundle layer (instead of the transport layer) to account for the deficiencies of the TCP protocol. Several research groups have developed simulations of the DTN implementation by incorporating the bundle protocol. However, since this work only needs to capture the impact mobility of nodes on end-to-end delay, only certain aspects of the bundle protocol are required. The aspects concerned with data security and are ignored at this time since they do not directly relate to the focus of this research.

2.5.4 Routing Protocols

Given the unique aspects regarding the DTN framework, routing protocols are crucial to node discovery and data delivery. As this is a hot topic of research, there are many protocols being constructed to meet the needs of the given application. The protocols focus on the different types of contacts involved and are classified as either forward or replication based. The types of contacts vary from scheduled or predictable to intermittent or opportunistic [26, 27]. For example, scheduled or predictable would describe a bus schedule and the way the buses move according to a schedule in a predictable way allowing their location to be known at any given time. Whereas intermittent or opportunistic contacts are new unexpected contacts that are within in transmission range and they can be used in an advantageous way for routing data. Current research is focused on replication based approaches and therefore they are the most common. Epidemic, ProPHET, MaxProp, Spray and Wait, and RAPID will be discussed [27-31].

Epidemic routing uses a flood-based approach in which nodes within communication regions continuously replicate and transmit messages when new contacts are met. The messages then spread to other regions through adjacent nodes until the desired destination is reached. This approach is typically used as a baseline for comparison with other approaches being developed [27, 28].

The PRoPHET protocol relies upon the history of encounters and transitivity by using probability to determine the end-to-end paths. This protocol exploits the non-randomness of real-world encounters by maintaining a set of probabilities for successful delivery to known destinations in the DTN (called delivery predictabilities). Replications of messages occur during opportunistic encounters if the noted that does not have the message appears to have a better chance of delivering it [29].

RAPID uses a utility function to optimize based on the expected contribution of each packet toward a given metric. The overall protocol is composed of the four steps of initialization, direct delivery, replication, and termination. Initialization is used to exchanges metadata is to estimate the overall packet utilities. Next, direct delivery of packets destined for immediate neighbors are transmitted. Then, packets are replicated based on the marginal utility. Finally, the protocol ends when contacts break or all packets have been replicated [30].

In MaxProp, when new contacts are discovered, messages are compared and if messages do not exist in one of the contacts they are attempted to be replicated and forwarded by the contact possessing them. So when two nodes meet, they exchange their estimated node-meeting likelihood vectors and shortest path is calculated using weight factors to determine the path cost. Then these costs are calculated over all possible paths to the desired destination. This cost approach creates and maintains an ordered-queue. This intelligent queue management approach is based on the destination of each message and the estimated likelihood of a future transitive path to that destination. Also, decisions are made as to which messages are transmitted first and which should be dropped first. This algorithm was developed, tested through simulation on traces from the UMassDieselNet testbed and then compared against four other routing protocols. The findings show that MaxProp outperforms the tested algorithms by delivering more packets to the destination while maintaining the smallest latency [31].

Spray and wait uses both replication and forward based routing in an attempt to benefit from each approach. This method achieves resource efficiency by controlling flooding by setting a strict upper bound on the number of copies per message allowed in the network. Once disseminated, the copies spread through the network looking for the destination. If it is reached, the search is over, if not, the process is repeated [28].

The aforementioned routing methods are just a few of the most commonly used within the DTN network topology framework. This thesis used some of the principles from an opportunistic, per hop routing approach. However, the approach used here is distinctly different. A novel cost rank matrix function approach was used with an ordered-queue to determine the most opportunistic route to take to find the end-to-end path. This will be discussed in detail in Chapter 3.

2.6 Cognitive Networks

By definition, "a cognitive network has a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account the end-to-end goals [32]". This added cognitive ability provides the network a forward-looking, instead of reactive observation, that allows problems to be addressed before they can occur. In general, taking this proactive approach enables preventive conflict resolution which helps maintain a stable operating environment. The difficulty with employing this technique comes from being able to identify which proactive measures to take. In this construct, this would be based upon ability to comprehend past outcomes and assimilate an appropriate response for future actions. This in effect would require the network to learn what an appropriate response would be given the particular situation.

One possible approach could be to use the end-to-end goals within a network wide knowledge scope. This would give the cognitive entity the ability to see all of the available options and make an educated decision. In most implementations, especially in a dynamic network, this is not possible, and a “best guess” estimate is used (based upon what information is available). In the cognitive network implementation used in this research, *pseudo-cognition* is used. pseudo-cognition determines the next action based upon:

1) a pre-defined look-ahead window of knowledge and
2) currently available metrics.

Pseudo-cognition does not fully include the learning aspect of a cognitive network since past decisions are not evaluated for effectiveness. Only the NTO and ATO documents are used as a repository of knowledge, and are not updated as the mission progresses. It could be considered that the currently available metric used in this research, average percent time loiter,incorporates past decisions, so it a form of feedback. In general, this process can best be represented in Boyd’s Observe, Orient, Decide, and Act (OODA) loop for decision making [32].

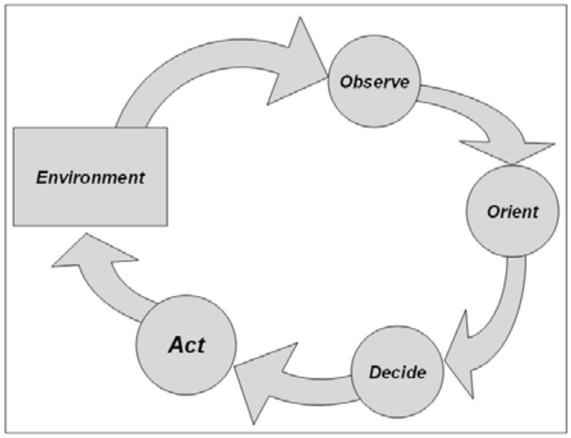

Figure 6. Boyd's OODA Loop

The environment block in Figure 6 is the network being monitored. The OODA loop behavior is accomplished by using network metrics and along with NTO and ATO knowledge as input to the decision making process (observe) and providing output in the form of a set of actions (act). As is evident in the diagram, the OODA loop requires feedback to be effective. Essentially, network conditions are observed, as a result the end-to-end goals drive the behavior of the system and the individual elements can either use this information for decisions separately or cooperatively.

2.7 Multi-Objective Optimization

Multi-objective optimization is a systematic technique used to find a solution to a problem with multiple possibly competing objectives. Typically, tradeoffs are made to find sets of acceptable solutions within a boundary limit, and of the solutions within the boundary, one is chosen. There are many methods for finding the optimal solution sets including stochastic, linear programming, goal programming, and game theory approaches. The most common are genetic and evolutionary based [33]. They use a set of candidate solutions which are modified by selection and variation and as a process they iteratively mutate or evolve towards a converging region [33, 34].

In general terms, a problem requires each objective being considered to have an objective function created that represents the particular task or goal that needs to be achieved. With this objective, there usually are various constraints that have to be considered. Multi-objective optimization (MOO) is a technique used to take possibly competing objectives, their constraints, and their possible decision and solution spaces and find a set of acceptable solutions common to both (called the pareto front) [33, 34]. The pareto front is the boundary line where the optimal solutions lie. Figure 7 shows the MOO tradeoff used for this research.

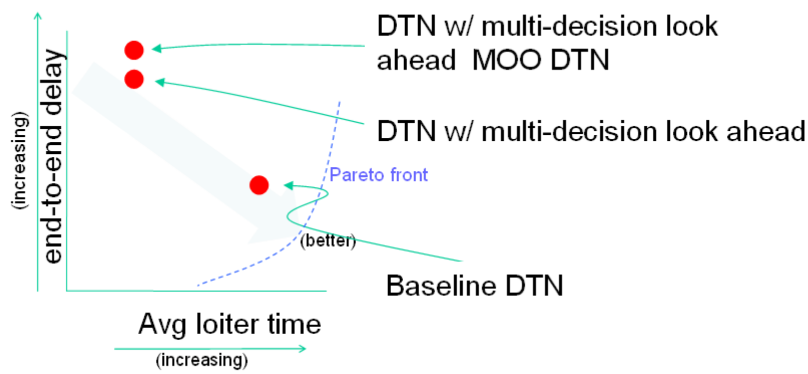

Figure 7. Example of Multi-Objective Optimization

Work in this area has focused primarily on static or dynamic problems addressed by post processing methods, meaning the analysis is conducted on data after the fact and an optimal solution is chosen. These approaches will not work in dynamic environments where decisions have to be made quickly. It is worth noting progress has been made in this regard by works from Goh [35], Liu [36], and Zheng [37]. However, due to time constraints and complexity with implementation, determining how to apply this work to the research problem will be reserved for future work. As a best first effort in this regard, this thesis will address an Air Force scenario that requires real-time response to dynamic network conditions. To address this dynamic problem, multi-objective techniques are garnered from robotic control theory; subsumption in particular, will be used to balance the competing objectives of the mission and network goals.

2.8 Robot Control Principles

Robot control can be defined as the ability to control movement based upon what is conditions are sensed in the environment. For example, this could mean if an obstacle is in a robot's path, the robot must possess the ability to sense it and then the control mechanism should determine a plan of action and then execute the plan in order to avoid collision. This is the essence of the "sense plan act" control approach [38]. Sense a circumstance, plan a response, and then execute the planned response.

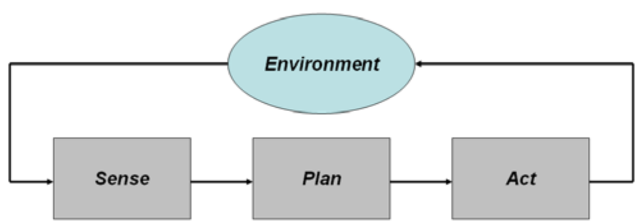

Figure 8. Sense-Plan-Act Diagram

Currently, there are four main approaches within the field of robot control: deliberative, reactive, hybrid, and behavior based [39, 40]. The control approaches range in complexity and response time from deliberative to reactive. Deliberative control means to use all available knowledge (internal representation of the world) to extensively plan for an event and try to anticipate and account for all possible responses within the realm of possibility. This can be very complex and computationally intensive as it requires complete and accurate knowledge about the environment in order to create a true world model from which to base its decisions.

Reactive control on the other hand, "tightly couples sensing to action" [41]. This makes the control very responsive relying upon rule based encodings to make decisions. This in turn is less computationally intensive and does not depend on an accurate model.

A third approach is the hybrid approach, which is a cross between deliberative and reactive. This is typically encapsulated within a three layer architecture with each layer addressing a certain component of the other approaches [38]. The lower layer is tied to the reactive control processes. These are typically the essential safety aspects of control (collision avoidance, reverse path) and therefore must be handled with a quick response time. The deliberative aspect focuses on the higher level planning of long term goals. The middle layer interfaces the upper and lower layers in an effort to sequentially implement the decisions of the other layers [38, 41]. This architecture was originally developed to follow the sense plan act approach while taking advantage of having all aspects included in the design.

An alternative to this hybrid approach is behavior based control. Behavior based control decomposes a task into modular components called behaviors. These "behaviors" run in parallel to one another and provide an object oriented approach as the structure can be easily changed by swapping out the modules for different intended response. Therefore this construct has the ability to run a hybrid approach capture all aspects of both the deliberative and reactive components [39]. Since all behaviors can run concurrently, sensor data must be acquired by all involved and decisions must be based upon a time of prioritized order in order to be executed in a synchronized way. The prioritized order in which actions are chosen is called action selection. There are two main categories regarding action selection: arbitration and command fusion. If two or more behaviors desire control at the same time, conflict resolution must be addressed with a behavior arbitration scheme. Arbitration is a formal way of choosing an action from a set of actions and then relaying it to the actuator to be executed.

There are many different forms of arbitration: fixed priority, command fusion, and several others which all have been used in one way or another with some form of success. The decision of which to choose is dependent on the particular application. The difficulty comes with implementing them, since they have to be software coded to adapt to the desired scenario. Command fusion deals with a voting based arbitration scheme in which the individual behavior cast votes based upon their likelihood to successfully complete their task. The votes are then fused to form an overall composite vote which must be interpreted to decide which action to implement. Rosenblatt, demonstrated this technique in the Distributed Architecture for Mobile Navigation architecture [42].

2.8.1 Subsumption

One of the most influential approaches in regards to behavior based robot control was proposed by Brooks in 1985, the subsumption architecture [43]. At the time he proposed it, this approach was a dramatic shift in the common view towards robot control techniques. It stemmed around the idea of decomposing the problem into a parallel structure of "task-achieving behaviors" instead of the normal sense-plan-act approach (where control flows sequentially) [43].

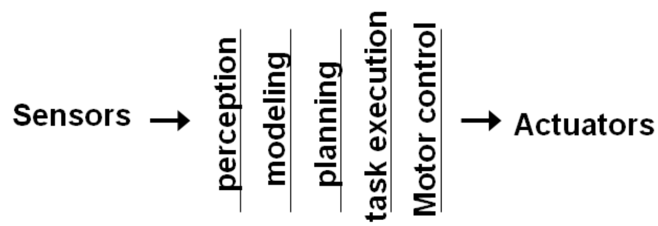

Figure 9. Traditional decomposition of mobile robot control system into functional modules [43]

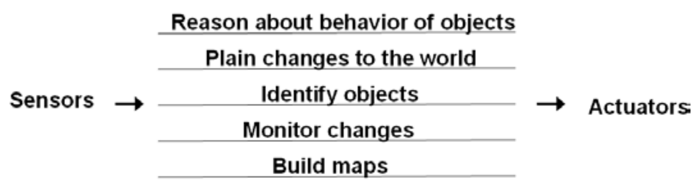

Figure 10. Decomposition of a mobile robot control system based on task-achieving behaviors [43]

This allowed behaviors to run concurrently in distinct layers with each layer working to achieve its particular goal. He also introduced the idea of "levels of competency" [43]. This concept revolved around the idea that increasingly more complex layers of the control system could easily be added to the existing architecture. Each time a new layer was added, the layer beneath it was a subset, this in turn built in a prioritized hierarchy. If the situation warranted it, this enabled upper layers to dominate the lower levels during operation. Coordination between layers was achieved when complex actions (i.e. higher layers) subsume or overrode less-complex behaviors (i.e. low level behaviors inhibit the higher layers). This hierarchy is maintained as a competitive architecture that uses rule based encodings and priority based arbitration.

This approach has been used in robotic control systems to achieve multi-objective optimization benefits [4]. In one work of particular interest [44], Burns used subsumption within controllers of mobile nodes in a DTN framework to act as intelligent intermediary hops to relay data. This resulted in increased network performance by creating new paths for data routing that didn't previously exist. This thesis will build upon this concept by allowing the nodes themselves to make autonomous decisions to improve not only network performance but also mission performance.

2.8.2 Unified Behavior Framework

The Unified Behavior Framework as presented by Woolley [4] will be discussed in detail as it is the foundation of the simulation used to represent the DTN scenario for this thesis. This controller will be incorporated in to a three level architecture so the desirable aspects of the reactive and deliberative elements can be used. This behavior based controller implementation provides the flexibility through modularity and code reuse. Also, the advantage of using a common interface that is able to observe the environmental conditions and dynamically swap behavior packages at runtime, enables the application of a particular behavior when it is most effective [4]. The sequencing for this implementation of the architecture is shown below in Figure 11.

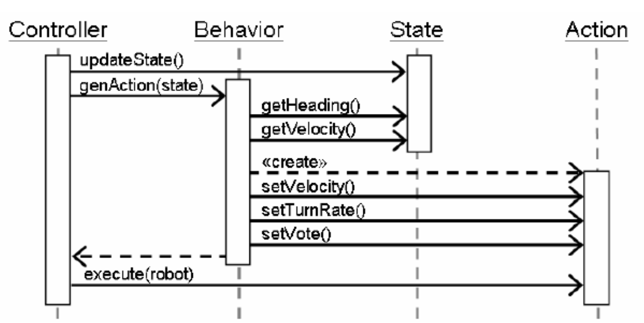

Figure 11. Sequence diagram of a controller using its behavior [4]

"First, the state is updated (updateState()) to represent the current conditions, then the behavior is asked to generate a recommended action (genAction(state)) then the proposed action is given the authority to issue commands directly to the motors (execute(robot))" [4]. The construct of the sequencing for this thesis will be explained later in Chapter 3. Next, the class diagram is shown for the UBF in Figure 12.

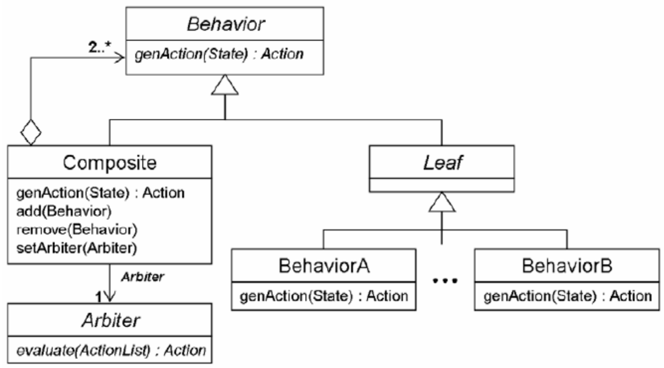

Figure 12. Class diagram for the Unified Behavior Framework [4]

Note that each behavior is modular and that if both behaviors (A and B) submit conflicting requests they can be formed into a composite behavior by the composite and arbiter nodes. Within this thesis, this construct is followed closely in which the *genOrbitchange( )* and *genData_routing( )* behaviors can also be fused into a composite behavior and both actions are executed in such a way that both the network and mission objectives are maximized.

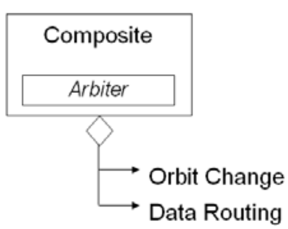

Figure 13. Composite Node Behavior [4]

2.9 Summary

This chapter presented the fundamental concepts and recent research in the areas of MANETs, DTNs, cognitive networks, multi-objective optimization, and subsumption. This introductory information should give the reader a familiarity needed to understand the concepts presented in the remainder of this document.

III. Methodology

3.1 Introduction

This chapter presents the methodology used to evaluate the proposed pseudo-cognitive multi-objective optimization approach and its effectiveness in balancing the mission and network goals within a given Air Force UAV surveillance mission scenario. First, the background, problem definition, and associated goals and hypothesis are discussed. Next, the approach will detail aspects regarding the given scenario and the overall UBF and software simulation structure. System boundaries and its services are then described followed by a detailed description of the performance metrics, parameters, factors, and workload. The experimental design and evaluation technique are then thoroughly discussed. Finally, a summary is presented to highlight some important aspects of the methodology

3.2 Background

As discussed within the literature review, there has been recent interest in the possibility of incorporating multi-objective optimization within a DTN topology. Burns [44] attempted to improve network performance by incorporating autonomous robots as intermediaries to data ferry the information to the destination. The approach within this thesis differs from the one Burns proposed, due to the fact that in this case the UAVs can change the actual DTN topology by making autonomous decisions and changing their orbit path in order to improve network performance.

The objectives or goals for this scenario are both mission and network related. The mission goal is to maximize the average percent loiter time over target and send the priority based images to HQ with the high priority images being sent as near real time as possible. Since this configuration is delay tolerant it must incorporate a store and forward approach by using UAV to UAV communication. The network goal is to minimize the end-to-end delay.

Subsumption is the approach incorporated to balance the multiple competing objectives. Recall that this architecture is a hierarchical structure based upon robotic control principles. This is implemented under the UBF construct and uses an arbiter to choose the required action among the recommendations from the individual behaviors. In this thesis, the behaviors evaluated are considered high level behaviors with the orbit change behavior taking precedence over the data routing behavior. The basic behaviors of the "zeroth level of competence" [43] used for primary robot motion control are assumed to exist and run underneath the higher layers. However, for this work they will not be evaluated or included in the overall effort.

Finally, the results for this effort will be gained by conducting the given implementation within simulated MATLAB environment. This will be discussed in great detail in the following sections.

3.3 Problem Definition

3.3.1 Hypothesis

As a result of mission goals dominating network goals, inefficiencies within the network can arise which in turn impact the overall effectiveness of the system. This can induce large queues, long delays, and even packet loss within the network. This work is based on the Surveyor SRV-1 robotic platform as this will be eventually be used to experimentally examine the interactions within this DTN topology scenario. The findings of this effort are used to compare and contrast the impact between mission and network goals. Cognition is incorporated into the system using a robotic control MOO algorithm subsumption approach to balance these two goals to achieve a more optimal solution.

This evaluation occurs within a particular scenario, using a pseudo-cognitive approach within a delay tolerant network configuration of autonomous robots (simulating UAVs). The goal is to choose the best orbit and routing actions to balance mission and network goals. The hypothesis is that the network will perform more efficiently (reduce end-to-end delay) and the time over target will increase (increase the average percent loiter time) under the MOO robotic control algorithm.

3.3.2 Approach

The scenario consists of a stealthy ISR mission conducted by a forward deployed special forces unit. Covertness during the mission is essential, therefore the unit will deploy small lightweight UAVs (similar to those shown in Figure 2) to conduct the surveillance.

If targets of interest are identified, images are taken and relayed to headquarters (HQ) to provide as near real time status as possible so command and control decisions are made promptly.

Since the targets monitored are out of direct communication range of the HQ, a DTN configuration is employed using a UAV as a data ferry to carry the information within transmission range. Also, note that since this is a network configuration and data routing is of interest, the term of UAV and node will be used interchangeably throughout this document.

Pseudo-cognition is added to the overall schema by incorporating aspects of an NTO within the deliberative layer of a three layer framework construct. This DTN configuration uses the UBF [4] for the controller and is then characterized by its effectiveness in balancing the mission and network goals.

Figure 14 depicts the three layer robot control architecture that could be used to implement this simulation on a real robotic platform such as the Surveyor SRV-1. The figure contains all requisites: a deliberator, a sequencer, and a controller. Although this thesis is primarily focused on the controller aspect, it could be executed within an overall three layer architecture, similar in concept as Pecarina [7] executed his HANC agent controller. As such, the NTO will be considered to be executed in the deliberator where the planning of the architecture exists. For this effort, the implicit function of the NTO (aspect of reducing end-to-end delay) was inherent and incorporated within the MATLAB code.

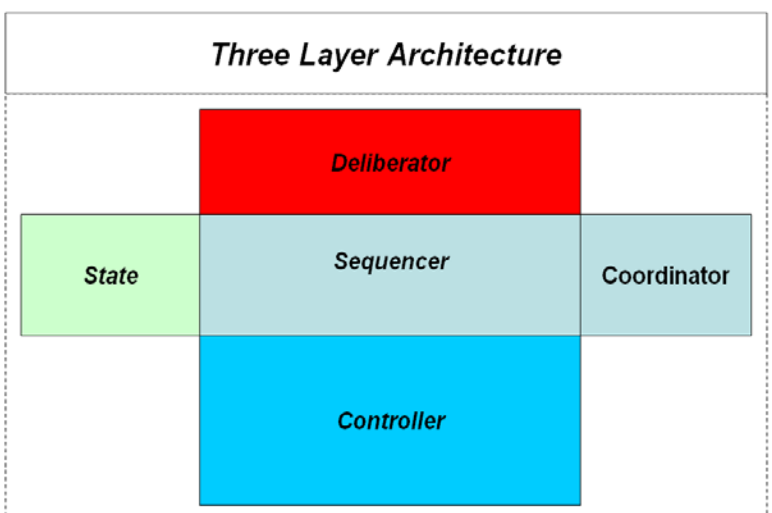

Figure 14. Three Layer Architecture

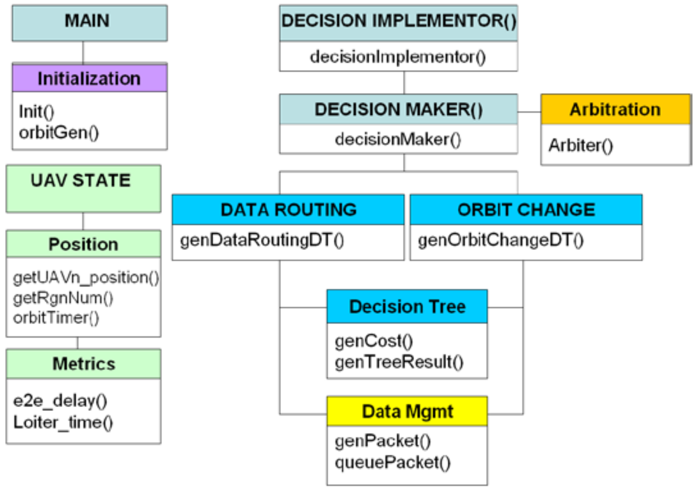

Figure 15. DTN Simulation Program

Figure 15 highlights the functions of the DTN MATLAB simulation program. In particular it highlights the significant groupings associated with the functions that encompass the entire program. Note that this structure is similar to Woolley's UBF controller [4] in (Figure 16) that the leaf behaviors are independent and they use an arbiter to form composite behavior where needed.

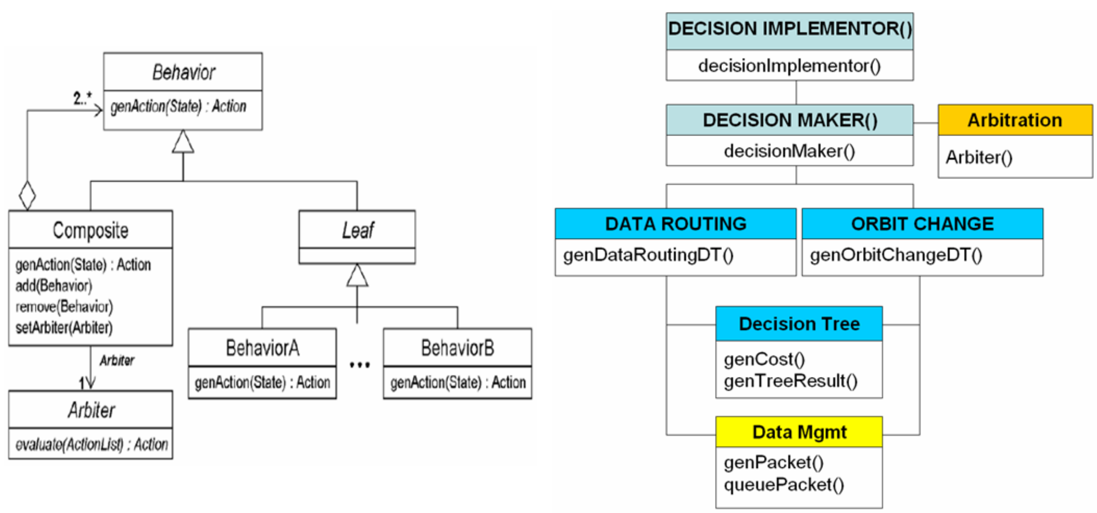

Figure 16. Program Implementation Comparison [4]

The simplicity of the design of the DTN simulation program is inherent in the logical structured flow. All aspects required to perform the necessary functions of the simulated DTN network are included. The functionality to generate separate orbit and data change decision trees, orbit timers to keep track of the current state and orbit, packet generating and queuing aspect to manage data flow, arbitration where needed, and a metric section to keep track of mission and network performance. Decision trees are structures used to create a hierarchal logical flow diagram that lists all possible options for a given condition. Within the context of this thesis, decision trees will be constructed and used to enumerate the possible options when making an orbit change and/or data routing decision.

3.4 System boundaries

The system under test is a delay tolerant network configuration that consists of two elliptical surveillance patterns (orbits) to observe three (3) targets of interest, four (4) UAVs, and the data destination (HQ). This is depicted below in Figure 17.

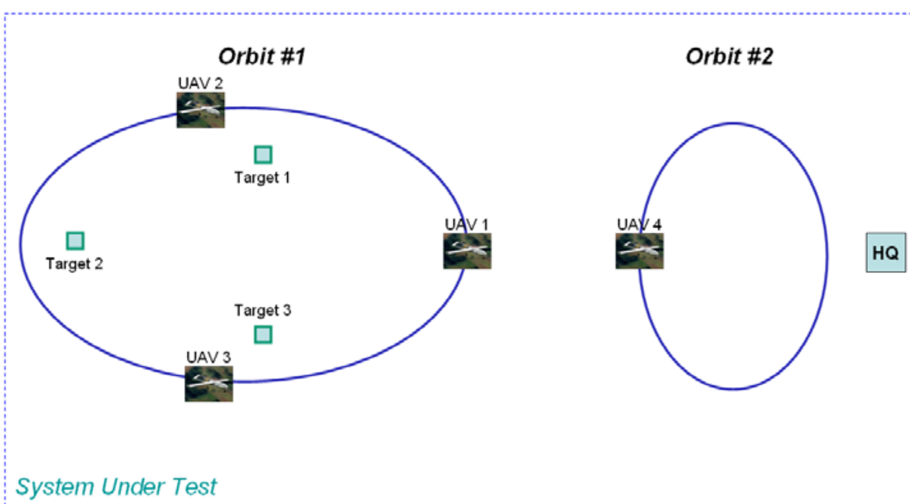

Figure 17. Delay Tolerant Network Configuration for SOCOM Mission

The orbits for the scenario have pre-defined fixed paths. Within the simulation each orbit has a dedicated number of “anchor points” where decisions are referenced. These two primary orbits have 150 and 90 reference points respectively. Based on the velocity of the UAV, it takes ≈ 1 second/anchor point. This equates to ≈ 2.5 minutes and 1.5 minutes to travel these complete orbit paths. Also, the distance between anchor points was broken down into 10 time steps to define a 6 inch time step as a way to approximate a straight line path on an orbit for experimental considerations. This resulted in 100msec (1sec/10) time steps in which the UAV must move. These discrete time steps were then used as the time window for which images could be detected and transmitted to make this a realistic environment for the Surveyor platform. This 100 msec window is used for calculating routes, deciding which route to take, and sending the image. Communications with each node was restricted by transmission power. This is defined in the simulation so the UAVs in orbit 1 can communicate with each other but not with the UAV in orbit 2 unless they are within the communication region portion of their orbit. Communication region refers to the region in which the distance between orbits is such that the transmission distance allows for UAV to UAV contact orbit to orbit. This is shown below in Figure 18. In this case, UAV1 could communicate with UAV4.

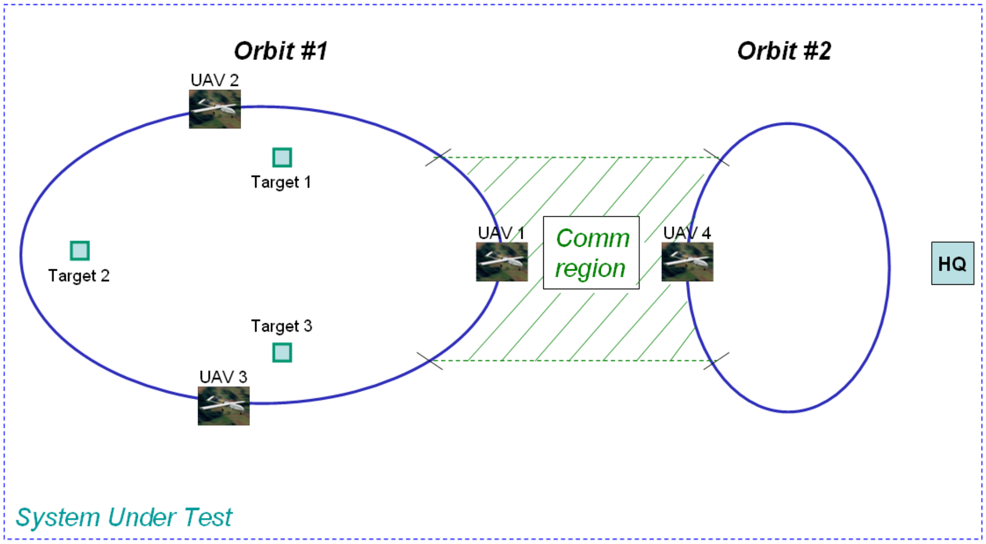

Figure 18. Orbit 1 to Orbit 2 Communication Regions

Within this scenario, the targets of interest remain stationary. Although when detected they can have a higher and lower priority of relevant information (i.e. aggressive posturing). Image detection and priority will be randomly generated. The component under test (CUT) is the "pseudo-cognition" (MOO implementation) within the overall network configuration. The CUT is shown below in Figure 19.

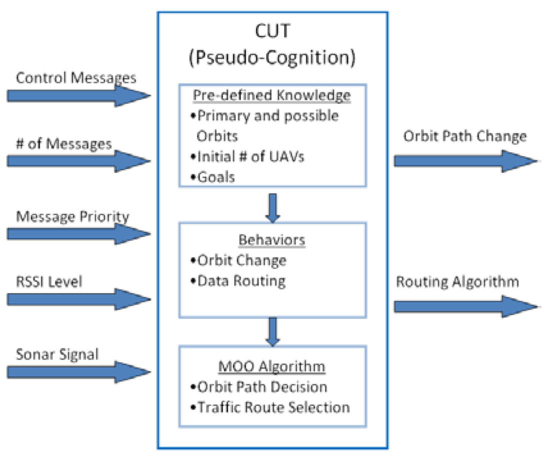

Figure 19. Component under test

The inputs for the CUT are the control messages, number of messages, message priority, path distances (i.e. RSSI levels and the sonar signals as represented in the simulation). The outputs for the CUT are the pseudo-cognitive robot decisions, a network type decision (routing algorithm) and/or mission type decision (orbit path change). In this component, the pseudo-cognitive aspect is implemented within the subsumption MOO approach. The essence is captured with an algorithm within the corresponding implementation hierarchy. This allows use of a fixed priority arbiter to make action selections based upon the inputs from the associated behaviors. The associated behaviors are the orbit change and data routing behaviors. Each behavior is focused on achieving one of the overall goals (either reducing end-to-end delay or increasing the average percent loiter time). These behaviors are analogous to objective functions under normal MOO terminology in that given a particular scenario and network state, each behavior (objective function) has their own optimum solutions in a solution space. This algorithm provides a way to converge on a list of acceptable solutions and then selects solutions that improve both.

Figure 20 illustrates the optional orbit paths and the UAV direction along the orbit (dashed line) that may be taken in effort to optimize the particular goal (mission or network) and improve overall system performance. The optional orbit is either chosen to either increase the average percent loiter time over a specific target or in attempt to decrease end-to-end delay. For example, if UAV2 takes Orbit #3 instead of staying on Orbit#1 UAV2 will have more opportunity to take images of Target #1 than if staying on Orbit#1. This will in turn increase the overall average percent loiter time since this is a ratio of image opportunities versus total simulation time steps. This will be discussed in additional detail in Section 3.6.

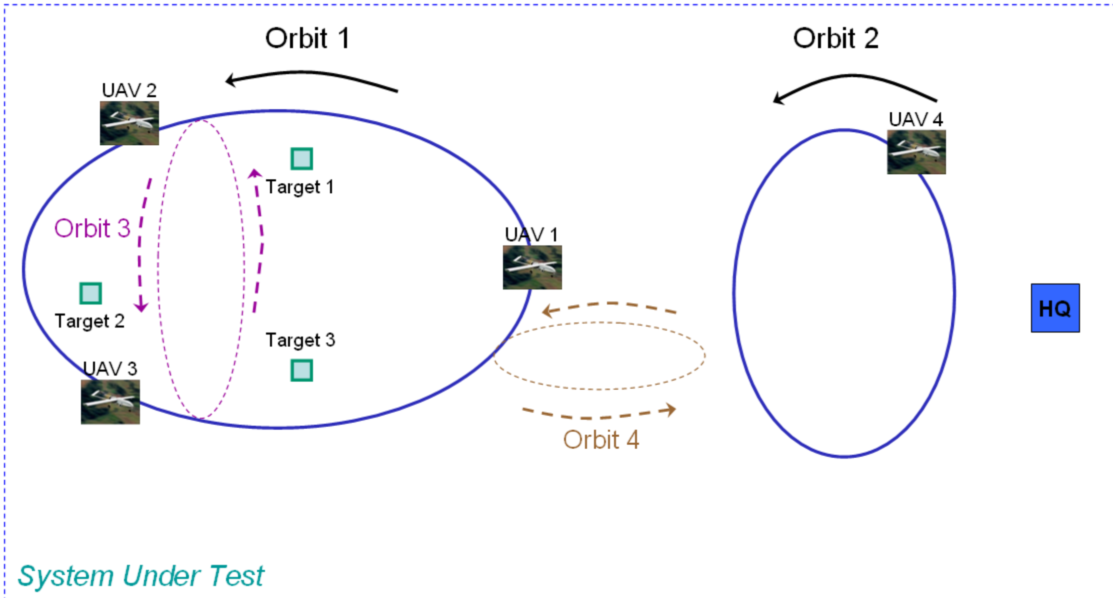

Figure 20. Optional orbits for the pseudo-cognitive MOO approach

3.5 System Services

The services this system provides is capturing and sending the detected image from a target to HQ within a delay tolerant configuration. This service is directly related to the mission and network goals.

3.6 Performance Metrics

The metrics for both the simulation and future experiment are the end-to-end delay, throughput, and the average percent loiter time. The end-to-end delay is measured as an overall delay (comprised of the low and high priority delays) of the entire system. The average overall delay is calculated as follows:

$$t_{End-to-End\ per\ image} = t_{HQ} - t_{Source}$$

$$t_{End-to-End\ Overall\ Average} = \left(\frac{\sum_{n=1}^{N} t_{n_{End-to-End\ per\ image}}}{N} \right)$$

Equation 1. Overall Average End-to-End Delay

Where t_{Source} is the time the image was captured, t_{HQ} is the time the image was received by the HQ queue, and $t_{End-to-End\ per\ image}$ is the delay for the image from source to destination. Therefore, $t_{End-to-End\ Overall\ Average}$ is the total overall average end-to-end delay with N being the total number of images captured.

The throughput is measured as the amount of data (images/min) that is transferred across the network from each node separately and through the entire system as a whole. The average percent loiter time is a ratio of the sum of the number of images opportunities for each UAVs (*K=3* in this case), and the total number of simulation time steps multiplied by the number of UAVs . In this representation, the answer is multiplied by 100 to convert the ratio into a percentage.

$$Overall\ average\ percent\ loiter\ time = \left(\frac{\sum_{k=1}^{K} number\ UAV_k\ image\ opportunities}{Total\ number\ time\ steps \cdot K} \right) \times 100$$

Equation 2. Overall Average Percent Loiter Time

3.7 Parameters

System parameters are all items that can have an effect on system performance. The system parameters for the SUT are the area of operations, attenuation level (dB), transmission power, information priority level, orbits, number of UAVs, UAV positions, UAV speeds, and target locations, background traffic, and the MOO algorithm. The listed parameters have a direct effect on the network topology since this is how the nodes are established and dictates whether they are able to communicate. In a DTN network this aspect is extremely important since this determines if an intermediary node can be reached to ensure a relay to the end destination.

The area of operations for the simulation was defined as a scaled map to be used for future experimentation with the Surveyor SRV-1 robotic platform. The boundaries were fixed and defined as in Figure 21 as 270 sq feet. A preliminary test was conducted with the robots to determine the overall footprint. The background traffic of the experiment is fixed at zero and will not be considered. The attenuation level is set to a fixed distance in the simulation (corresponds to a fixed distance in the experimentation that would be equivalent to using a -30dB attenuator) to limit the range of communications of the UAVs as depicted in Figure 21.

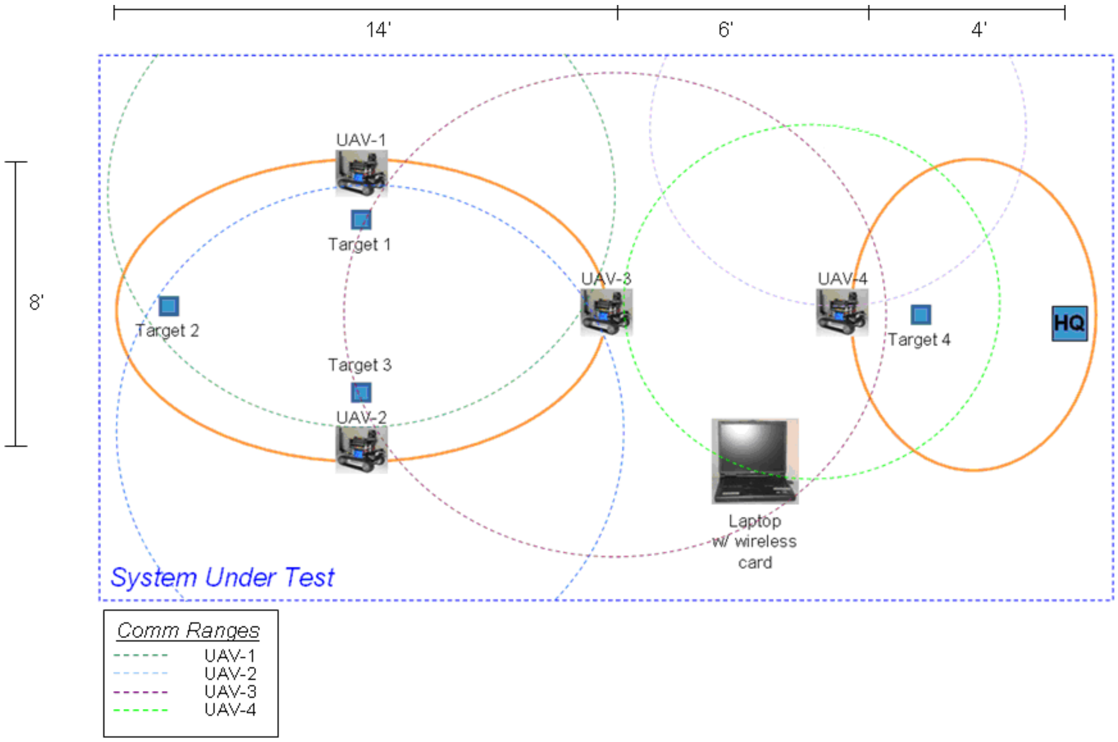

Figure 21. Communication Regions of Simulated Robots/UAVs

3.8 Factors

The factors of the system are the properties, which when changed, can impact the performance of the system. These include both system parameters and the workload parameters. The system and workload parameters for the SUT are further described, with the fixed experimental parameters factor/level table displayed in Table 1.

Table 1. Factors and Levels

Factors	Level 1	Level 2	Level 3
Workload	5 images/min	10 images/min	N/A
DTN Configuration	Baseline	Look-ahead approach	Pseudo-cognitive MOO

The orbit locations and distances, number of UAVs, and UAV speeds are considered secondary factors in this research since they will not be quantified or varied directly. The UAV locations are also secondary factors; however their movement is inherent in the design and will be performed as a direct result of the implementation of the primary factor. In this case, the DTN configuration is the primary factor because it decides how these secondary factors are changed and thereby directly adjusting the overall network to meet mission and network goals. The pseudo-cognitive MOO subsumption approach (as it departs from the baseline of a normal DTN) optimizes the current situation based on present inputs to directly affect the system performance and the overall end-to-end delay metric. The workload is also considered a primary factor since, when varied it adjusts the demand on the system therefore impacting the end-to-end delay metric.

3.9 Workload

The workload for the simulation and experiment are based on high priority and low priority images to be sent to HQ via the DTN network. The priority of the images is defined based on significance of action by the targets of interest under surveillance. A high priority image is representative of a significant event occurring (change in posture of target is detected). A low priority image is defined as something of interest but not as significant. Within this experiment there will be no actual detection of the high and low priority events, however these events will be randomly generated using the MATLAB pseudorandom number generator which produces uniformly distributed values over the interval between 0 and 1. Therefore, when triggered a detection is assumed to be successful and an image is taken by the UAV, and based on the given priority level is routed appropriately. The size of the images is based on the resolution of the camera board for the SRV-1 robot. Each image captured by the camera is approximately a 320 x 240 bit ≈ 5-10k bits/image.

Therefore two levels of abstraction are done to get to our simulation setup. First, we are assuming that the surveyor robots can represent UAVs. This is assumed reasonable because we are mainly testing mobility patterns of a DTN network and their affect on the overall network. Secondly, we are abstracting away the robots within a simulation environment. This is a safe assumption since the size of the image isn't critical in this setup. For this treatment, each image is considered to be a set size with a packet being an entire image. This is a safe assumption since we are really only interested in the mobility patterns.

The priority and time stamp of the images is initially assigned deterministically by way of the *RAND* function within MATLAB and the current time step within the simulation. Orbit location and the UAV responsible for the image detection are pre-defined for simplicity of experimentation and simulation. The workload level is generated based upon the available target region possibilities and the probability defined using the MATLAB *RAND* function. The *RAND* function is based upon the Mersanne Twister pseudorandom number generating algorithm developed by Makoto Matsumoto and Takuji Nishimura in 1996/1997 [45]. Within the MATLAB simulation environment this generator returns a pseudorandom, scalar value drawn from a uniform distribution on the unit interval [45]. Therefore, the images are captured and have a priority assigned to them within the main control loop of the program. The pseudocode for the image detection process is shown below in Figure 22.

```
If in target region
    • Determine if image is detected (detected =1, not detected=0)
      If detected (probability > 0.98)
        Set image detected = 1
          • Generate a priority (High =1, Low=0)
            If high priority (probability > 0.98)
                Set the image priority = 1
            else low priority (probability < 0.98)
                Set the image priority = 0
      else not detected (probablity <= 0.98)
        Set image detected = 0
else not in the target region
end

If image detected == 1
    generate a packet
else imaged  not detected
end
```

Figure 22. Pseudocode for the Image Detection Process

Note that the image detection process randomly generates both detection and priority. By adjusting the threshold limit of the random number (from 0.99 to 0.98) this resulted in two distinct levels at ≈ 5 images/min and ≈ 10 images/min. These were chosen to portray a significant demand on the system in order to accurately characterize the system response due to the specified utilization levels.

The target regions are where images can be taken if the above conditions are met. The regions are defined based upon an observable distance limit (5 ft) from the anchor points on an associated orbit (1 or 3). Figure 23 details the target regions with blue lines. The endpoint coordinates for each region are listed and therefore any anchor point encountered between the endpoints is also considered a possible image opportunity. Lastly, recall that between each anchor point are approximately ten time steps and therefore are considered image opportunities as well. Table 2 identifies the average percent loiter time for the given orbits shown in Figure 23. Note that the average percent loiter time was calculated as described in section 3.6.

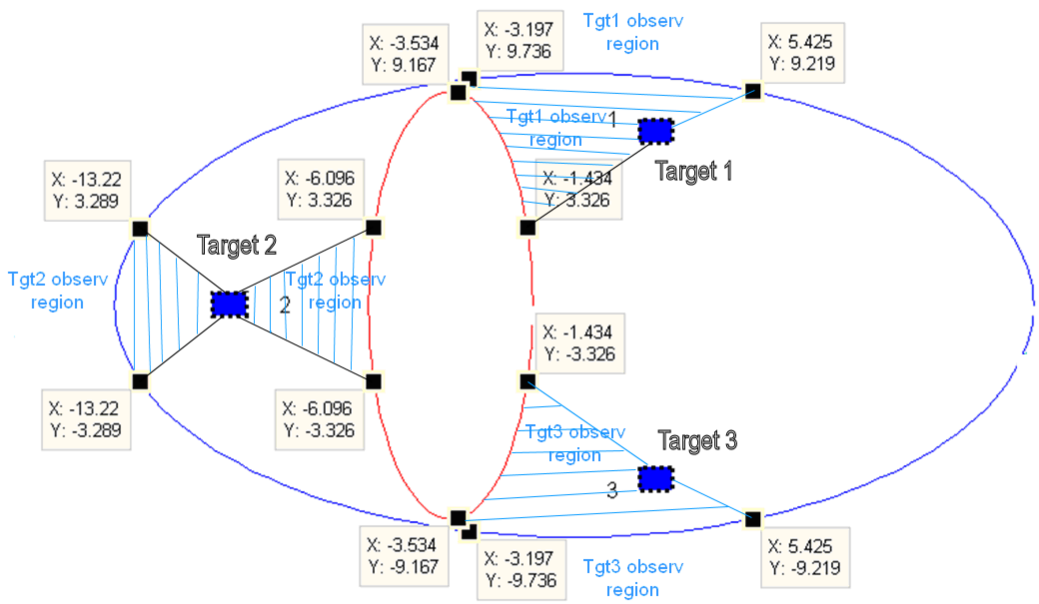

Figure 23. Target Image Opportunity Regions

Table 2. Number of orbit Image Opportunities and associated average percent loiter time

Number of Image Opportunities for Target Regions				*Total time steps per orbit*	*Average percent loiter time per orbit*
	Target 1	*Target 2*	*Target 3*		
Orbit 1	160	130	160	1500	37.5
Orbit 3	130	90	130	680	51.5

3.10 Experimental Design

For this experiment there were no real world data traces available to compare to regarding the effects of adding cognition as compared to a standard DTN implementation. As such, several assumptions, simplifications, and preliminary experiments were made in an effort to develop a viable compromise. Of particular importance, the orbits and UAV positions are considered to be pre-defined in the ATO. Image generation workload was set to an approximately constant rate by way of a random number generator with a uniform distribution. Constant rate refers to the approximate overall number of images taken per minute (5 and 10 images/min). For the simulation this equates to 1 and 2 percent of the total image opportunities. This in turn is approximately 160 and 320 images respectively.

The experimental design was incremental thereby each phase built upon the next. The design consisted of three phases: 1) design and implement the simulation using a standard baseline DTN approach 2) design and implement the DTN look-ahead approach based upon results from the baseline approach 3) design and implement the pseudo-cognitive process with the MOO algorithm using the look-ahead approach framework. Also, use any findings from the look-ahead approach to enhance the pseudo-cognitive approach. Finally, use these three approaches to determine the impact as compared to the baseline DTN approach.

In the first phase, the standard baseline DTN approach was designed and evaluated. This first step embodies a reliable approach since this configuration renders the most predictable results which were validated analytically. Within this approach the routing of data was accomplished within the 100 msec time window (this encompasses the calculation, decision, and routing of the image) between simulation time steps. This dictates the first hop in the data path is based upon the current position of the other UAVs. The decision process resulted in a simple decision tree based upon the current positions of the other UAVs and HQ. Figure 24 shows an example of the baseline DTN decision tree with UAV1 as the source making the decision.

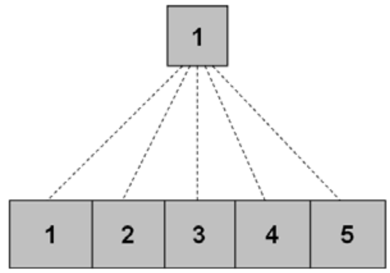

Figure 24. Example of a Decision Tree for the Baseline DTN Approach

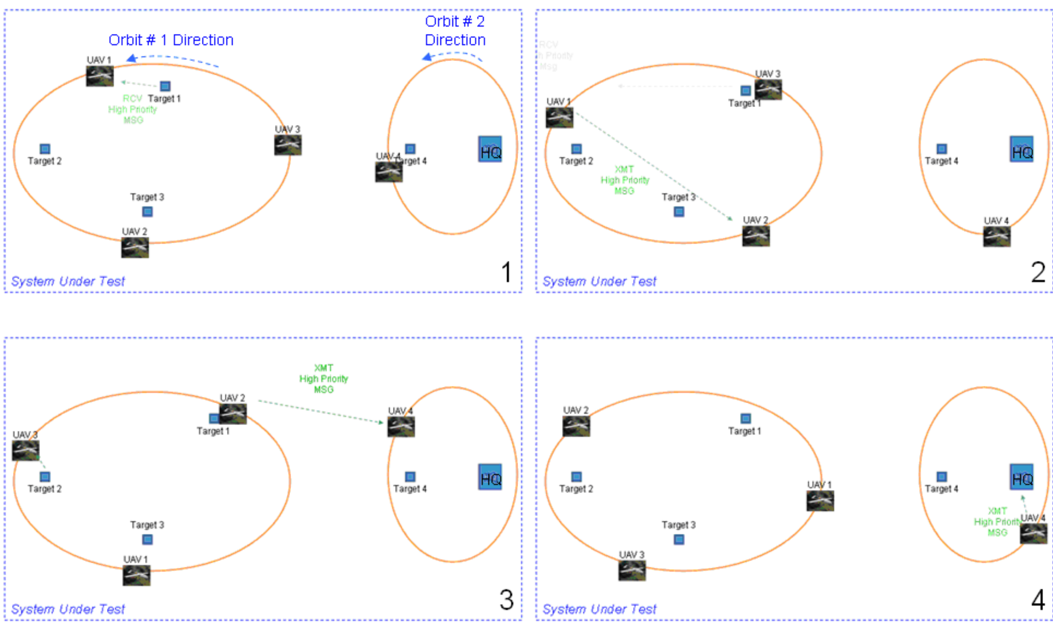

Figure 25. UAV to UAV Multi-hop Communication in the Baseline DTN Configuration

Figure 25 illustrates an example of a possible scenario that could occur during the routing of images within the baseline DTN approach. In this situation, a high priority image is taken at target 1 by UAV1 and sent wirelessly to the end destination (HQ) using a multi-hop approach. The sequence of this scenario is indicated by the number in the lower right corner of each frame (1-4). In sequence 1, UAV1 detects the image then routes the image to UAV2 in sequence 2. In sequence 3, UAV2 sends the image to UAV4 as they are both within the communication region between orbits 1 and 2. Then finally, UAV4 relays the message to HQ where the image is queued and the end-to-end delay is calculated.

In the second phase of design, the look-ahead approach built upon the baseline DTN approach. The look-ahead approach refers to the number of anchor points and hops into the future the UAV basis its next decision upon. The number of anchor points refers to the granularity of the hop size and hop means the next node the information is forwarded to. Therefore, two parameters are defined here: 1) the granularity of the step size (number of anchor points) and 2) the number of hops in the data route.

For this implementation these parameters are referred to as *step_size* and *steps_look_ahead*. For clarification, within this approach all decisions are calculated and executed within the given 100 millisecond (msec) simulation time step. Therefore, although the look-ahead calculation is in addition to the current position decision that was accomplished in the baseline DTN approach, it is still executed within the same 100 msec time window. This results in the first decision (baseline approach) being calculated based upon on current positions and captured within the first level of a decision tree. Then the look-ahead calculation is based upon the next hops (beyond the baseline level decision) next hop possibilities. In turn, this requires an additional calculation to the first step of baseline DTN routing determination. For example, if UAV2 has a packet that needs to be routed and the UAV is currently at [11.66;5.33] (x coordinate; y coordinate), and the baseline DTN decision was determined for UAV2 to store the packet, the next step within the same simulation time step would be based upon the look-ahead approach. In this example the granularity is set to *step_size* of 5 and *steps_look_ahead* of 1. This is shown in Figure 26.

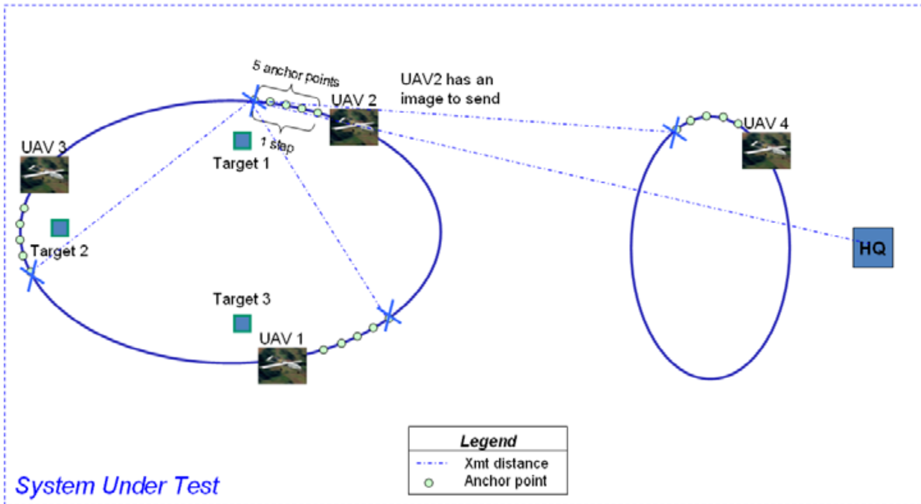

Figure 26. Example of Look-ahead Approach *step_ size* = 5 and *steps_look_ahead* = 1

This would result in a decision tree being created. UAV2 would be the source at the top of the tree, with the baseline DTN decision as the option to store the message, chosen in the first level of the tree. The other option choices would then be listed in the second level of the tree. The example (Figure 27) with the UAVs and HQ listed as nodes (UAVs = 1-4, HQ = 5).

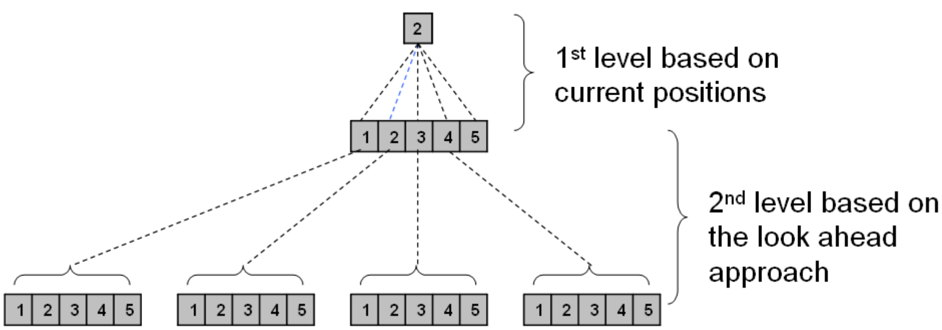

Figure 27. Decision Tree look-ahead Approach

In the third phase, the results of the first and second phases were used to identify the significant factors for implementation. This led to further refinement of the overall process. The look-ahead approach framework was augmented with additional algorithms to incorporate the pseudo-cognitive MOO algorithm aspect. The significant difference between the look-ahead and the MOO algorithm approaches was the latter incorporated pseudo-cognition to determine what decision would best improve the mission and network goals. This process was enhanced with both the MOO algorithm and additional optional orbit choices. The optional orbit choices were designated in such a way as to offer improvement for a particular goal. In this case, orbit 3 was designed to improve loiter time and orbit 4 to reduce end-to-end delay.

The MOO algorithm was designed to make decisions based upon the subsumption architecture approach. In this case, the mission goal of orbit choices dominated (a level above) the network goal of data routing.

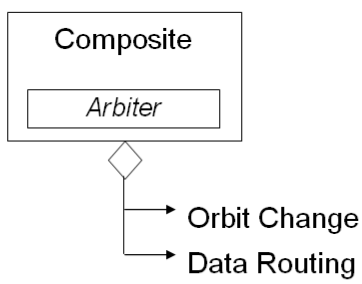

Figure 28. Subsumption Approach Architecture [4, 43]

As stated, the orbit change layer (behavior) dominates the data routing behavior therefore when an orbit change decision point is encountered, the orbit change behavior gets to determine what option best satisfies its goals first. In this design there are two orbit choices and therefore, two orbit change decision points (Figure 29).

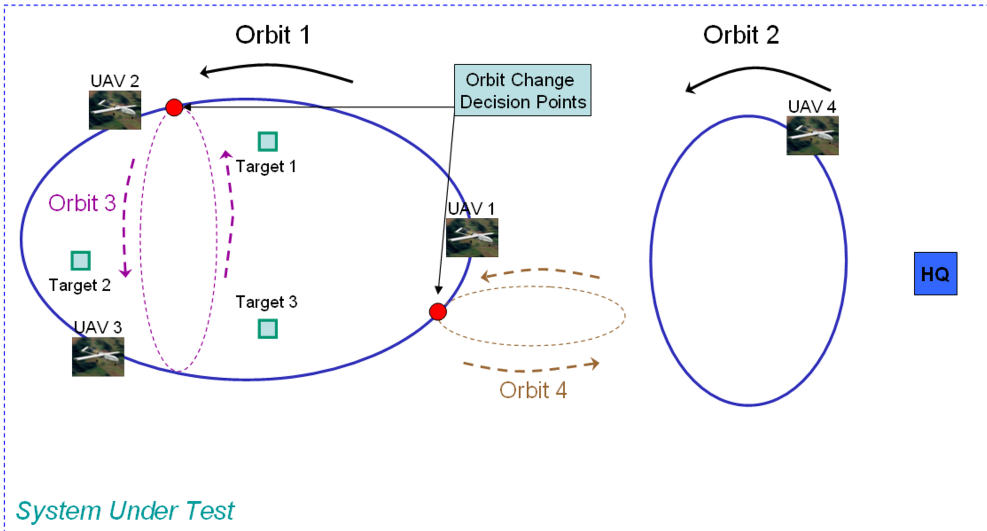

Figure 29. Diagram Labeling the Orbit Change Decision Points

If orbit change point encountered

- *Determine if loiter time threshold value is met*

  If loiter time < threshold

  - *The orbit change layer dominates and chooses the orbit choice with the least cost (inverse of overall percent average loiter time)*
  - *Set the orbit choice to the orbit with the least cost*

  elseif loiter time >= threshold

  - *Either orbit choice can be chosen, orbit change layer inhibits the output of the data routing layer*
  - *If there is a packet to route, data routing layer generates a decision tree for each orbit*
    - *Least cost path is chosen between the data routing decision trees*
    - *The path corresponds to a particular decision tree and therefore a particular orbit*
    - *Set the orbit to the path chosen*
  - *else keep the UAV on the same orbit*
  - *end*

elseif orbit change point not encountered

- *If there is a packet to route, a data routing decision tree is created, the least cost path is determined, and the packet is routed along that path*

end

Figure 30. Pseudo-Cognitive MOO Algorithm Pseudo code

The pseudo code for the pseudo-cognitive MOO algorithm is shown in figure 30. The pseudo-cognitive approach is structured so the orbit change behavior only is active when an orbit change decision point is encountered.

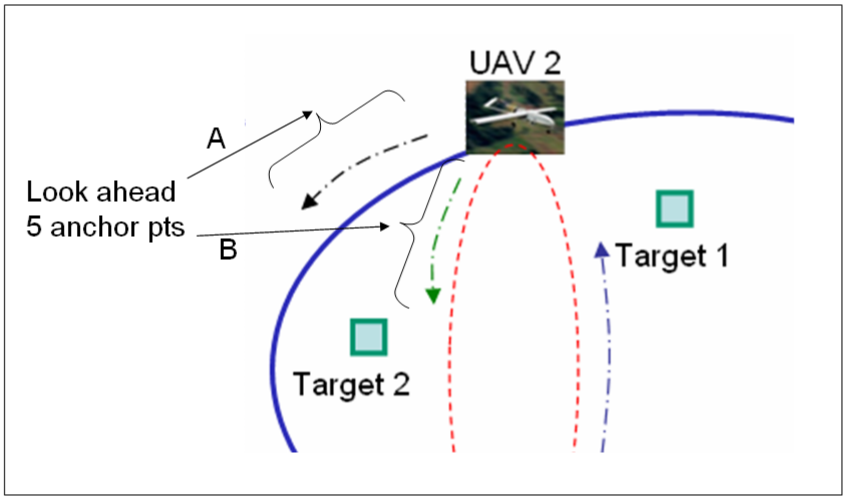

Figure 31. Example of MOO Algorithm Orbit Change Point Decision

Once this occurs, a comparison is made between the loiter time threshold and the current running average of the system loiter time (global metric). If the current loiter time is less than the threshold, the behavior layer subsumes or suppresses the output of the data routing layer temporarily, identifies which orbit path has the least cost (to increase loiter time), and sets the orbit change flag so an orbit change will occur on the next simulation time step. The least cost is determined by taking the inverse of the overall average percent loiter time for the given orbit. If the current loiter time is greater than or equal to the threshold, the orbit change behavior doesn't know which orbit to choose. This is where the cognitive aspect takes over. Since the orbit change behavior doesn't know which orbit choice is the best choice to meet the overall goals, the *decisionMaker* inhibits the data routing layer output so the arbiter can use this information to enhance the overall decision. If the data routing layer doesn't have a decision tree generated (no image to send) there is no additional information that can be provide and therefore the decision by the orbit change behavior is to stay on the current orbit.

However, if there is an image to send, a flag is triggered when the orbit change decision point is encountered, thus requiring the data routing decision tree to create a decision tree for each possible orbit. The *decisionMaker* then sends all decision trees to the arbiter so arbitration can occur. The *arbiter* identifies the lowest cost path between the data routing decision trees which relates directly to the orbit the UAV should take. The arbiter then sets the resulting decision tree for the orbit change behavior so this orbit change is taken on the next simulation step. If there is a tie between costs between the decision trees, the arbiter chooses for the UAV to stay on its current orbit.

In the following discussion, the MOO algorithm was discussed and under this implementation, when an image needs to be routed, the decision is based upon the information gleaned from knowing all participant's positions a certain number of steps into the future. This can be beneficial if additional information is obtained (i.e. possibly within the communication region or not). This is highly likely in this case since this is a military operation and all events are scheduled and predictable.

As a result of this systematic process, a full factorial design was used which consisted of testing two distinct workloads and three overall network configurations. This resulted in the need to conduct six (6) standard experiments to test all possible experimental combinations.

3.11 Evaluation Technique

The primary evaluation technique chosen was simulation using MATLAB software. Cognition was infused within the simulation in the DTN routing process and overall topology control. This was done to investigate if the delay can be mitigated and if the mission could be accomplished without significant degradation to the overall network performance. Since this process was defined pseudo-cognitive, it was similar to other cognitive network concepts. This was incorporated by monitoring the current network conditions (with a *UAV_state( )* structure), identifying shortcomings in the metrics, making decisions whether to route or store the image, and determine where to route the image. The process then executed the decisions by choosing the next action, and evaluating the response of the resultant metrics. This implementation is not true cognition but it encompasses all attributes with exception being in regards to the learning aspect. This aspect of cognition was not implemented at this time but was left to future work. This was chosen due the complexity involved with dynamic MOO and the significant overhead associated with it.

3.11.1 Implementation Details

Each of the configurations under test have unique aspects and assumptions that must be understood in order to gain an overall understanding in the underlying framework. The assumptions are as follows:

1) All orbits are pre-defined in the ATO and communication regions are defined in the NTO, therefore the topology is known at a given time and position; complete global knowledge.

2) Queuing and mobility delay are the significant delay factors and therefore transmission and processing delays are negligible.

3) Image detection and priority assignment are binomial random processes.

4) Image fragmentation and reliable data transfer will not be implemented for ease of simulation.

5) Within the initial stage, all nodes are considered reliable and available.

The overall simulation that was created was comprised of MATLAB structures to represent the UAV state at any given time. As such, each time step of the simulation stored the UAV status within an array of the structures. This made it possible to access the UAV state for any prescribed time. This proved invaluable for debugging, verification, and validation of the simulation.

```
UAV_state(T_COUNT,i).uav                 %Indicates the UAVi
UAV_state(T_COUNT,i).uav_queue           %Indicates the UAVi queue
UAV_state(T_COUNT,i).num_images_rcv      %Indicates the number of images the UAVi has received
UAV_state(T_COUNT,i). num_images_xmt     %Indicates the number of images the UAVi has transmitted
UAV_state(T_COUNT,i).num_of_steps        %indicates the level of the tree desired
UAV_state(T_COUNT,i).num_of_routes       %indicates the number of possible routes for the given level
UAV_state(T_COUNT,i).orbit_change        %indicates if an orbit change decision pt has been encountered
UAV_state(T_COUNT,i).data_route          %indicates if an image needs to be routed
UAV_state(T_COUNT,i).orbit_tree          %indicates if an orbit change decision tree is required
UAV_state(T_COUNT,i).data_tree           %indicates if a data routing decision tree is required
UAV_state(T_COUNT,i).total_trees         %indicates the total number of decision trees required; used for indexing purposes
UAV_state(T_COUNT,i).orbit_changed       %indicates if an orbit change has occured
UAV_state(T_COUNT,i).route_mat           %indicates the UAV routes possible from the UAV making the decision
UAV_state(T_COUNT,i).current_position    %indicates the current position of both the source and destination UAVs
UAV_state(T_COUNT,i).future_position     %indicates the future position of both the source and destination UAVs
UAV_state(T_COUNT,i).distance            %stores the distance between the source and destination UAVs
UAV_state(T_COUNT,i).init_tree           %stores possible routes based on the transmission distance of the UAVs
UAV_state(T_COUNT,i).cost                %stores cost associated w/ possible routes indicated in the init_tree
UAV_state(T_COUNT,i).tree_result         %stores the result of the path chosen based on the cost
UAV_state(T_COUNT,i).time_count          %Displays the time count of the simulation
UAV_state(T_COUNT,i).avg_delay           %Indicates the average end-to-end delay
UAV_state(T_COUNT,i).loiter_time         %Indicates the average loiter time
```

Figure 32. UAV State Structure

As shown above in Figure 32, the state information included fields regarding every aspect of the state for the UAV. Of particular importance are the *T_COUNT* and *i* fields. *T_COUNT* is the time index of the simulation. This value was used to store and index the structure at every time instance in the simulation. The *i* value refers to the UAV number that the state is reflecting. Therefore at every time step within the simulation, a structure is created for every UAV and their states are stored within an array of structures.

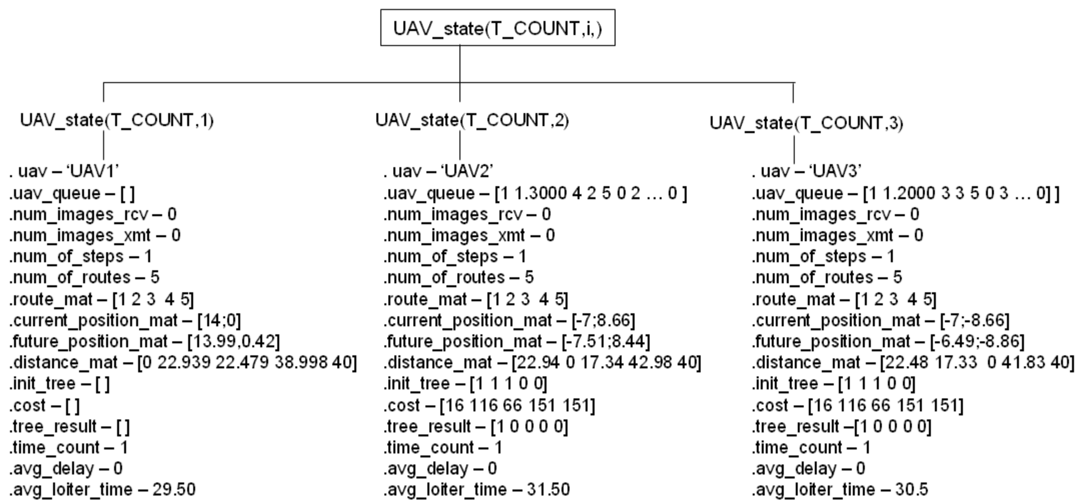

Figure 33. Example of the UAV_state structure

Shown above is a condensed version of the *UAV_state* structure with some fields omitted for ease of explanation. This example shows the details of how image routing is captured within the simulation. Here is a quick synopsis of what happens. An image is taken and stored in the *uav_queue*. Then the current positions are received and distance between the source UAV and the other UAVs is calculated and stored in *distance_mat*. Next, the possible communication links are determined by identifying all nodes within communication range. This is based upon the distances calculated and the possible transmission range of the UAVs. Next an initial decision tree is created and stored in the *init_tree* field (1 is a path; 0 is not). The cost function is then accessed and the link cost is determined and stored in the *cost* field. Finally, all cost values are evaluated and the lowest cost is chosen. This next hop node is then identified in resultant decision tree in the *tree_result* field. Then based upon the decision tree result the packet is either forwarded or stored.

A novel cost function approach was developed in order to place emphasis on the best routing path for a given circumstance. The cost function orders the choices as follows: direct to HQ (most desired), UAV4 (data ferry), and then the closest UAV to the communication region of contact of UAV4. This was determined by creating a rank matrix with distance and direction as factors to determine which UAV was the next hop for the image. Note in Figure 34 that the cost increases in A to D fashion.

This means that if the current/next hop position (depends if using baseline, look-ahead or MOO approach) of the UAV is at position A on the orbit, this is the beginning of the communication region to orbit 2 and therefore there is a higher likelihood that contact with UAV4 will be made therefore enabling a data transfer to orbit 2 and then eventually to the destination. The lower the cost given by the rank matrix results in more likelihood of data transfer to orbit 2, therefore cost A is more beneficial than cost D.

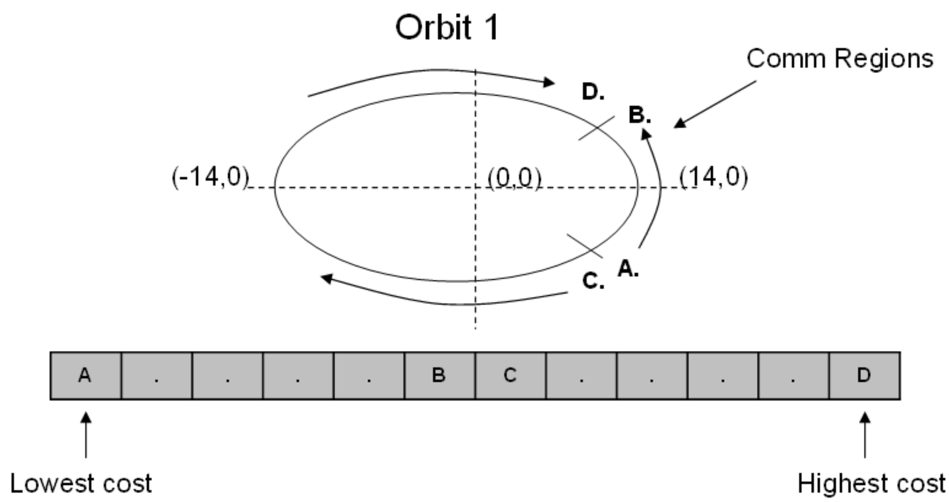

Figure 34. Example of an Orbit 1 Cost Function

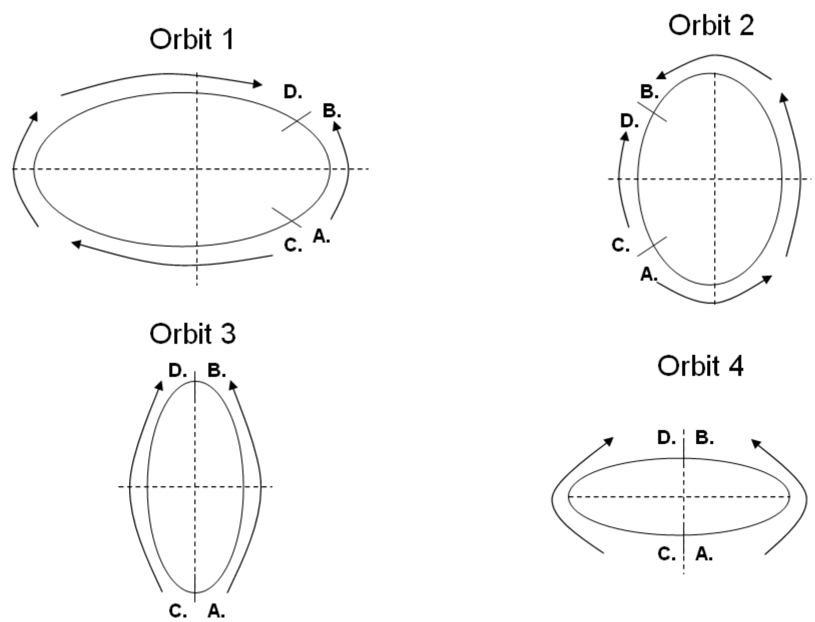

Figure 35. Cost Function Orientations for all Orbits used

Note that the cost function precedence is based upon the orientation of HQ. If it was located in a different position, these preference vectors would need to be recalculated.

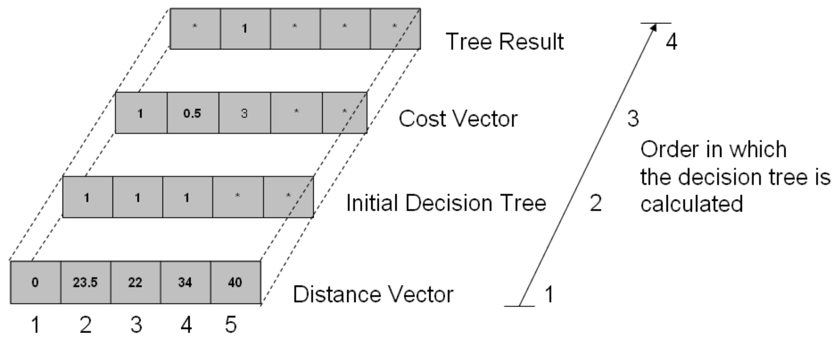

Figure 36. Decision Tree Process

Figure 36 illustrates the sequence of calculations that occur when generating the overall decision trees. The sequence goes from calculating the distances, generating the initial decision tree based upon distances less than the possible transmission distance, assigning cost values to initialized values within the decision tree, and finally choosing the lowest cost value and designating the routing choice in the resultant decision tree. This sequence is shown on Figure 36 as occurring in the order from 1 to 4. Since this is a major aspect of the simulation, further elaboration is required.

In reference to Figure 36, the distances between the current position vector and next possible position (possible routes) vector are calculated and stored in a distance vector in which index refers to the UAV#, the zero represents the source, and the other number is the distance from the source to the next UAV. Based on this information an initial decision tree is constructed for the distances that are less than the defined maximum communication distance (max range). In this case, the maximum range is defined as 24.5 and therefore UAV4 and HQ (index of 5) are not within communication range. Next, the cost function is evaluated based on the current position for each element of the initial tree with a value of one (indicating a possible communication link). The resultant cost is stored in the cost vector. Finally, the next hop is based on the lowest cost value and the image is routed accordingly.

As was defined earlier, routing for the look-ahead approach enhances the decision-making process by providing additional insight based upon a defined number of time steps into the future. This approach used the pre-defined orbit information to make an estimate of the next contact. This was accomplished in an attempt to improve the routing process. Additional layers are added to the decision tree for each step of look-ahead. The layers of the decision trees were stored within cells, for easy indexing and access.

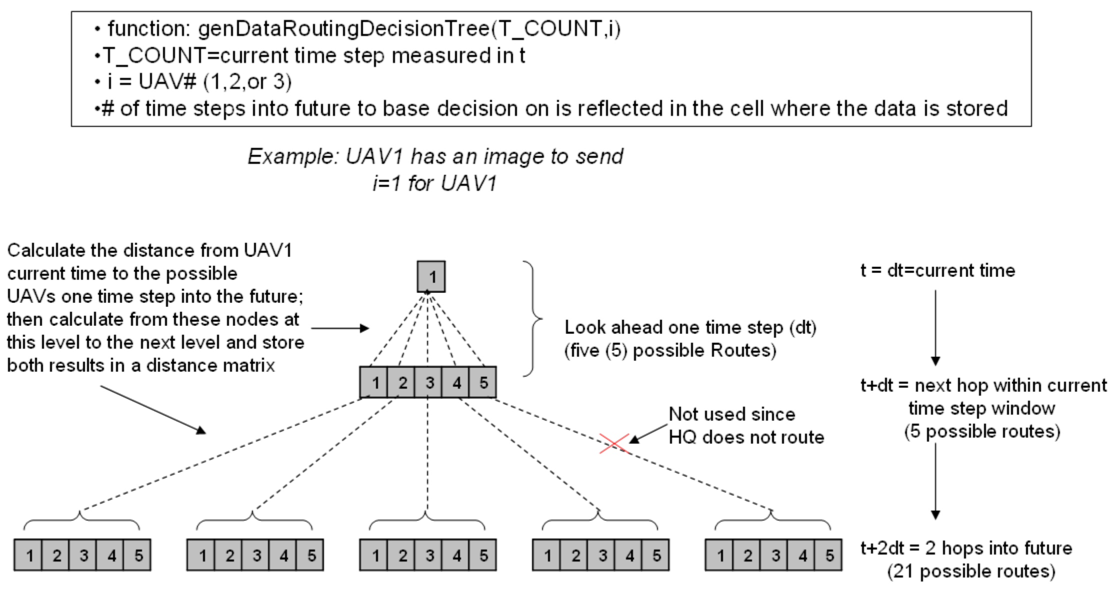

Figure 37. genDecisionTree Function

This figure illustrates the genDataRoutingDecisionTree function. This is how the decision tree is created. Essentially, once a UAV has an image to send, the genDataRoutingDecisionTree function is called to generate a decision tree to determine the next hop to forward the image to. Since the orbits and UAV transmission range are pre-determined, the distances between nodes can be calculated, making a matrix of the estimated distances of possible routes.

The time for the data transfers was based upon the step size (dt) for the simulation. This was determined to be 100 msec to represent a data transfer of approximately 11 msec/packet. In this construct, it is allowable to send one and receive two packets during a time step.

All images captured were stored as packets and consisted of the following data fields: image priority, time created, current simulation time step, source, destination, next hop, target id, UAV position and the forward flag. Note that the forward flag was created to prevent a packet being forwarded more than one time within a current time window.

The packet attributes are highlighted in Figure 38. These were based upon the fields within the "bundle" as depicted in a RFC 4838 DTN "bundle" construct [24].

1	1.1	2	1	5	3	1	13.99	0.42	0
Image Priority	Time created	Current Time Step	Source	Dest	Next Hop	Target Id	UAV X Position	UAV Y Position	Forward Flag

Figure 38. Packet Format

The queuing policy for the system was based upon an image priority, then time image created scheme. This was done in an effort to ensure the high priority images in fact were higher priority.

As was mentioned previously, applying standard multi-objective optimization methods to this application proved to be difficult because of the dynamic environment. Therefore, the subsumption architecture was used. Subsumption has been used in robotic control systems to achieve multi-objective benefits [43]. As defined earlier, subsumption is a layered approach, based on task decomposition where each layer works to achieve its particular goal. Coordination between layers is achieved when complex actions (i.e. higher layers) subsume less-complex behaviors (i.e. low level behaviors inhibit the higher layers) [43]. It is a competitive architecture that uses rule based encodings and priority based arbitration based on hierarchical priority.

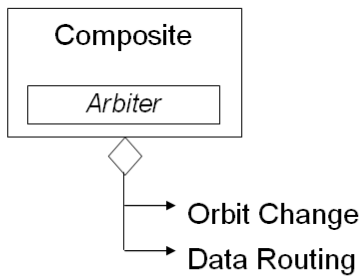

Figure 39. Subsumption Architecture [4]

This was implemented using the Unified Behavior Framework (UBF) [4]. Within this construct each task layer represented a leaf behavior. In this case, orbit change and data routing are the behaviors. The intention is that behavior selects the best option based on state information and sends the selection recommendation to the priority arbiter. The priority arbiter then serves as a merging mechanism to construct the composite behavior which models subsumption [4].

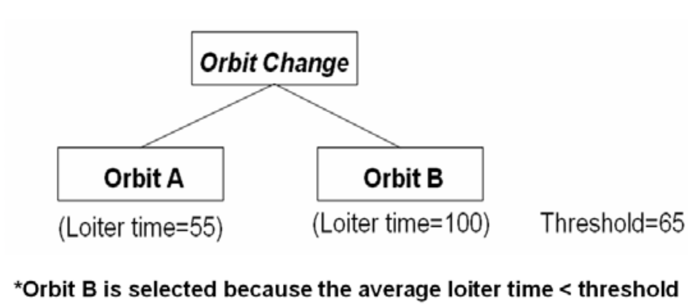

Figure 40. Subsumption Tree

The subsumption process for this simulation was captured within the decision tree construct. This is similar to the data routing process as described before with the exception that the orbit change cost is generated as the inverse of the average percent overall loiter time for each orbit choice. Within this process, each behavior determines the best action based on a predefined threshold. The overall composite behavior is then determined by the arbiter and executed. As such, this condition will occur when the current average loiter time is greater than or equal to the loiter time threshold. This occurs since the orbit change behavior doesn't know which orbit would best improve the system performance. This results in the arbiter creating a combination (composite behavior) of a UAV movement (orbit choice) and particular information route (routing choice). This was achieved by combining the levels of the decision trees (baseline in addition to each successive step of look-ahead), into a composite decision tree. Once this composite score was calculated for each orbit path, the lowest cost path was chosen.

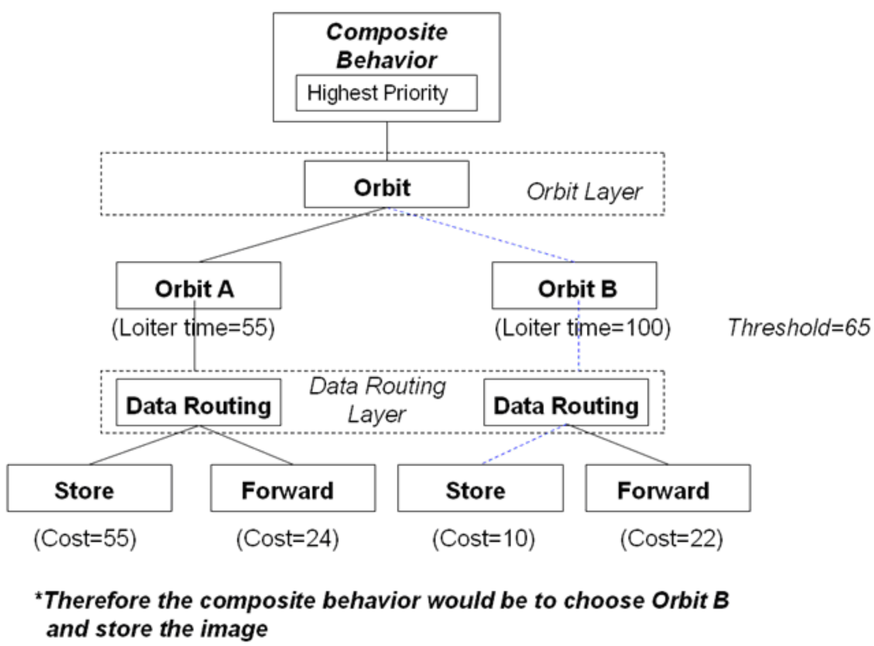

Figure 41. Example of an Overall Composite Behavior Diagram

In Figure 41 it can be observed that given a current loiter time of greater than or equal to 65 would result in uncertainty in the orbit change behavior. This would require the data routing layer output to be inhibited by the *decisionMaker*. This in turn would enable the arbiter to use arbitration between the two separate decision trees (Orbit A and Orbit B). A composite data routing decision tree would be required for each orbit. This could be a summation of several levels (baseline and each step look-ahead level). Once the composite trees are created for each data routing decision tree, the one with the lowest overall cost is chosen (i.e. the one with the optimal route). Since this refers to a particular orbit, this orbit is then chosen by setting the resultant tree to reflect the next orbit choice. Finally, the decisionImplementor executes the decision and the orbit change is chosen on the next simulation time step. In this example, only one level is listed for the data routing decision trees so the lowest cost is 10 and since this corresponds to Orbit B, it is chosen to be the next orbit.

3.11.2 Verification and Validation

The validation strategy consists of validating the simulation model assumptions, input parameter values and distributions, and the output values and conclusions. These were compared both to specifications of the SRV-1 mobile robotic platform and through use of the MATLAB simulation debugging tools with analytical comparison. Statistical methods were used to confirm compare the simulation model and the measured data.

The sample size was determined based on the method of comparing two alternatives. In this case, a 95% confidence interval was achieved for the results by running each simulation for 25 iterations.

3.12 Summary

This research examined the interactions between mission and network goals within a novel military scenario using both a standard and cognitive DTN topology. A simplistic form of cognition using a multiobjective optimization algorithm was used within the system to balance these two goals and achieve an optimal solution to achieve both objectives. The experimental design was full factorial using two (2) distinct workloads (5 images/min, 10 images/min) and three (3) network configurations (baseline DTN topology, intelligent DTN topology, and a DTN topology with cognition) thus resulting in the need to perform six (6) experiments to test all possible combinations. However, to achieve the 95% confidence interval 25 iterations of each experiment was required thus requiring 150 total experiments. The average loiter time and average end-to-end delay were the metrics used to evaluate the overall performance of each experiment/simulation.

IV. Results and Analysis

4.1 Introduction

This chapter presents and analyzes the experimental results. First, the methods used to validate the architecture models are discussed. Next, the results of each individual performance metric are presented. Finally, an overall analysis of the results is provided.

4.2 Statistical Accuracy

To determine results are an accurate reflection of the sample population the simulation was run until the transient period was encountered and defined. This was estimated to occur when the average end-to-end delay was < 5% of the long term overall average. This occurred at approximately 500 seconds. This can be seen in Figure 42, which shows a single run of the baseline scenario.

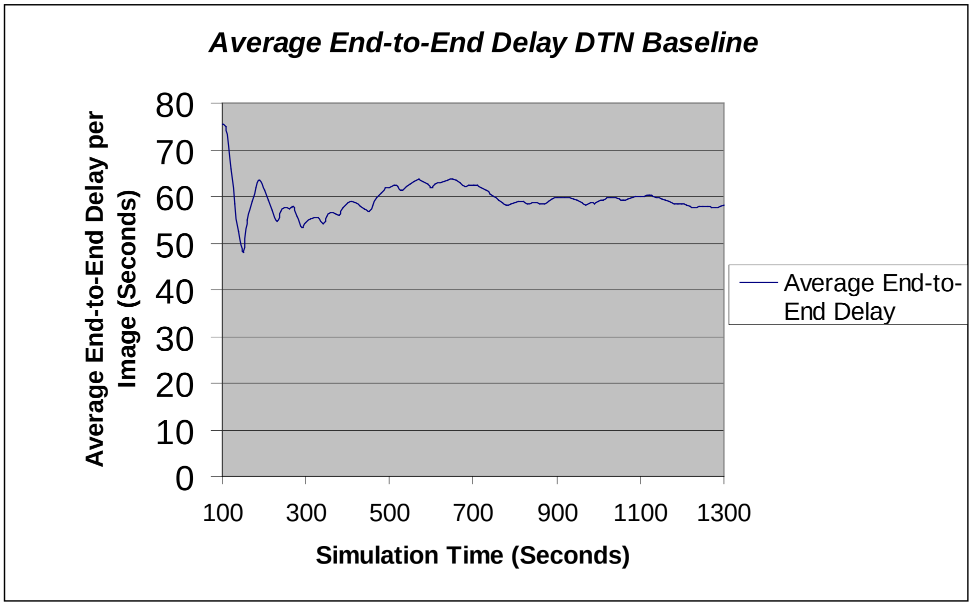

Figure 42. Simulation Transient Period Estimation

This was done in an effort to determine a simulation run time that was long enough for the system to reach a steady state, in order to prevent transients skewing the data.

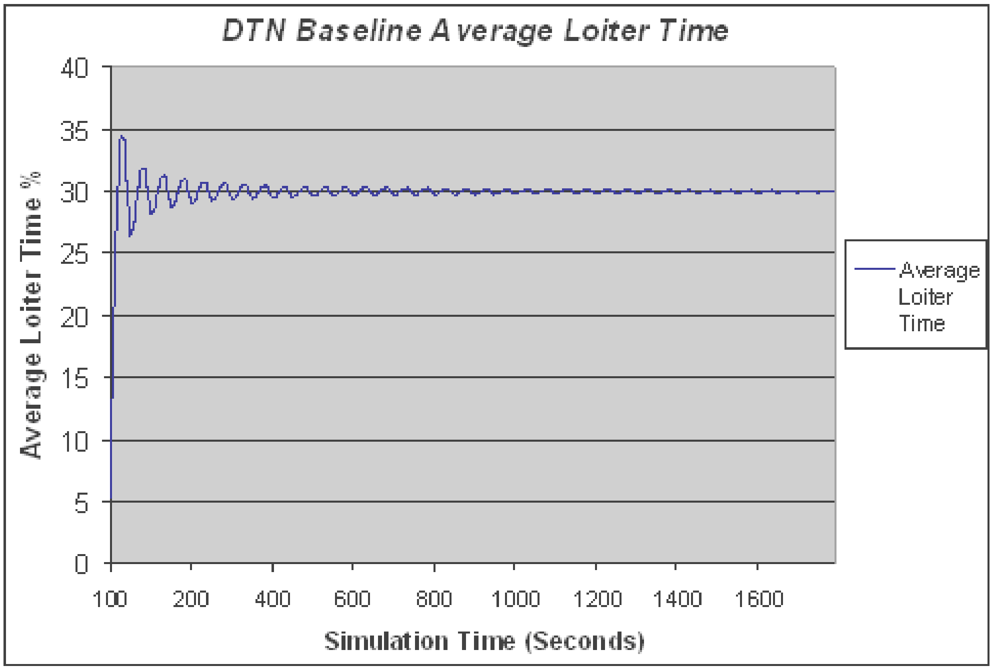

Figure 43. Average Overall Loiter Time

The average loiter time also converged after a period of time within the simulation. This appeared to be based upon the periodic nature of the orbits and contacts that were present within the baseline DTN approach. Figure 43 shows the average loiter time plot for both the baseline and look-ahead approaches since no orbit changes were introduced. The average loiter time reached steady state when the average percent loiter time was < 3% of the overall long term average. This occurred in approximately 500 seconds maintaining a level of approximately 30 percent for the remainder of the simulation.

4.3 Results and Analysis of Performance Metrics

This section interprets and analyzes the relevant data collected from the simulations. As a preliminary step, some initial analysis was done to get a feel for how the DTN network would operate under the given scenario. First, analysis was conducted on how the network traffic metric would be affected (end-to-end delay incurred) if the intermediary node on orbit 2 was used only as a relay (100% communication availability to HQ) versus used as a data ferry with a restricted communication range to HQ (63% uptime on the communication link to HQ). As a data ferry the communication would only be allowed for 57 anchor points of the available 90 on orbit 2. This is shown in Figure 44.

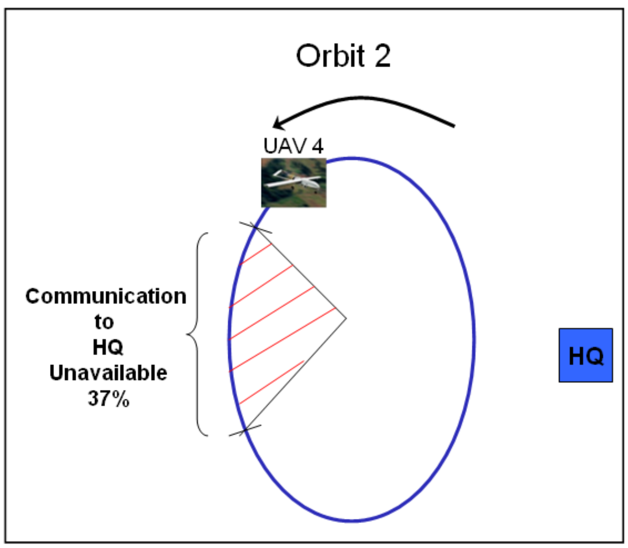

Figure 44. Communication Range to HQ Restricted to 63% Available

The results of this analysis are shown in Figure 45 and Table 3 below. As one would expect the delay was substantially longer for the data ferry. Therefore, this typical performance characteristic of a DTN architecture held true.

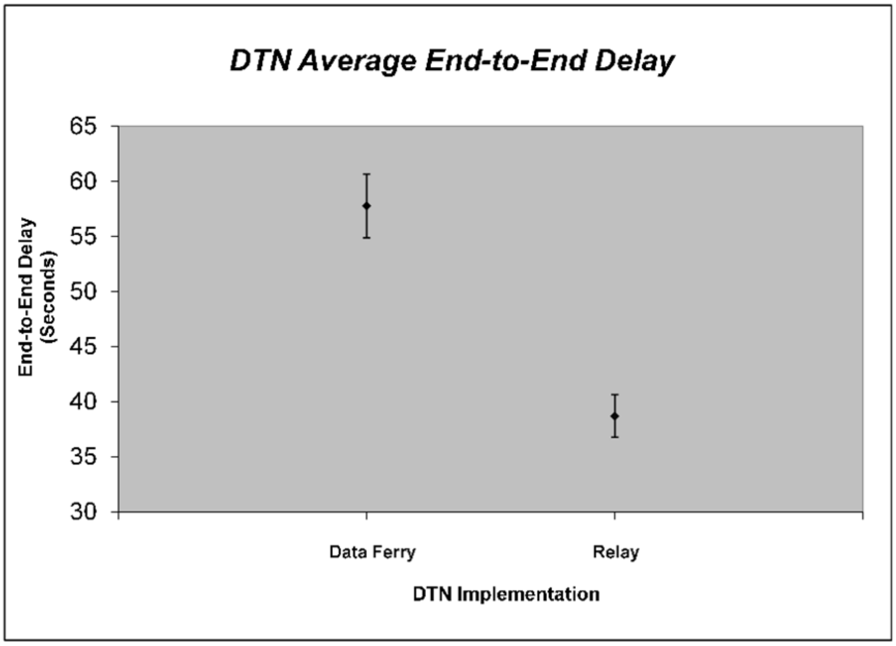

Figure 45. Intermediary Hop as Relay versus a Data Ferry

Table 3. End-to-End Delay comparison between a relay and a Data Ferry

Configuration	***Mean***	***Standard Deviation***	***95% Confidence Interval***	
			from	***to***
UAV as Data Ferry	57.76265	2.420495142	56.8138348	58.71146814
UAV as Relay	38.69571	1.723287953	38.0201913	39.37122423

*Values listed are for the end-to-end delay mean and confidence interval in seconds

Next, the impact of communication link availability to the HQ from the intermediary node (UAV4) was assessed. It was shown that the less time the communication link was accessible, more end-to-end delay was incurred due to the mobility required to ferry the information within communication range of the HQ

.

Table 4. Impact Assessment of Intermediary Communication Link Availability on the average end-to-end delay

Percent Communication	HQ Position	Mean	Standard Deviation	95% Confidence Interval	
				from	to
100%	X=55;Y=0	38.87955105	1.317813627	38.06277671	39.69632539
73%	X=61;Y=0	54.67759199	2.748700842	52.97396086	56.38122312
63%	X=62;Y=0	57.76265001	2.420495142	56.8138348	58.71146814
53%	X=63;Y=0	63.61902066	1.683491848	62.57560078	64.66244054

*Values listed are for the end-to-end delay mean and confidence interval in seconds

For testing purposes, the communication region from the data ferry to HQ was defined as 63 percent available. This value was chosen based upon the data from the impact assessment in Table 4. Figure 46 displays the associated communication patterns for this setting.

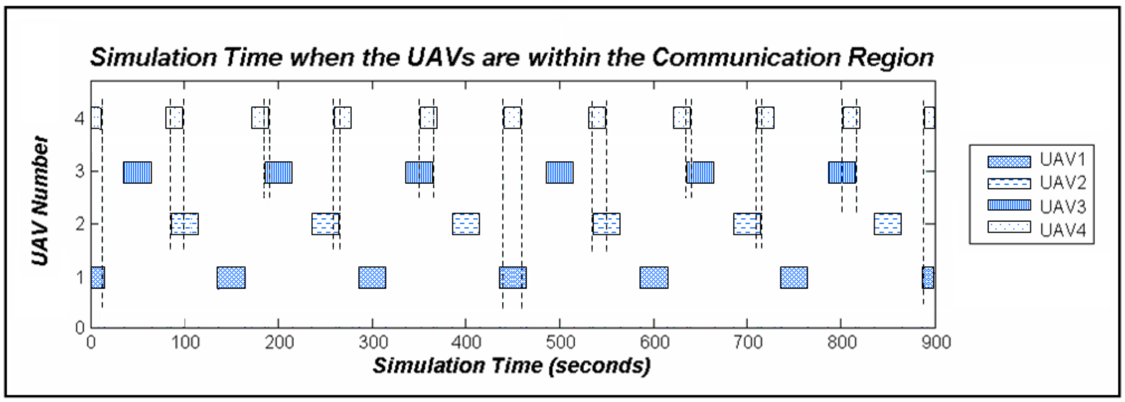

Figure 46. Orbit to Orbit Communication Patterns

Figure 46 highlights the cyclic nature of the availability of the orbit to orbit communication links. This is the time when the UAVs are within the communication ranges to either orbit 2 (UAV1 – UAV3) or orbit 1 (UAV4). Note that when the communication regions overlap with the UAVs in orbit 1 (UAV1 – UAV3) and UAV4 in orbit 2 (identified with dashed lines) a communication link is established and the UAVs are allowed to send data. An example of this is when the simulation time is 86.1 seconds. The dashed line represents a communication link exists for UAV2. This means that UAV2 and UAV4 can communicate from simulation time 86.1 to 99 seconds (129 time steps).

The simulation time period chosen was set to 1,800 seconds and the distance between consecutive anchor points within the same orbit was 1 second. This was chosen because it enabled twelve complete orbit passes on orbit 1 which provided time for the system to reach its steady state and sufficient time to characterize the system behavior. This in turn resulted in 18,000 simulation time steps (ten time steps between anchor points with each 100 msec) in which a potential image could be detected and routed to HQ.

The workloads tested were 1 and 2 percent of the total possible image opportunities which was generated as described in Section 3.9. This resulted in approximately 150 or 300 images (depending on workload) being generated for target observations on orbit 1 within the simulation time period. Also, each simulation was conducted for 25 iterations to ensure statistical accuracy within a 95 percent confidence interval.

The baseline DTN approach was designed to test the performance of achieving the single objective of routing images to HQ. The goal of the objective was to send the images as fast as possible using a greedy approach with the decision tree scheme that was previously mentioned. This configuration was tested using the above settings and in turn resulted in an overall average loiter time of thirty seconds and an overall average end-to-end delay of 57.76 and 58.46 seconds for the corresponding workloads. Since this approach didn't introduce orbit changes, the UAV paths were repeated and the communication region passes were periodic and as a result the data throughput traffic and loiter time was consistent. This approach characterized the system for comparison to the other two approaches under study.

Table 5. DTN Baseline Approach Results

DTN Configuration	Mean	Standard Deviation	95% Confidence Interval	
			from	to
Workload = 1%				
Average End-to-End Delay				
Baseline	57.76265148	2.420495142	56.81383482	58.71146814
Average Loiter Time				
Baseline	30	-	-	-
Workload = 2%				
Average End-to-End Delay				
Baseline	58.46258476	2.113364786	57.63416099	59.29100854
Average Loiter Time				
Baseline	30	-	-	-

*Values the end-to-end delay are listed are in seconds and loiter time are listed as %

Next, the look-ahead approach was incorporated within the baseline DTN approach. This approach introduced extra computation in an effort to enhance the data routing decision. Similar to the baseline DTN methodology, this approach also pursued the single objective of routing data as fast as possible to HQ. This was tested with the same simulation parameters (comm. region 63%, T=1,800 seconds) and in the same way as the baseline DTN approach. The *steps_look_ahead* parameter was set to two (with step size one being the current decision), and the *step_size* parameter was set to five and ten anchor points for testing purposes. The *steps_look_ahead* of two, refers to calculating the current next hop (step 1 based upon the current positions of the UAVs) in the data path and the future next hop (step 2) after the current next hop. The *step_size* is used to determine the number of anchor points from the current position, the future next hop will be located. This was described in Figure 26 with all routes determined and the decision made and the image transmitted to the current next hop within the current 100 msec time window. These results were statistically identical because the means were within each others confidence interval. Also, since there were no orbit changes the average overall loiter time was the same as the baseline DTN approach.

Table 6. Comparison between the DTN baseline and look-ahead approaches

DTN Configuration	*Granularity*	*Mean*	*Standard Deviation*	*95% Confidence Interval*	
				from	*to*
Workload = 1%					
Baseline	N/A	57.76265148	2.420495142	56.81383482	58.71146814
2 Step Look-ahead	5	58.24713361	2.234789349	57.37111228	59.12315494
2 Step Look-ahead	10	58.54292603	2.098855718	57.7201897	59.36566235
Workload = 2%					
Baseline	N/A	58.46258476	2.113364786	57.63416099	59.29100854
2 Step Look-ahead	5	58.63527871	1.738480491	57.95380688	59.31675054
2 Step Look-ahead	10	59.22278205	1.966741657	58.45183348	59.99373061

*Values listed are for the end-to-end delay mean and confidence interval in seconds

In fact, the results for the 10 *step_size* granularity were slightly worse. This was likely due to either the fact that some optimal routes were overlooked by taking such a long look (large amount of anchor points) into the future state or that not long enough look was taken. This was somewhat anticipated due to the repeatability of the UAV behavior in the baseline and look-ahead approaches.

The MOO approach was built upon the look-ahead approach with the addition of the subsumption architecture. This construct enabled the pursuit of multiple objectives simultaneously in an effort to improve both. The loiter time threshold parameter was the primary factor that directly influenced the ability to improve system performance.

Within the MOO approach the communication time windows were significantly affected since UAVs now had the opportunity to take other optional orbits thereby impacting the predictability of the communication links. Figures 47 and 48 illustrate the impact of a particular threshold value (in this case 36) on orbit selection and the resultant communication links.

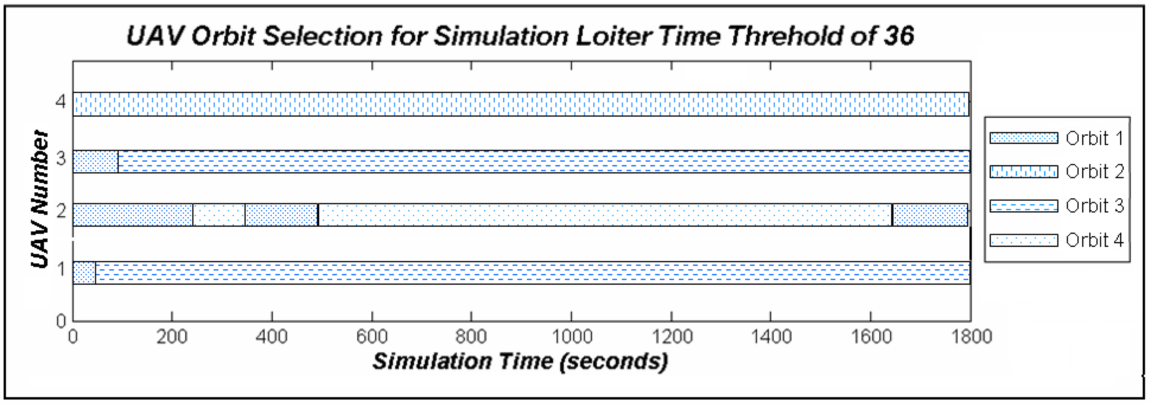

Figure 47. Depiction of UAV Orbit Selection during the Simulation Period

Figure 47 shows the orbit selection for a sample run of the MOO scenario. In this case, UAV1 was on orbit 1 for 44.1 seconds then switched to orbit 3 for the rest of the simulation period. UAV2 was on orbit 1 for 244.1 seconds, then switched to orbit 4 where it remained until simulation time of 344.3 seconds, then switched to orbit 1 until simulation time of 494.3 seconds, then switched to orbit 4 until a simulation time of 794.8 seconds, then switched to orbit 1 until 944.9 seconds and then finally on orbit 4 for the remainder of the simulation period. UAV3 was on orbit 1 for 94.1 seconds and then on orbit 3 for the remainder of the simulation period.

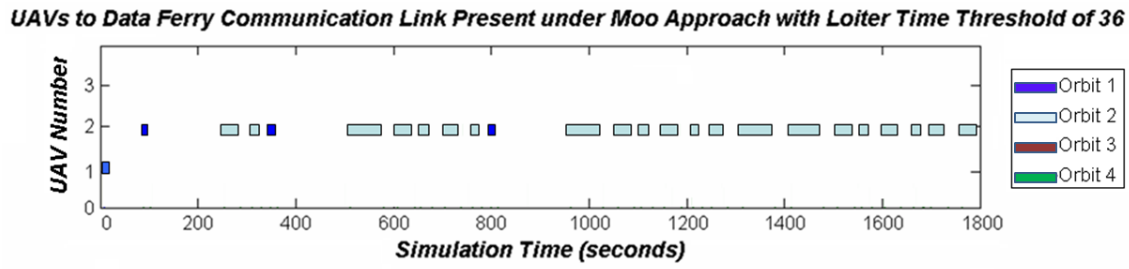

Figure 48. UAVs to Data Ferry Communication Links under the Moo approach with loiter time threshold of 36

Figure 48 shows the time windows for a single run of the MOO scenario in which the UAVs could communicate with the data ferry in orbit 2. For the given loiter time threshold of 36, this figure indicates which communication links exist with a particular UAV and on which orbit the UAV occupies. From the figure it is clear that UAV1 can only send data within the first 15 seconds of the simulation time period, this is due to the fact it switches to and stays on orbit 3 and thus cannot communicate directly with the data ferry. UAV2 switches between orbits 1 and 4 and has a varied communication pattern. UAV3 is never in the communication region when the data ferry (UAV4) is and therefore, never has a communication link. As is evident in this figure, UAV2 relays almost every image to the data ferry. As such, the delay is smaller than the baseline configuration because there is an increased amount of communication link opportunities with the data ferry (UAV4). This is because orbit 4 was designed for this purpose and therefore if it is used, images are sent in less time than in the baseline DTN configuration. Figure 49 illustrates this point.

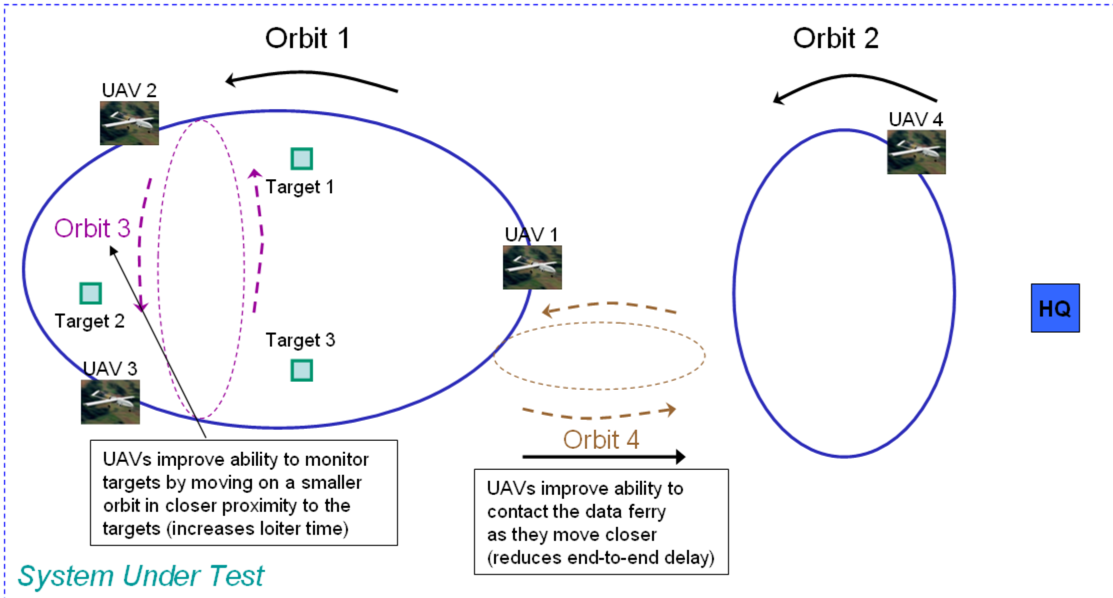

Figure 49. DTN Topology showing how optional orbits can impact the loiter time and end-to-end delay

Since the loiter time threshold value directly impacted system performance, testing of the MOO algorithm was undertaken as an iterative process in which this value was systematically increased with each subsequent trial. The testing started with an initial threshold value of 30 seconds and progressed until the baseline DTN average end-to-end value of 57.76 seconds was exceeded. The results of this process are listed within Table 7.

Table 7. Results from the MOO Algorithm with a given Loiter Time Threshold value

Threshold Value	*Loiter Time (percentage)*	*End-to-end delay (seconds)*
32	35.94829342	27.13952012
34	34.93740393	27.16110315
36	36.74278201	28.43601832
38	38.56695742	44.8979542
40	39.93671125	81.9505002

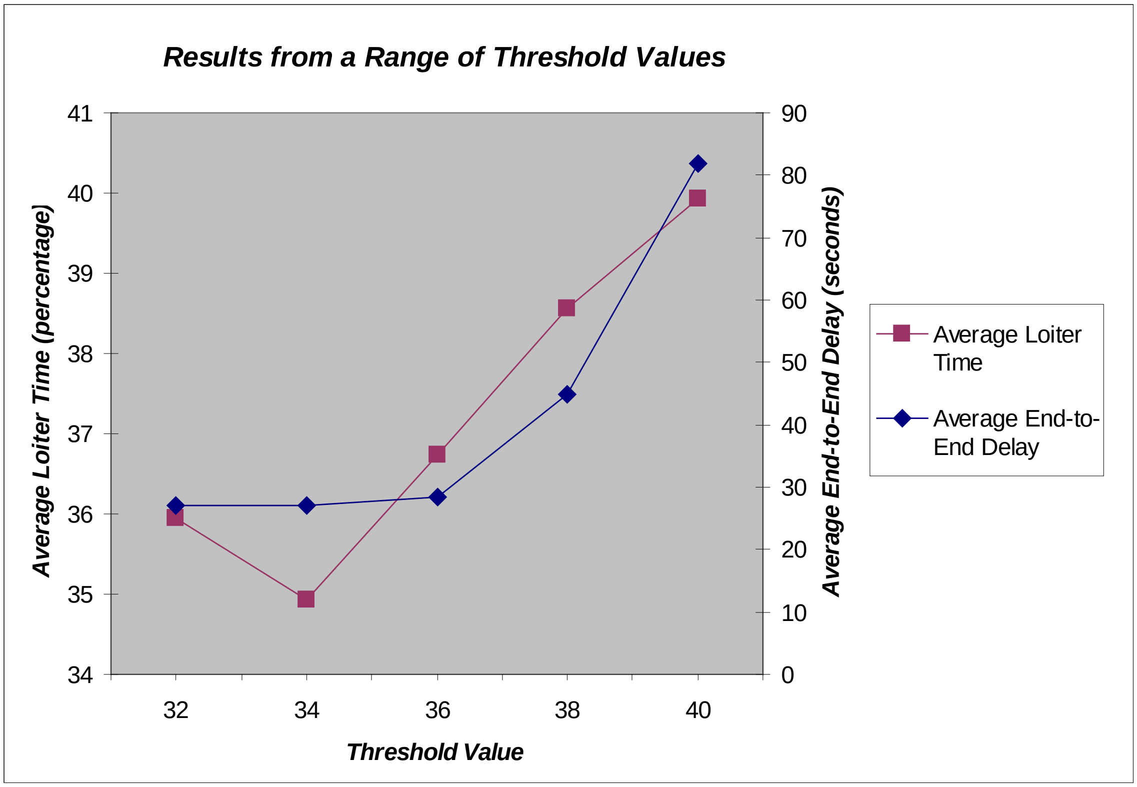

Figure 50. Initial Test to Determine Impact for a Range of Threshold Values

Note that this was merely done to get a quick snapshot as to which would perform better. Each threshold value was run for 5 iterations with the workload set at 1 percent. This was done to identify which threshold value would be run for 25 iterations for comparison with the baseline and look-ahead approach values. As is evident from Table 7, the threshold value of 36 performed well in regards to achieving both the network goal of decreasing the end-to-end delay and the mission goal of increasing loiter time. In this case, this resulted in a 51 percent decrease in end-to-end delay and 6.7 percent increase in loiter time when compared to the baseline approach.

Tables 8 and 9 list the final comparison of the average end-to-end delay and loiter time between the baseline, look-ahead and pseudo-cognitive MOO approaches. As is evident, the MOO approach performed vastly better than the other approaches. This was mainly due to the autonomous decisions of the UAVs to obtain the additional image opportunities via orbit 3 and the additional communication links that were provided by orbit 4.

Table 8. Comparison of the baseline, look-ahead, and pseudo-cognitive Moo approach Average End-to-End Delay Data

DTN Configuration	Mean	Standard Deviation	95% Confidence Interval	
			from	to
Workload = 1%				
Baseline	57.76265148	2.420495142	56.81383482	58.71146814
2 Step Look-ahead	58.24713361	2.234789349	57.37111228	59.12315494
Pseudo-cognitive MOO	28.43601832	2.623676134	27.40755617	29.46448046
Workload = 2%				
Baseline	58.46258476	2.113364786	57.63416099	59.29100854
2 Step Look-ahead	58.63527871	1.738480491	57.95380688	59.31675054
Pseudo-cognitive MOO	30.42110924	1.998074361	29.63787848	31.20433999

*Values listed are for the end-to-end delay mean and confidence interval in seconds

Table 9. Comparison of the baseline, look-ahead, and pseudo-cognitive Moo approach Average Loiter Time

DTN Configuration	Mean
Workload = 1%	
Baseline	30
Look-ahead	30
Pseudo-cognitive MOO	35.9074
Workload = 2%	
Baseline	30
Look-ahead	30
Pseudo-cognitive MOO	36.7428

*Values listed are the percentage time over target

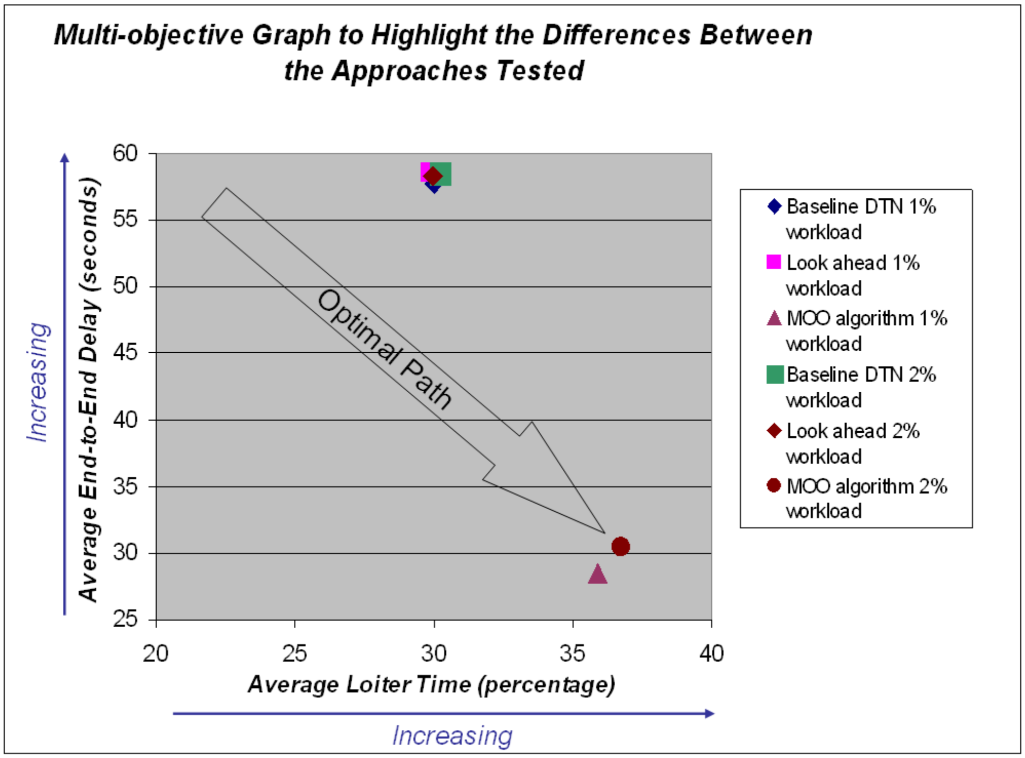

Figure 51. This Multi-Objective Optimization illustrates the results based upon the Loiter Time threshold chosen.

Figure 51 displays the results within a typical MOO graph highlighting the optimal path. This figure shows that the MOO algorithm results are near optimal and therefore are the best solution of the options tested. This in turn emphasizes the approximate 48.6 and 52 percent reduction in the end-to-end delay and an increase of 6 and 6.75 percent in the overall loiter time for the workloads of 1 and 2 percent that were tested.

4.4 Overall Analysis

Several conclusions can be drawn from the simulations conducted. The transitive regions were considered during analysis which ensured data integrity and a credible analysis. It was determined that the use of a data ferry induced delay within the overall end-to-end metric. This is caused by the queuing delay incurred by the mobility required to move within communication range to establish a link. Also, it was concluded that the look-ahead approach alone did not improve the overall system end-to-end delay. All data analyzed was assessed based upon its 95% confidence interval with no overlapping regions to ensure validity. Lastly, the pseudo-cognitive subsumption approach greatly improved system performance.

Additionally, as a result of this process, several trends were identified and the system was characterized. First, as the average percent loiter time threshold was increased the UAVs in orbit 1 had to change to orbit 3 in order to achieve the desired threshold level. Recall, that the purpose for this orbit was specifically to increase average percent loiter time. This resulted in an increased time over target which in turn increased the amount of images taken. This is evident in Table 9 as the values listed are larger than the baseline value of 16,200 opportunities. However, as an effort to offset this increase, orbit 4 was designed as to improve end-to-end delay by providing more opportunity to establish a communication link with the data ferry.

If orbit 4 was not chosen by one of the UAVs the end-to-end delay increased dramatically. Also, if the threshold level was too large the UAVs spent all of their time on orbit 3 trying to achieve the higher threshold level. This resulted in extremely large amounts of traffic between the UAVs in orbit 3 with messages transferring back and forth within the same orbit since these were the only communication paths within communication range.

4.5 Summary

This chapter presented and analyzed the data collected from the simulations of the different approaches related to this novel military application. First, validation of the overall DTN implementation was made with some common sense analytical checks. Next, validation of the three models was conducted with the hypothesis amongst them affirmed. As the testing confirmed, the pseudo-cognitive subsumption approach improved the mission goal by increasing the average percent loiter time between 6.00 and 6.75 percent and the network goal of end-to-end delay between 48.6 and 52.0 percent over the baseline DTN and look-ahead approaches for the workloads of 1 and 2 percent respectively. This result was proven statistically within a 95 percent confidence interval.

V. Conclusions and Recommendations

5.1 Introduction

This chapter summarizes the overall conclusions of the research. First the conclusions captured from the experimental results are presented. Next, the significance of this research is presented. Finally recommendations for possible areas of future work are described.

5.2 Conclusions of Research

This thesis demonstrated how aspects of an NTO cold be incorporated within a pseudo-cognitive process used to simultaneously achieve conflicting mission and network goals within a DTN topology. This was done in an effort to account for network performance objectives that often get neglected as a result of the dominance of mission goals within the Air Force planning process and creation of the ATO. This work provided a novel approach in which to incorporate both the mission and network goals within a robust framework based upon behavior based robotic control principles. The UBF controller architecture provided a means to foster the multi-objective optimization subsumption approach in an effort to allow UAVs to make autonomous dynamic decisions to adjust mission and network parameters to improve or balance both mission and network objectives. Further this work provided a novel approach to improve network performance within a DTN topology. Within the simulation, UAVs were able to dynamically adapt to changing network and mission conditions to best achieve pre-defined goals within the ATO and NTO. This work built upon studies within the areas of cognitive networks, multi-objective optimization, and behavior based robotic control principals.

5.3 Significance of Research

This thesis employed and proved novel concepts in which ATO and NTO parameters could be incorporated within a dynamic pseudo-cognitive multi-objective process to simultaneously balance and or improve mission and network goals. This was indicated as the pseudo-cognitive approach achieved a significant improvement over the standard baseline and look-ahead approaches. This resulted in an increase of 6 and 6.75 percent in the average percent loiter time and a reduction of 52 percent in the end-to-end delay over the baseline DTN and look-ahead approaches for the workloads of 1 and 2 percent respectively. These findings and supporting documentation provides the foundation for future efforts within this area of study. If realized, this could greatly enhance Air Force operational effectiveness by providing commanders and or senior leaders the ability to incorporate aspects of an NTO and ATO within autonomous UAV missions. Hopefully this effort will spawn motivation and new areas of research within this topic of study.

5.4 Recommendations for Future Research

Future research within this topic should focus on providing additional complexities to expand the MATLAB simulation to include incorporating additional orbit choices, additional UAVs, include UGVs (ground based nodes), and incorporate a node failure aspect. The orbit choices should be chosen both arbitrarily and deliberately planned to investigate the responsiveness and adaptability of the UAVs within this construct. Additional nodes (UAVs and UGVs) should be added to see if other optimal patterns or solutions exist. This would provide a means in which to determine the scalability and flexibility of this framework. Adding the learning aspect of cognition would provide the architecture a feedback loop in which to base decisions on past actions instead of purely estimating future responses on present circumstance. This approach would provide an optimal solution but may prove difficult given the complexity of managing this amount of sensory information and limited processing and storage constraints. Lastly, this approach should be attempted experimentally through use of the Surveyor SRV-1 robotic platform. This would provide an avenue to test significant factors that currently were not captured with the context of this simulation. This would provide a means to test the pseudo-cognitive approach under real world conditions and allow for validation of its fundamental common sense principles.

5.5 Summary

In summary, this research presented a novel approach to incorporate cognition by way of behavior based robotic control principles (subsumption) to balance mission and network goals within a DTN topology. This provided commanders a framework to automate aspects of both an ATO and NTO while simultaneously improving both goals thus improving resource utilization and mission effectiveness.

Bibliography

1. Stookey, David E., *"A notional battlespace for simulating and testing dynamic wireless networks"*. MS thesis, AFIT/GCS/ENG/IC4-06J. Graduate School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB, OH, June 2006.

2. Compton, Matthew, Kenneth Hopkinson, and Scott Graham, *The Network Tasking Order (NTO)*, in *Military Communications Conference, 2008. MILCOM 2008. IEEE*. 2008. p. 1-7.

3. Samad, Tariq., John Bay, and Datta Godbole, *Network-Centric Systems for Military Operations in Urban Terrain: The Role of UAVs.* Proceedings of the IEEE, 2007. 95(1): p. 92-107.

4. Woolley, Brian., *Unified Behavior Framework for Reactive Robot Control.* MS thesis, AFIT/GCS/ENG/07-11. Graduate School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB, OH, March 2007.

5. *DODI 8000.01 Management of the Department of Defense Information Enterprise*. 2009. [cited 2009 February 13]; Available from: http://www.dtic.mil/whs/directives/corres/pdf/800001p.pdf

6. *AF Joint Vision 2020*. 2000. [cited 2009 February 13]; Available from: https://acc.dau.mil

7. Pecarina, John M., *Creating an Agent Based Framework to Maximize Information Utility*. MS thesis, AFIT/GCS/ENG/08-19. Graduate School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB, OH, March 2008.

8. *The United States Department of Defense Unmanned Systems Roadmap for 2007-2032*. 2007. [cited 2009 February 4]; Available from: http://www.acq.osd.mil/usd/Unmanned%20Systems%20Roadmap.2007-2032.pdf

9. *Army National Guard: Raven's camera a commanders eyes during a domestic emergency*. 2009 [cited 2009 March 4]; Available from: http://www.ngb.army.mil.

10. *Air Force Link Image of the RQ-11 Raven*. 2009 [cited March 4]; Available from: http://www.af.mil/shared/media/photodb/web/070706-f-6751b-204.jpg.

11. Technology, A. *Army Technology Image of SHADOW Unmanned Aerial Vehicle*. 2009 [cited 2009 March 4]; Available from: www.army-technology.com/.../shadow200uav4.html.

12. Frew, Eric W. and Timothy Brown, *Airborne Communication Networks for Small Unmanned Aircraft Systems.* Proceedings of the IEEE, 2008. **96**(12): p. 2008-2027.

13. *Georgia Institute of Technology UAV Research Group*. [cited 2009 February 20]; Available from: http://uav.ae.gatech.edu/.

14. *University of Minnesota UAV Research Group*. [cited 2009 February 20]; Available from: http://www.aem.umn.edu/~uav/.

15. *Harvard University UAV Networking Lab*. [cited 2009 February 1]; Available from: http://www.eecs.harvard.edu/uavlab/.

16. Frew, Eric W., et al., *Establishment and Maintenance of a Delay Tolerant Network through Decentralized Mobility Control*, in Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on. 2006. p. 584-589.

17. Kurose, James K., *Computer Networking: A Top-Down Approach*. 4th ed. 2007: Pearson. 852.

18. Thomas, Ryan W., *EENG 699 Robotics Network Optimization Lecture Slides*. Graduate School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB, OH 2008.

19. Farrell, Siboihan., et al., *When TCP Breaks: Delay- and Disruption- Tolerant Networking.* Internet Computing, IEEE, 2006. **10**(4): p. 72-78.

20. *The Internet Engineering Task Force*. 2009 [cited 2009 January 15]; Available from: http://www.ietf.org/index.html.

21. Fall, Kevin, *Delay Tolerant Networking Research Group*. 2008. [cited 2008 December 15]; Available: http://www.dtnrg.org/wiki

22. Ott, Jorg, Dirk Kutscher, and Christoph Dwertmann. *Integrating DTN and MANET Routing*. in *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks*. New York, NY, USA: ACM.

23. Cerf, Vincent, *RFC 4838 Delay-Tolerant Networking Architecture*. 2007 [cited 2009 February 1]; Available from: http://tools.ietf.org/html/rfc4838.

24. Scott, K. *Bundle Protocol Specification*. 2007 [cited 2009 February 1]; Available from: http://tools.ietf.org/search/rfc5050.

25. *The Internet Research Task Force*. 2009 [cited 2009 February 10]; Available from: http://www.irtf.org/.

26. Mundur, Padma., and Matthew Seligman. *Delay tolerant network routing: Beyond epidemic routing*, in *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*. 2008. p. 550-553.

27. Abdulla, Muhammad and Robert Simon. *The Impact of the Mobility Model on Delay Tolerant Networking Performance Analysis*. in *ANSS '07: Proceedings of the 40th Annual Simulation Symposium*. Washington, DC, USA: IEEE Computer Society.

28. Jones, Evan., Lily Li, and Paul Ward. *Practical Routing in Delay-Tolerant Networks.* Mobile Computing, IEEE Transactions on, 2007. **6**(8): p. 943-959.

29. Lindgren, Anders., Avri Doria, and Olov Schelen. *Probabilistic Routing Protocol for Intermittently Connected Networks*. 2006 [cited 2009 Jan 20]; Available from: http://www.dtnrg.org/docs/specs/draft-lindgren-dtnrg-prophet-02.txt.

30. Balasubramanian, Aruna., Brian Neil Levine, Arun Venkataramani. *DTN Routing as a Resource Allocation Problem.* 2007.

31. Burgess, John and others. *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks*, in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. 2006. p. 1-11.

32. Thomas, Ryan W. *Cognitive networks*, in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, L.A. DaSilva, Editor. 2005. p. 352-360.

33. Coello, Carlos A., *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques*, in *Knowledge and Information Systems*. 1999. p. 269-308.

34. Zitzler, Eckart, Marco Laumanns, and Stefan Bleuler. *A tutorial on Evolutionary Multiobjective Optimization*, in *In Metaheuristics for Multiobjective Optimisation*. 2004, Springer-Verlag. p. 3-38.

35. Goh, Chi-Keong and Kay Chen Tan. *A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization.* Evolutionary Computation, 2009. **13**(1): p. 103-127.

36. Liu, Chun-an and Yuping Wang. *Dynamic Multi-objective Optimization Evolutionary Algorithm*, in *Natural Computation*. 2007. p. 456-459.

37. Zheng, Biojin. *A New Dynamic Multi-objective Optimization Evolutionary Algorithm*, in *Natural Computation*. 2007.

38. Gat, Erann. *On three-layer architectures*. in *Artificial Intelligence and Mobile Robots*. Jet Propulsion Laboratory, California Institute of Technology. 1998.

39. Matariac, Maja J. *Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior.* Journal of Experimental and Theoretical Artificial Intelligence, 1997. **9**.

40. Dutta, Indranil., Agnieszka Bogobowicz, and Jason Gu. *Collective robotics - a survey of control and communication techniques* Intelligent Mechatronics and Automation, 2004. Proceedings. 2004 International Conference 2004.

41. Jones, Chris. and Maja Mataric, *Communication in Multi-Robot Coordination*. Computer Science Dept, University of Southern CA 2004.

42. Rosenblatt, Julio K. *DAMN: A Distributed Architecture for Mobile Navigation*, Carnegie Mellon University, PA 1995

43. Brooks, Rodney. *A Robust Layered Control System for a Mobile Robot.* IEEE Journal of Robotics and Automation, 1986. **2**(1): p. 14-23.

44. Burns, Brendan., Oliver Brock, and Brian Neil Levine. *MORA routing and capacity building in disruption-tolerant networks.* Ad Hoc Networks, 2008. **6**(4): p. 600-620.

45. Matsumoto, Makoto. *Mersanne Twister Algorithm*. 1997 [cited; Available from: http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/ewhat-is-mt.html.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
27-03-2009	Master's Thesis	August 2007-March 2009

4. TITLE AND SUBTITLE

Using a Multiobjective Approach to Balance Mission and Network Goals within a Delay Tolerant Network Topology.

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Larweck, Anthony L., Captain, USAF

5d. PROJECT NUMBER
JON 09-181

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 DSN: 785-3636

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/09-26

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research (AFOSR/NL)
ATTN: Dr. Robert Bonneau
875 N. Randolph St
Ste 325 Rm 3112
Arlington, VA 22203 Com: (703)696-9545 DSN: 426-9545 email:robert.bonneau@afosr.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR/NL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This thesis investigates how to incorporate aspects of an Air Tasking Order (ATO), a Communications Tasking Order (CTO), and a Network Tasking Order (NTO) within a cognitive network framework. This was done in an effort to aid the commander and or network operator by providing automation for battlespace management to improve response time and potential inconsistent problem resolution. In particular, autonomous weapon systems such as unmanned aerial vehicles (UAVs) were the focus of this research This work implemented a simple cognitive process by incorporating aspects of behavior based robotic control principles to solve the multi-objective optimization problem of balancing both network and mission goals. The cognitive process consisted of both a multi-move look ahead component, in which the future outcomes of decisions were estimated, and a subsumption decision making architecture in which these decision-outcome pairs were selected so they co-optimized the dual goals. This was tested within a novel Air force mission scenario consisting of a UAV surveillance mission within a delay tolerant network (DTN) topology. This scenario used a team of small scale UAVs (operating as a team but each running the cognitive process independently) to balance the mission goal of maintaining maximum overall UAV time-on-target and the network goal of minimizing the packet end-to-end delays experienced within the DTN. The testing was accomplished within a MATLAB discrete event simulation. The results indicated that this proposed approach could successfully simultaneously improve both goals as the network goal improved 52% and the mission goal improved by approximately 6%.

15. SUBJECT TERMS
Air Tasking Order, Communications Tasking Order, Network Tasking Order, unmanned aerial vehicles, cognitive network, behavior based robotic control, subsumption, Unified Behavior Framework, and delay tolerant network

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Capt. Ryan W. Thomas, PhD, ENG
REPORT	**ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	**U**	**U**	**UU**	108	(937) 255-6565; email: Ryan.Thomas@afit.edu

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