3-5-2008

Feasibility Study of Encoding Operational Mission Metadata into IPv6 Packet Headers

Timothy R. Policarpio

FEASIBILITY STUDY OF ENCODING OPERATIONAL MISSION METADATA INTO IPV6 PACKET HEADERS

THESIS

Timothy R. Policarpio, Captain, USAF

AFIT/GE/ENG/08-23

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/08-23

FEASIBILITY STUDY OF ENCODING OPERATIONAL MISSION METADATA INTO IPV6 PACKET HEADERS

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

Timothy R. Policarpio, BS

Captain, USAF

March 2008

AFIT/GE/ENG/08-23

FEASIBILITY STUDY OF ENCODING OPERATIONAL MISSION METADATA INTO IPV6 PACKET HEADERS

Timothy R. Policarpio, BS
Captain, USAF

Approved:

_____/signed/_____	5 March 2008
Dr. Robert F. Mills (Chairman)	date
_____/signed/_____	5 March 2008
Major Paul D. Williams, PhD (Member)	date
_____/signed/_____	5 March 2008
Dr. Michael R. Grimaila (Member)	date

AFIT/GE/ENG/08-23

Abstract

The purpose of this research is to determine the feasibility of using the header fields and header extensions of IPv6 packets to encode mission metadata into computer network streams. Specifically, this thesis seeks to answer several research questions addressing the performance of different packet header encoding methods, specifically which method provides the least end-to-end delay of a file transfer over a hypothetical network as well as which method produces the least amount of additional network overhead during its operation in the hypothetical network. The research questions are answered through a comprehensive literature review and with the use of several network performance calculations. Results are analyzed and a final recommendation is given for which method would best meet the stated need. Ultimately, this research highlights a new way of tracking and reporting to military leaders the status of operational missions and tasks should a network outage or degradation occur.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Robert Mills, for his guidance, support, and patience throughout the course of this thesis effort. The insight and experience was certainly appreciated. I would also like to thank the IPv6 Transition Office at the Air Force Communications Agency for their insight and opinions about my research interest. Specifically, I would like to thank 1st Lt Amanda Uyenishi and Mr. James E. Kohliem, Jr., for their responsive and knowledgeable email correspondence for this research.

I am also indebted to my wife for being patient with me during the long and stressful nights of research.

Timothy R. Policarpio

Table of Contents

List of Figures

List of Tables

List of Equations

FEASIBILITY STUDY OF ENCODING OPERATIONAL MISSION METADATA INTO IPV6 PACKET HEADERS

I. Introduction

1.1 Motivation

Since the attacks on the World Trade Center on September 11, 2001, the U.S. military dramatically increased the tempo of their operations. With this increase in operations tempo, the U.S. military witnessed a quick ramp up of military strength at deployed military installations throughout the world. For military forces to act as one coherent and global team, it is very important for military leaders to assign appropriate missions to deployed forces. At forward deployed installations, it is essential for deployed military commanders to receive real-time status of their assigned mission areas. Deployed military commanders rely on the real-time information to make accurate and timely decisions as well as provide military leaders and other deployed military commanders the status of their missions.

One of the U.S. Air Force's (USAF) missions is air and space dominance throughout the world. The USAF accomplishes this mission with technologically advanced command and control centers known as Air Operations Centers (AOC). Deployed military commanders use AOCs to ensure air and space dominance in their

respective area of responsibility (AOR). Knowing the exact status of all mission areas and tasks within the AOC is essential to effective command and control of military forces.

Currently, within the AOCs, there are many methods and processes to provide commanders with the most current mission, task, or asset status. Although these methods are adequate, they are reactionary. Commanders generally do not know that a mission or system is degraded until after the impact is felt. Improving the assessment and reporting process means quickly determining, through an automated process, which communication systems and network resources are degraded, proactively determining which AOC mission systems are affected by the degraded communication system, and providing commanders an assessment of which missions and tasks are affected and to what extent.

The U.S. Air Force has expanded its mission to include cyberspace as a mission area. The mission of the U.S. Air Force is to "deliver sovereign options for the defense of the United States of America and its global interests -- to fly and fight in Air, Space, and Cyberspace." [1] Operations in and through cyberspace that allow communications and coordination between all forces are necessary. Using cyber assets to ensure the military leaders' commands gets to the troops is vital. The U.S. military has become more reliant than ever on computer networks and the "cyber realm" to allow quick and efficient command and control operations as well as intelligence, reconnaissance, and recovery missions.

1.2 Problem Statement

One of the biggest challenges leaders face is correlating changes and outages of the computer networks and communications systems to effects they have on missions and tasks that rely on those systems. This research attempts to explore a proposed method for addressing this challenge. One of the challenges communications personnel have to meet is determining exactly what leaders need to know. There are so many aspects and statistics associated with managing networks, and determining which of those aspects or statistics leaders care about is a daunting task. Communications personnel are well equipped in managing networks with the use of a myriad of network management tools. Unfortunately, however, much of the information produced by these tools is not required or is not well understood by senior leaders.

This research attempts to bring the communications personnel one step closer to providing leaders with meaningful information amidst a sea of numerical reports and superfluous statistics. What leaders want to know is how the network is affecting the missions and the tasks that are currently being performed. Leaders want to know how a network change or outage affects the missions at hand.

For example, when a piece of computer networking equipment fails (say in a command and control system), leaders need to know how that failure affects those using the system as well as the missions that are being affected by the outage. Unfortunately, what is provided to leaders instead, time and time again, are statistics and numbers about what link has gone down and how much network bandwidth is no longer available, which is arguably meaningless, in the commander's eyes. Bridging this information gap is what this research attempts to close in on.

1.3 Research Goals

This research intends to answer several questions. First, this research sets out to determine ways to insert metadata about missions directly into the network traffic. This is done by inserting metadata into the packet headers. Then, the research examines how the encoding affects network latency when information is transmitted across a hypothetical network. Specifically, this research examines how encoding mission data will affect how long it takes the file to reach its destination, otherwise known as the end-to-end latency. The research determines how much additional overhead each of the encoding methods introduces into the network. Overhead is the amount of additional bits introduced into the network as a result of encoding of the metadata and the amount of additional bits required to update a network management system. Finally, the results are analyzed to determine which method of encoding is "best" in terms of lowest latency of a file transfer and smallest overhead. Ultimately, this research attempts to provide a way to help leaders attain the most pertinent situational awareness of the missions and tasks they are performing.

1.4 Limitations, Assumptions, Scope

The research is a feasibility study of using packet headers to store metadata about operational missions. This research does not set out to provide an extensive, in-depth network analysis of all performance aspects of a network. Only a small number of network performance concepts are selected and used. Since there are many aspects that

affect network performance, this research does not set out to specify every aspect and detail of the hypothetical network.

Also, assumptions about propagation, processing and queuing delay within the network nodes are made. Assumptions and simplifications about these types of delay are made in order to simplify the complex operations of computer networks.

Another assumption made is that the hypothetical network is entirely based on IPv6 protocols. There is no use of IPv4 nodes. Including IPv4 nodes introduces another level of complexity in the networks, which in turn, requires more complex experimentation and calculation. Furthermore, IPv4 is a legacy protocol that will be phased out in the not too distant future.

Finally, an essential part of this research is the existence of the notion outlined in past research by Alfred Shaw. The mission database proposed in Shaw's work is assumed to be operational and populated with information about network assets and how they correspond to missions in order for the solutions of the research presented in this paper to be relevant to the stated problem statement.

1.5 Methodology

Since determining the feasibility of encoding mission metadata onto network traffic is the primary goal, this research sets out to determine how to encode mission data into network traffic using the IPv6 header and extension headers. Several methods of encoding the metadata are examined. This research determines which method is most feasible by calculating network performance aspects of the end-to-end latency and additional network overhead produced for each method. The method that has the least

amount of latency for a packet transmittal across the network and the smallest additional network overhead is deemed the best method.

1.6 Preview

Chapter 1 has provided a brief introduction to the research conducted in this thesis. Chapter 2 provides more detail on the background of the aspects and ideas used in this research. Chapter 3 discusses the methods for determining the end-to-end latency and additional network overhead for each one of the mission metadata encoding methods. Chapter 4 provides results and analyses of the calculations as well as advantages and disadvantages for each method. Finally, Chapter 5 concludes this by stating whether or not the objectives of the research have been accomplished, and provides recommendations for possible future research in this area of study.

II. Literature Review

2.1 Background

Although it may be obvious how important cyberspace has become to carry out missions, what is difficult to grasp is the sheer amount of network traffic being produced to support these types of activities. Even more difficult is tracking the different types of information exchanges that support different military missions. Also, understanding how network components support these missions and tasks can quickly become confusing.

Determining how computer networking assets affect a specific information flow that supports a mission process is arguably one of the most difficult challenges faced by network technicians. Likewise, it is also difficult to trace operational tasks and activities to the underlying support infrastructure. Consequently, military commanders demand they have situational awareness of all assets and operations. They require that they know the status of all missions going on under their watch. When the success of missions and tasks relies on the use of a computer network, commanders quickly realize how important the health of the computer network becomes.

The rest of this chapter provides background information about joint doctrine and principles of network centricity, the importance of situational awareness, the role of the AOC, the significance of past research that servers as a prelude to this research, the aspects of the Combat Information Transport system, the importance of IPv6 in the future of U.S. Air Force networks, and how network management systems provide the best situational awareness to military commanders.

2.2 Network Centric Operations, Joint Vision, and Situational Awareness

To improve cyber situational awareness my mapping missions to systems, network centric operations, joint vision, and situational awareness need to be understood by the maintainers and operators. This section briefly describes each aspect and explains why these aspects are important in solving the cyber situational awareness problem.

2.2.1 Network Centric Operations Revisited

As defined earlier, network centric warfare is the emerging combination of strategies that allows the U.S. military to use emerging technologies to fight conflicts in the Information Age [2]. There are four basic tenets of NCW that enhance the capabilities of the U.S. military. These tenets are the following:

1) Information sharing is improved by a robustly networked force
2) The quality of information and shared situational awareness is enhance by information sharing
3) Shared situational awareness enables collaboration and self-synchronization, and enhances sustainability and speed of command
4) These together increase mission effectiveness [2]

These four tenets allow all the U.S. military services to jointly plan and execute operations with a warfighting advantage attained by NCW. By continuously thinking in terms of joint operations, the military services are coming closer to the ideas and goals presented in Joint Vision 2020.

2.2.2 Ideas of Joint Vision 2020

Joint Vision 2020 (JV2020) is a continuation and clarification of concepts presented in Joint Vision 2010 (JV2010). The focus of Joint Vision is full spectrum dominance through each of the military services' use of dominant maneuver, precision engagement, focused logistics, and full dimensional protection [3].

According to [3], full spectrum dominance is defined as "the ability of US forces, operating unilaterally or in combination with multinational and interagency partners, to defeat any adversary and control any situation across the full range of military operations [3]." Dominant maneuver, which allows U.S. military forces to gain a decisive advantage by controlling the breadth, depth, and height of the battle space, is one of the concepts that allows full spectrum dominance to happen [4]. In addition to dominant maneuver is precision engagement, which is the ability of our forces to find a target, engage the target, determine effects on the target, and re-engage the target if necessary [4]. Additionally, the ability to control the battle space to allow our forces to perform their missions is full-dimension protection [4]. Finally, the last concept that allows our forces to enjoy full spectrum dominance is focused logistics, which is the fusion of information, logistics and transportation technologies and operations [4].

What ties all the four concepts of full spectrum dominance and allows the military services to successfully operate in a joint manner is information superiority. Information superiority is defined as the capability to collect, process, and disseminate an uninterrupted flow of information while exploiting or denying an adversary's ability to do the same [3]."

According to [3], when used properly, information superiority can provide the joint force advantages over our adversaries. Becoming experts in these four concepts allows the services of the U.S. military to ready themselves for future conflicts. To become experts in the four concepts, constantly installing and implementing new technologies is required. [3]. However, tracking how the new technologies affect how the military services affect joint operations and the improved situational awareness joint operations provides continues to be difficult.

Providing leaders and decision makers with the most up to date information about the tasks and missions they are performing is important in any organization. Staying true to the concepts presented in JV2010 and JV2020, one of the goals of this study is to find a feasible way to provide leaders with the most current status about the missions and tasks that are currently being worked. In order to provide leaders with the best situational awareness, a thorough understanding of situational awareness is required [5].

2.3 The Role of the AOC

The Air Operations Center (AOC) is the standard U.S. Air Force command center at the operational level of warfare. Deployed throughout the world, AOCs provide commanders a system to control all air operations in a given theater of operations [6]. One of the many roles the AOC has is directing the air assets in a given region of conflict. In order for the AOC to do this, it must first produce an Air Tasking Order (ATO). This ATO defines the schedule for aircraft that are to fly on a given day [6].

In an AOC, producing the ATO is one of the most important missions to be performed. Tracking this mission to completion is a very important task. Commanders

must know the status of the ATO production at any time, and if there is an issue, it must be tracked down and reported to the commander in a timely manner. Much research has been focused to situations like this in which commanders require they have the best situational awareness about a specific mission. Commanders require they know about what is affecting ongoing missions, as quickly as possible.

2.4 Continuation of Past Research

Two research papers provide the foundation for the research presented in this paper. In *Graph Theoretical Analysis of Network Centric Operations Using Multi-layer Models*, Wong-Jiru examines how a layered approach to understanding missions and network assets is essential to providing leaders the most up to date situational awareness in the operational environment [7]. In *A Model For Performing Mission Impact Analysis of Network Outages*, Alfred Shaw presented a frame work for which a database can be designed to map mission tasks to network components [6]. Both of these efforts are discussed briefly in this section.

2.4.1 Layered Approach to Understanding Missions and Network Assets

Wong-Jiru introduced a concept of layering all aspects associated with accomplishing specified tasks, processes, or missions. Wong-Jiru detailed a multi-layer approach to mapping mission assets to mission tasks. The purpose of this multi-layered model is to provide higher level leadership the details about the effects of network changes on mission effectiveness [7].

The main motivation for Wong-Jiru's research is to answer an issue that has surfaced many times in the world of U.S. Air Force Communications – how a change in the network, whether subtle or significant, affects mission effectiveness. Wong-Jiru's research specifically tries to shed light on the issue of determining who is affected and what tasks are disrupted when a communications system or asset fails in a command and control system [7]. Wong-Jiru's multi-layer model (Figure 2.1) for network centric operations allows analysis of mission effectiveness by inter-relating the cause and effect of all networks that contribute to those network centric operations.

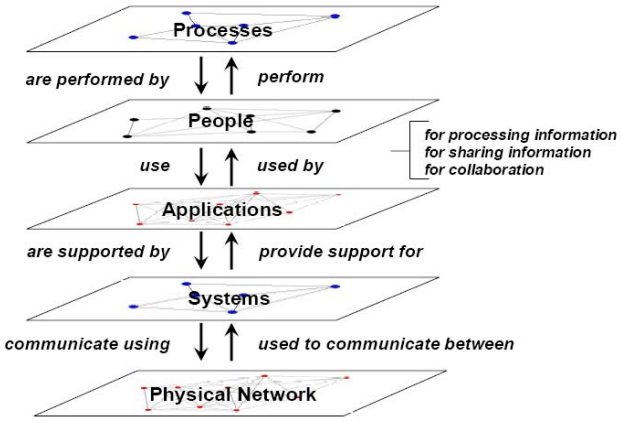

Figure 2.1: Wong-Jiru's Multi-layer Approach [7]

In Wong-Jiru's model, nodes in each layer share similar characteristics to nodes in the same layer. Additionally, nodes directly affect other nodes on the same level. Finally, lower levels directly or indirectly affect higher levels, to some measurable extent. The main idea behind Wong-Jiru's model is to show that any negative or positive occurrences at lower levels provide a corresponding effect at higher levels [7].

More specifically, Wong-Jiru introduces methods as well as diagrams on how to map layers onto other layers. Wong-Jiru specifically shows how to map the people layer with the processes layer. An example of this method is illustrated in Figure 2.2 [7].

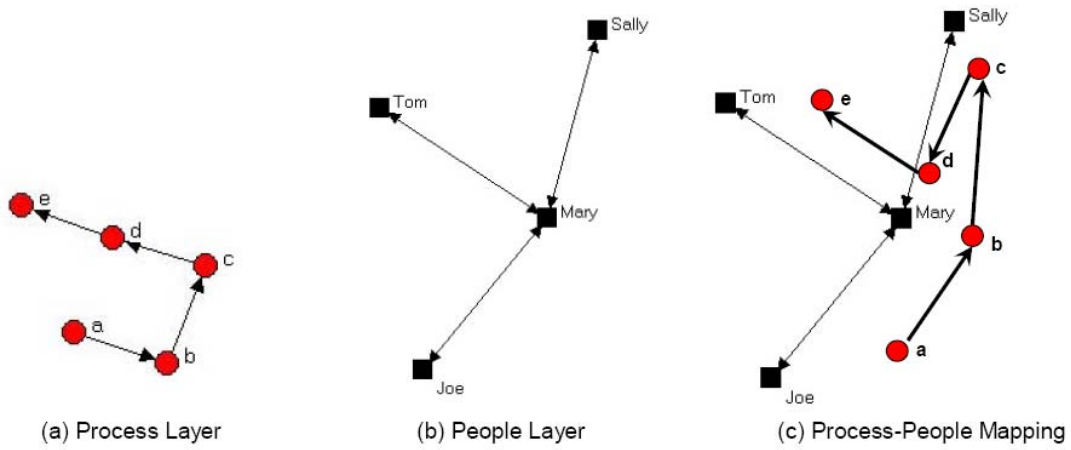

Figure 2.2: Wong-Jiru's Mapping Method [7]

Also shown, the tasks of a process are mapped to the people doing the task. This method can be extended to other layers. Using this method, we can map higher level layer tasks to the lowest level, the physical network layer.

2.4.2 Mapping of Mission Tasks to Network Components

The research accomplished by Shaw applies Wong-Jiru's layering methodology to map specific systems to network assets. The goal of Shaw's research is to propose a model that aids in determining the impact of network outages on missions. Shaw's research specifies three layers and provides a methodology of mapping those three layers to accomplish his research goal. In Figure 2.3, Shaw uses ideas introduced in Wong-Jiru's research and focuses more specifically on how network components affect organizations and missions.

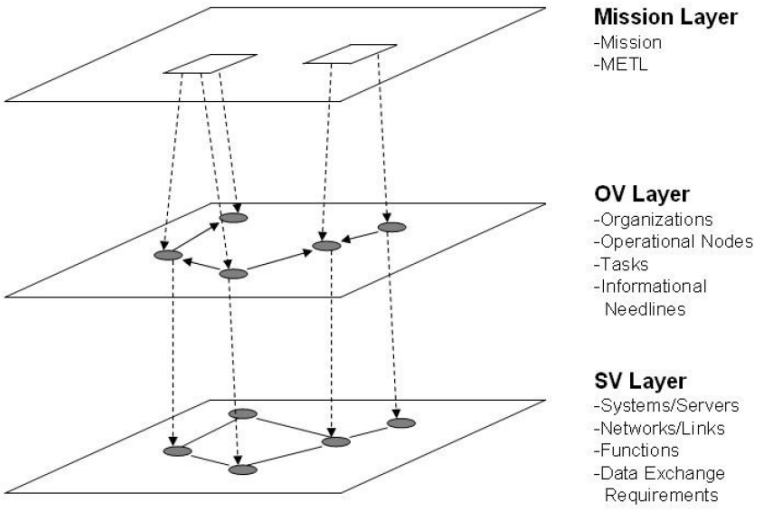

Figure 2.3: Shaw's Three-layer Model [6]

Shaw's three-layer model (Figure 2.3) is a specific application of Wong-Jiru's multi-layer model from Figure 2.1. Moreover, Shaw limited his three layer model to focus on network assets and mission effects [6].

The first layer of Shaw's model shows the missions and mission essential tasks of a given architecture. In the second layer, Shaw specifies the organizations and operational tasks these organizations perform. Shaw shows, in the third layer, the networks and systems and the functions the systems perform. Shaw further explains that within the model, once tasks and missions are identified, we are able to assess the impact of an outage using either a top-down or a bottom-up approach [6].

2.5 Applying Ideas to AOC concepts

Using the concepts presented in Shaw's and Wong-Jiru's research, one mission is selected and examined in this research. Shaw proposed a mission database to identify and map missions to specific organizations. Additionally, missions and organizations are mapped to network assets [6]. Therefore, identifying the network assets that support missions and organizations can be accomplished by tracing the information flows that support those missions or information flows that go from one organization to another. This research continues this concept of tracing the information flows and provides additional examination of how those flows can be monitored as they traverse the network.

This research uses an AOC mission that Shaw has already traced [6]. The mission was dissected to determine the activities, tasks, information flow type, and network assets that are used to support the mission. This research uses the example mission of ATO production [6]. According to Shaw, the mission of ATO production depends on collecting data for the Airlift Import Manager (AIM). The type of data that traverses the network that allows this activity or task to happen is the Air Battle Plan

(ABP) data. Many servers are required to collect the data for the AIM. These servers are the Air Operations Database (AODB) server, Theater Battle Management Core Systems (TBMCS) Airlift Import Manager (AIM) server, Incorporated Research Institutions for Seismology (IRIS) Messaging server, and Command and Control Information Processing System (C2IPS) server [6].

To illustrate how the information flows from one server to another, consider the diagram in Figure 2.4 in which servers are placed in different locations. At location 1, the C2IPS server resides. At location 2, the AODB server resides. Finally, at location 3, the TBMCS AIM and IRIS Messaging servers reside. Location 1 represents a deployed or downrange base. Location 2 represents a base located in the Continental United States (CONUS). Finally, location 3 represents a base where the main command and control structure is located.

Although the diagram depicts where specific servers are at different locations, it is not a complete and accurate depiction of how the AOC operates. The diagram is merely depicting several servers and arbitrarily chosen locations. In addition to the server is a computer network which the servers rely on. There are a number of network assets in between each one of the servers. These assets consist of network switches and routers.

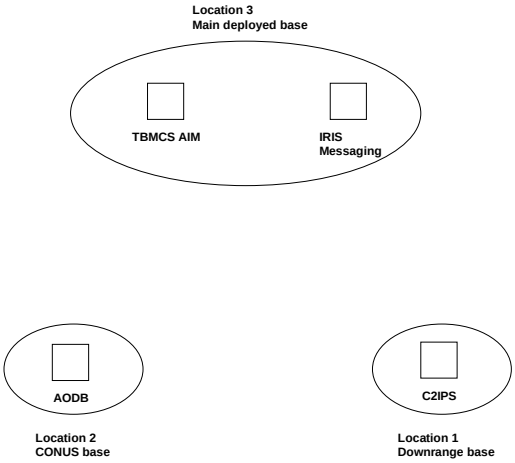

Figure 2.4: Logical Representation of ATO Server Location

Using Shaw's methods, interaction between the servers can be identified. In Figure 2.5, arrows are used to depict how information flows from one server at one location to another server at a different location. The information flows also traverse the local area network of that specific location, as well as the inter-networking components between the different locations, which are not depicted.

There is an information flow between the C2IPS server at location 1 to the IRIS messaging server at location 3. Another information flow occurs between the IRIS messaging server at location 3 and the AODB server at location 2. Finally, the last information flow occurs between the AODB server at location 2 to the TBMCS AIM server at location 3. It is important to note that there are numerous network nodes that lay between the three locations. These network nodes are essential to the transmission of the information flows and are mapped to the specific information flows also. Therefore, the entire path of communications assets that the information traverses can be determined

and examined. Additionally, files and packets that traverse this path can be monitored and traced.

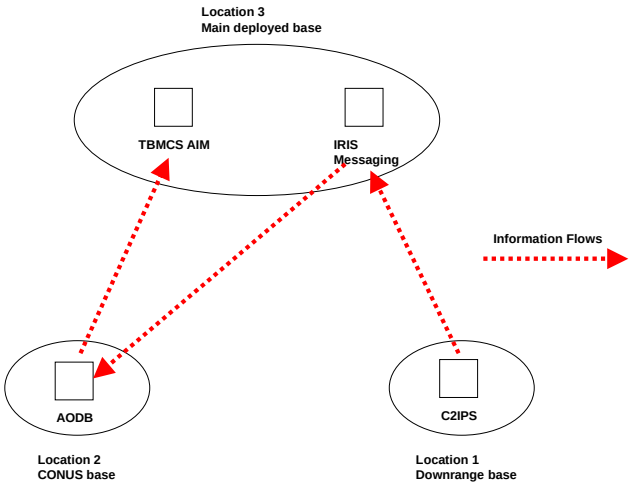

Figure 2.5: Information Flows Supporting ATO Production

2.6. Metadata and the Mission Database

One of the assumptions for this research is that the mission database proposed by Shaw has been built and is operational [6]. However, this research requires more than just having a populated database with information about missions, organizations, and network equipment. It is also required that Shaw's mission-to-asset database interfaces with a separate management system which provides inputs to the database. The database, in return, outputs information about the affected missions in question.

2.6.1 Definition of Metadata

The inputs and outputs of the mission database can contain data about the missions, organizations, or network assets. This data about missions, organizations, or network assets is considered metadata. According to [8], metadata is "structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource." In more simple terms, metadata is "data about data" or "information about information." [8] The reason why the term metadata is used in this research is because it provides a descriptive way to either aid in the discovery of relevant information, to help in the organization of electronic resources, to provide digital identification, or to support the archiving or preservation of data [8].

An example of the use of metadata is in the library cataloging system. The entries in this cataloging system have information about the books that are in the library. For example, if one entry of the cataloging system is examined, it will show some information about the book in question, such as the title of the book, the author of the book, and the book's publishing date. The title, author, and publishing date are considered the metadata of the book. They describe the book but do not necessarily show the contents of the book.

2.6.2. Design of the Mission Database

The mission database contains information about how the missions, organizations, and network assets are correlated to each other. An entry in the database shows that a certain mission can be associated with a certain group of network assets. Also, a network asset is shown as being associated with a certain group of missions. In this database, a

mission code identifies which network assets are associated to which missions and vice versa. Therefore, in this database, the mission code is considered the metadata of the mission.

Depending on the input into the database, a certain output can be produced. If the input is metadata that describes a mission, the output is a list of network assets that support that mission. If the input into the database is the name of a network asset, the output is a list of missions supporting that network device.

The mission database receives inputs from the system that manages the computer network as well as the user interface to the database. In addition to receiving database queries from a user, the mission database interfaces directly into the network management system. A more in depth explanation of the network management system is provided later in this chapter. However, it is important to show that the mission database is considered part of this system because the inputs and outputs of the database come directly from the network management system.

2.7 Combat Information Transport System Block 30

Completely specifying the computer network between the different locations is out of the scope of this research. However, what needs to be recognized is that the methods used in this research can be applied to how future computer networks will be set up for the U.S. Air Force. Combat Information Transport System (CITS) Block 30 specifies how U.S. Air Force networks will be designed and operated [9]. CITS Block 30 specifies the interfaces between the U.S. Air Force networks and the Global Information Grid (GIG), the interfaces between U.S. Air Force networks and sister service networks,

and the interfaces between garrison and deployed bases via the U.S. Air Force Intranet [9].

In Figure 2.6, the previous examined AOC mission example is applied to a CITS Block 30 architecture. Location 1 is the downrange, deployed base in this diagram. Additionally, location 2 and 3 are the garrison, CONUS base and main deployed base, respectively. In between the bases lies the network architecture that comprises the Air Force Intranet, while the I-NOSC provides overall management and security for all network components within the Air Force network.

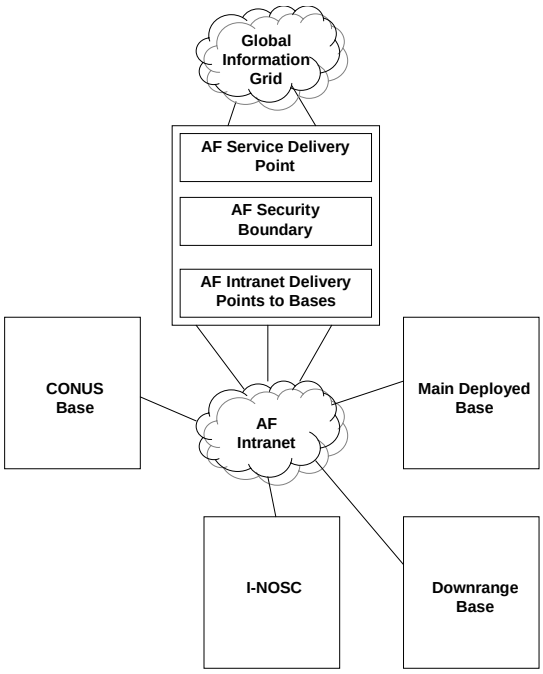

Figure 2.6: Logical Diagram CITS Block 30 Architecture [9]

When data is transmitted from one location to another location, that traffic first traverses its local base network. The local network is comprised of many different types of computer nodes and assets, to include servers, switches and routers. Figure 2.7 presents a base network according to the CITS Block 30 specification. In our AOC example, the AOC server resides in the network behind the Information Transfer Node (ITN) router and switches. The information flow travels from the server in the ITN portion of the network, through the ITN router, through the load balancer switch, past the Virtual Private Network (VPN) concentrator, through the base router, and finally out of the base, into the Air Force Intranet [9].

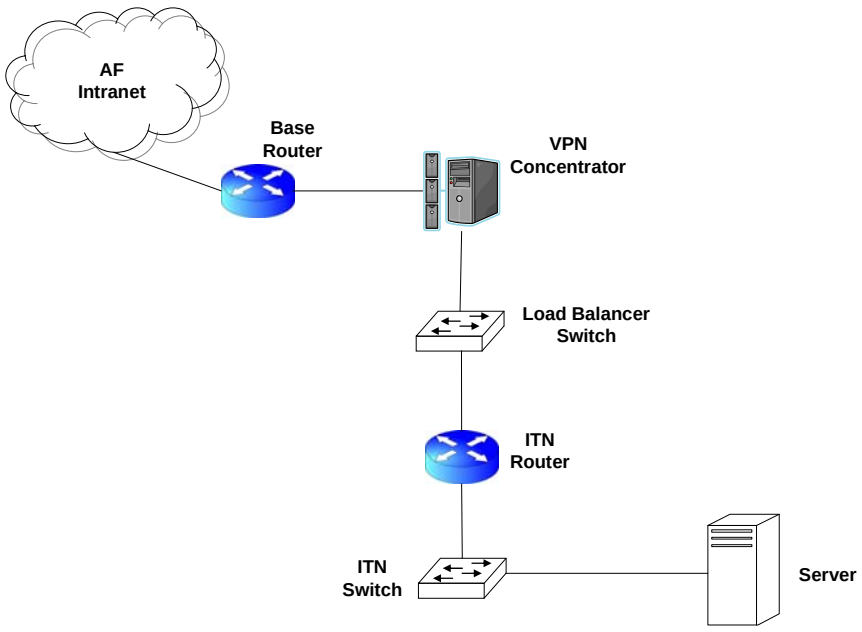

Figure 2.7: Base Network Diagram According to CITS Block 30 Specifications [9]

Once the information flow leaves the base, it traverses the AF intranet and out through the GIG to other bases. In the AOC example, the information stream goes from the C2IPS server at the downrange deployed base at location 1 to the IRIS messaging server at location 3, which is the main deployed base. Then, the IRIS messaging server sends the information flow to the AODB server at location 2, which is the CONUS base. Finally, the AODB server sends the information flow to the TBMCS AIM server at the main deployed base. This information flow is depicted in Figure 2.8.

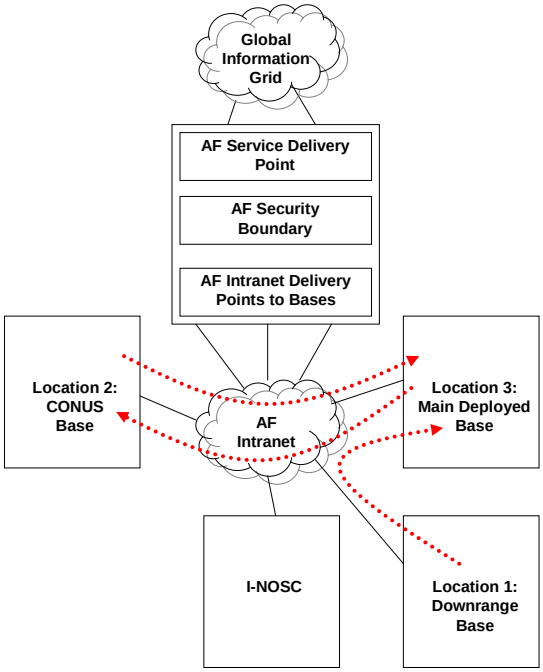

Figure 2.8: CITS Block 30 Architecture with Information Flows [9]

Since the purpose of this research is not to completely specify the entire network, or examine different routing protocols or techniques, generic network nodes and generic network links are used to simplify the analysis. However, applying the concepts to any

network would be possible, if all aspects of the network's components are specified. The generic network diagrams analyzed in this research are based on the CITS Block 30 base network configurations. The composition of the different networks to be examined in this study is discussed in detail in the Chapter 3 Methodology.

2.8 Bridging the Gap Between Mission and Network – The Mail System Analogy

In the communications field, understanding how changes in the computer networks and communication systems, on which missions are relying, is one of the communications career field's biggest challenges. Even more difficult is determining which missions and tasks are affected as the network changes or degrades. As mentioned before, one of the hardest challenges for personnel in the communications career field is bridging the gap between knowing the extent of a network outage, determining how it affects ongoing missions, and reporting this knowledge to senior leaders.

How is this gap bridged? One possible solution is the method proposed in this research - the novel use of inserting mission data into the network traffic streams. This is accomplished by encoding mission metadata into the network transmissions themselves, specifically encoding the mission metadata directly onto network packets.

To further illustrate this line of thinking, the U.S. postal system is used as an example. If a high ranking official wants to send correspondence to another official in another part of the country, the official writes a message, seals it in an envelope, puts the return and destination addresses on the outside of the envelope, pastes the proper postage on the envelope, and drops it off in a mailbox for delivery. To further illustrate, the message is in direct support of an ongoing tasking or mission that affects both the sender

and receiver. In order for anyone, other than the sender or receiver, to know that the message in the envelope is important and supports a certain important tasking, information about the tasking can either be put inside the envelope (the payload) or on the outside of the envelope (the header).

2.8.1 Payload Approach

One way to put mission data onto the network transmissions is to put the mission code information directly into the packet's message body or payload. In this approach, the mission data information is appended at the end of the packet's payload. Using the mail analogy, this approach is akin to inserting a sentence or two at the end of the important official letter or message.

There are some advantages and disadvantages with this approach. At a cursory glance, this approach seems to be the easiest to implement. The receiver, sender, or anyone else who handles the message can simply determine the mission the message supports by reading the contents or payload. However, this action is also a con for this approach. Everyone who handles the message must open the packet to read the purpose of the message. This simply is not feasible, since the sender and the receiver may not intend for everyone who handles the message to open the envelope to determine its purpose. Similarly, in a network, the nodes in between the source and the destination would have to read the packet's payload to determine the mission it supports. This is not the job of each network node to do. The network nodes simply pass along the packets until they get to their final destination.

2.8.2 Header Approach

A more reasonable approach is to find a way to embed mission data on the outside of the message body or onto the packet header. In this way, the packet header can be examined by all nodes along that packet's routing path. Referring back to the mail analogy, this approach is like printing a mission code on the outside of the envelope. In this way, those who handle the message can determine what mission it supports and how important the message is by reading the mission code on the outside of the envelope, without having to open the envelope at all.

Although there may be some processing time associated with reading the mission code in the packet header (or on the outside of the envelope), a significant amount of time is saved because the contents of the message do not have to be examined in order to determine what mission is being supported. This is the approach this research examines - embedding mission metadata into network streams by inserting the code into the network's packet headers.

In a real world application, having someone simply read the mission code on the outside of an envelope and knowing the mission that it supports is not a very secure approach. Therefore, instead of printing what mission is being supported, a coded number or message can printed instead. Only those who have access to what the code means would know what mission is being supported. This is where a secured database, like the database specified earlier by Shaw, becomes important. The database tracks what all the different codes mean and what missions are associated with each of the codes.

2.9 IPv6 Packet Header Traits

Now that an approach to embedding mission data into the network and a method of keeping track of all the mission codes have been determined, a method of inserting the mission code into the network data streams needs to be examined. In a network packet, the source and destination information resides in the Internet Protocol (IP) layer or portion of the packet. This IP portion is located in the header of the packet. In this research, the IPv6 protocol and corresponding packet header are examined to determine locations where mission code information can be stored.

2.9.1 IPv6 Background

IPv6 is a computer networking protocol that allows computers to communicate with each other. IPv6 is the replacement of IPv4 and provides many new attributes. One of these attributes is the more robust header and header extensions. The changes from IPv4 to IPv6 fall primarily into five categories: 1) Expanded Addressing Capabilities, 2) Header Format Simplification 3) Improved Support for Extensions and Options, 4) Flow Labeling Capability, and 5) Authentication and Privacy Capabilities [10]. This research examines how three of the five categories can help solve the problem statement. The three in question are the header format simplification, improved support for extensions and options, and the flow labeling capability [10].

2.9.2 Header Format

As discussed in [10], simplifying the header of the IPv6 network packet allows for easier networking between nodes and easier packet handling. Therefore, some of the

fields in IPv4 have been dropped and are not used in IPv6. This is called Header Format Simplification, and it allows for easier network node packet handling [10].

In Figure 2.9, the format of the IPv6 header is depicted. The main portion of the header is 40 bytes large. This main portion does not fluctuate in size. The fields as well as the size of the fields do not change either. The "version" is a 4-bit Internet Protocol version number which is 6 for IPv6. The "DS" field or "Traffic Class" field is an 8-bit traffic class field which can be used to identify and distinguish between different classes or priorities of IPv6 packets. "Flow Label" is a 20-bit flow label which will be specified in more detail later in the research paper. "Payload Length" is 16-bit unsigned integer that denotes the length of the IPv6 payload, which is the rest of the packet following this IPv6 header, in octets. The "Next Header" field is an 8-bit field that identifies the type of header immediately following the main IPv6 header. This is the part of the header that indicates whether or not there is an extension header. "Hop limit" is an 8-bit field that is decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero. The "source" and "destination" address fields are both 128-bits to denote the originator and intended recipient of the packet. [10]

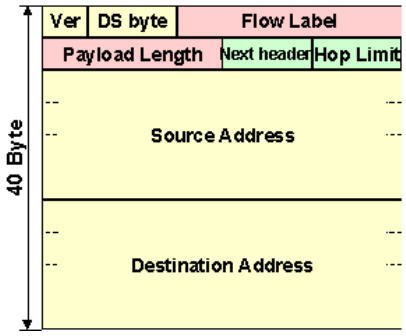

Figure 2.9: IPv6 Header Format [11]

2.9.3 Improved Support for Extensions and Options

IPv6 allows for the use of extensions and options to the main header. These extensions and options provide the ability to encode additional information in the packet header. According to [10], the way IPv6 header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future, compared to the way IPv4 options were encoded [10].

There are two types of extension headers this research examines. They are the hop-by-hop options extension header and the destination options extension header. They both are used to carry optional information. They differ in that the hop-by-hop options extension header requires all nodes in a packet's transmission path examine the packet's optional information stored in the extension header, while the destination options extension header only requires the destination to examine the packet's optional information. Both extensions are set to 24 bytes long, where eight bits of the extension is reserved for the next header field of the extension header, and another eight bits is reserved for the length of the extension header, not including the first eight bits [10].

2.9.4 Flow Labeling Capability

IPv6 provides a new capability to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling [10]. The use of this field allows for the labeling of traffic belonging to a specific task or mission.

2.10 Future Direction of IPv6 on U.S. Air Force Networks

Another reason why this research is relevant to current cyberspace research is because of the push by the U.S. Air Force to transition from IPv4 networks to networks that are IPv6 capable [12]. The USAF IPv6 Transition Plan states it is the U.S. Air Force's goal to complete the transition to an IPv6 capable network by the fiscal year of 2012 [12]. In light of this timeframe, finding new ways to leverage the capabilities of IPv6 networks is important.

2.11 Network Management Overview

Another aspect this research examines is finding a way for leaders to get the latest status of the mission. This is accomplished by monitoring all network assets through the use of a network management system (NMS).

In Figure 2.10, a typical NMS is depicted. A typical NMS consists of an overall management entity, at least one managed device, the management software agents and management databases on the managed devices, and a protocol of exchanging information between the managed devices and the overall management entity. The overall management entity is responsible for overseeing the management of the entire network and network nodes. The overall management entity provides commands or requests for information to the managed devices via a network management protocol. Simple Network Management Protocol (SNMP) is the name of the protocol that allows communication between the overall management entity and the managed devices. The software agents on the managed devices process the commands from the management

entity, and either, provide a response back to the management entity, or store information onto the agent database, which is also located on the managed device [13].

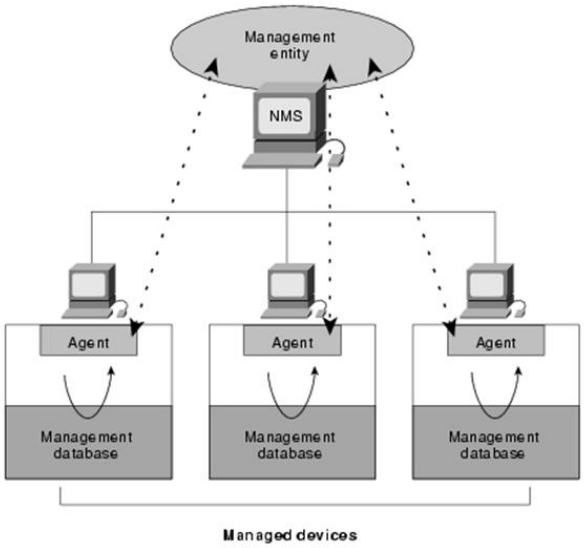

Figure 2.10: Typical Setup of a Network Management System (NMS) [13]

The research presented in this paper leverages the capabilities of a network management system. As mentioned earlier, inside the managed devices resides a management database. This database stores all network management aspects that are relevant to that specific managed device. Specifically, information about the packets that traverse the managed device can be stored on the managed device's database. Finally, the managed device can send the information stored in its database to the overall management entity for further examination.

In the CITS Block 30 specifications, the main NMS would be located in the I-NOSC portion of the network. In Figure 2.11, the overall NMS that oversees all the

network components in AF network is located in the I-NOSC block. Each base or location may have their own local NMS, but the main NMS located at the I-NOSC is the NMS that collects information from lower levels to manage the entire AF intranet. Also, the main NMS at the I-NOSC is where the mission code database resides, which enables the correlation between missions to network components and vice versa.

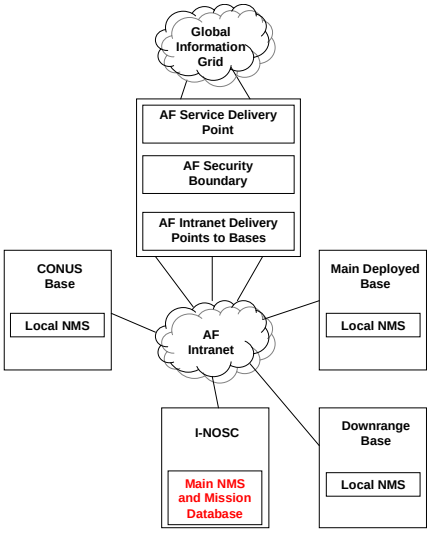

Figure 2.11: Location of Overall NMS and Mission Database [9]

III. Methodology

3.1 Introduction to the Methodology

To provide leaders with the most up to date situational awareness of ongoing missions and operations, this research proposes injecting mission metadata directly into the network and data streams. One way to do this is by embedding mission metadata into the header of the packets, specifically the IPv6 packet header. This chapter outlines a way to determine the feasibility of embedding information into the header. To determine the feasibility of embedding mission information into the packet header, several methods of embedding the data into the header are examined. Analytical experiments are set up and calculations are conducted to determine which of the methods provides the best end-to-end latency of a file transfer, as well as which method produces the least amount of additional network overhead required for that specific method's operation. Finally, validation of the network and the experiments is provided.

3.2 Goals of the Study

As mentioned previously, this study has several goals. The over-arching, strategic goal of the study is to determine a way to provide leaders with most up-to-date status of critical missions. In addition to looking at traditional ways of providing status, via the use of management systems, this research examines how to improve on these methods and management systems by investigating some novel approaches. The tactical goal of this research is determining which missions are affected when network degradation and outages occur. In order to do this, this research first determines if encoding a mission

code onto an IPv6 packet is feasible. Next, different methods of encoding mission information onto IPv6 packets are determined and evaluated. The methods are evaluated by determining which allows the best end-to-end latency of a file transfer. Several factors are examined such as network size, file size, and frequency of updates sent to the management system. For this research, end-to-end latency is defined as the time it takes for the first packet of a file to be transmitted by the source to when the last packet of the same file is received by the destination. Additionally, additional network overhead is defined as the additional bits introduced into the network and is in the form of additional header bits and additional bits associated with management packets that update the management system. These management packets are referred to as update packets elsewhere in this paper.

3.2.1 Determining Feasibility of Encoding Metadata onto Headers

To answer question of feasibility, this research first determines locations where metadata can be stored on an IPv6 packet. Then an examination of the best encoding method of metadata is performed. Finally, a determination of whether or not the best method would be feasible on U.S. Air Force cyber command assets is made.

3.2.2 Determining Best Method of Encoding Metadata

To determine the best way to imbed mission metadata into headers, we look at the different methods of imbedding info. The three methods that are examined utilize the IPv6's header fields. The flow label method examines the use of the flow label field in the main portion of the IPv6 header. The other two methods use the extension headers to store metadata about missions. These methods are named the "hop-by-hop options extension header" method or "hop-by-hop" method and the "destination options extension header" method or "destination" method.

For each of the different methods, certain networking performance aspects are examined. Specifically, this research determines which method performs best in terms of having the least amount of end-to-end latency for a packet transmission as well as introducing the least amount of additional network overhead into the network. These performance aspects are compared to baseline calculations that have had no encoding done on the packets.

3.3 Overall Approach

To determine if it is feasible to encode mission data onto network packets, research is first performed on determining the location on a network packet this type of information can be encoded. Looking specifically at the IPv6 packet, it needs to be determined what portions of the IPv6 header can be modified to accommodate mission data type information. Once it is determined that mission data can be encoded onto a IPv6 packet, measuring how encoding mission data affects the network, and more specifically the mission critical files using mission-encoded IPv6 packets, needs to be

completed. End-to-end latency of the transmission of mission-encoded files needs to be measured. Additionally, the additional amount of network overhead incurred by encoding mission data onto packets needs to be measured. Finally, measuring the amount of network overhead, which the network incurs when a management system retrieves the mission data, is required to be accomplished.

3.4 Defining the System Under Test

The System Under Test (SUT) is the Theater/Enterprise-wide computer network supporting the commanders missions and tasks. The Components Under Test (CUT) are the network infrastructure, mission/task system servers, network management system overseeing all aspects of the network infrastructure, the database that holds all mission information and corresponding mission codes, and the mission files and packets that traverse the network. Thirteen parameters are examined, of which four are factors that are varied during the analytical calculations. Additionally, three workloads are taken into account, with three outcomes produced. The SUT is depicted in Figure 3.1.

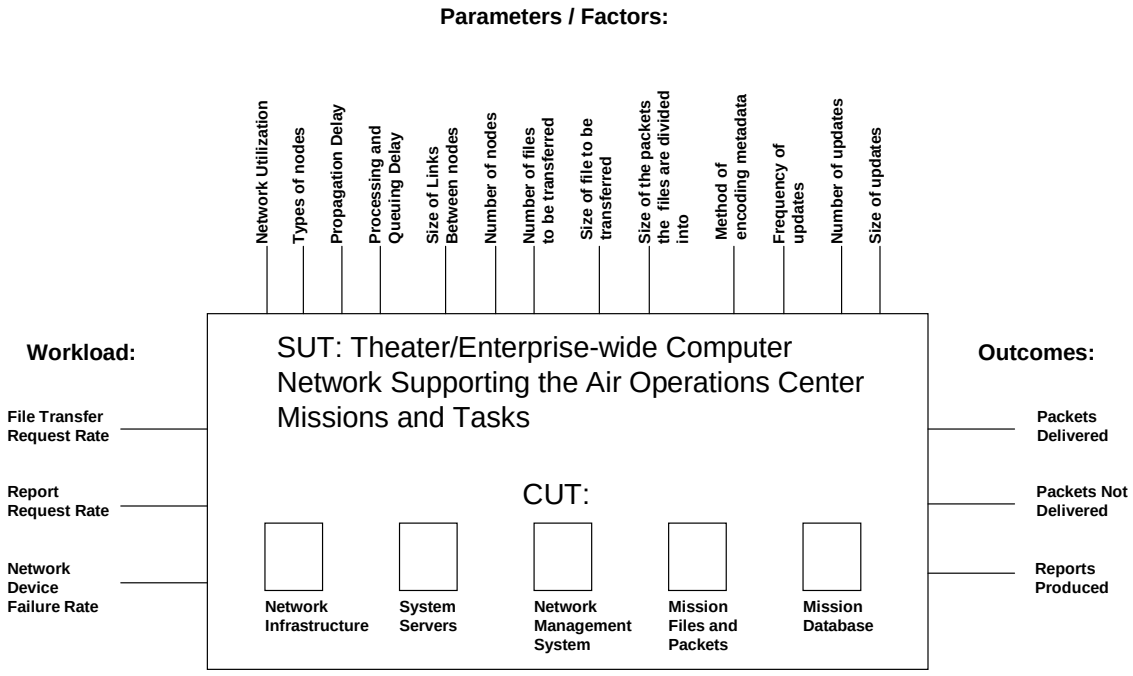

Figure 3.1: System Under Test (SUT)

3.5 Listing System Services and Outcomes

The services of the SUT include transporting mission critical data packets and tracking status of mission data and packets. Outcome of the services provided by the Theater/Enterprise-wide computer network are the following: 1) packets being delivered, 2) packets not being delivered, and 3) reports being produced that show which mission is being affected by a network outage or degradation.

3.6 Selecting Appropriate Metrics to Examine

The purpose of this study is not an extensive examination of a specific network. Instead the research sets out to determine the feasibility of encoding mission type data onto packet headers. Although there are many aspects that determine the performance of any network item, the metrics that are pertinent to determine the feasibility of this study has been narrowed down to two: end-to-end latency of file traversing the network and the additional network overhead introduced into the system for each of the different encoding methods.

To examine the end-to-end latency of a packet, this research examines the time it takes for the packet to traverse the source node, all intermediate nodes, and the end node. The four main aspects of end-to-end latency are examined, although not all four aspects are specified in detail to determine each method's end-to-end latency.

To examine the additional network overhead, the total number of additional bits introduced into the network due to the different encoding methods is compared to the baseline calculation of a packet traversing the network with no mission encoding. Both of these metrics are discussed in more detail in this section.

3.6.1 End-to-end Latency

To calculate end-to-end latency for a file transmission, four types of latency or delay are considered. They are the transmission delay, propagation delay, queuing delay, and processing delay. Transmission delay is determined by calculating the time it takes for a node to transmit a certain sized or length packet or file onto a network link of a certain rate. "*L*" is used to denote the size or length of a packet or file and is usually in

bits. "*R*" is used to denote the rate or bandwidth of the link onto which the packet or file is being transmitted on and is usually in bits per second [14]. The equation for transmission delay or T_{Tr} is

$$T_{Tr} = {}^{L}\!/_{R} \text{ (seconds)}$$

Equation 3.1: Transmission Delay [14]

Propagation delay is the time it takes for something to propagate or traverse a certain distance. Propagation delay is determined by calculating the time it takes for a bit of the packet or file to traverse a certain distance. The speed of the bit on the transmission link is close to the speed of light. "*D*" denotes distance between nodes and is usually in meters. "*C*" denotes the speed of light in the transmission medium and ranges between 2.5 to 3 X 10^8 meters per second, depending on the medium of the transmission link [14]. The equation for propagation delay or T_{Prop} is

$$T_{Prop} = {}^{D}\!/_{C} \text{ (seconds)}$$

Equation 3.2: Propagation Delay [14]

Queuing delay, or T_Q, is the time that information waits in a queue at a node before being transmitted. The amount of time waiting in the queue depends on the size of the queue, the speed at which information is taken out of the queue to be processed or transmitted, and the rate at which new information arrives at the node and enter the

queue. There is a level of uncertainty as to when network traffic arrives the nodes, and the probability of arrivals is taken into account when determining the queuing delay [15].

Processing delay or T_{Proc} is the time it takes a node to process a bit, packet, or file. Processing at a node occurs when a bit, packet, or file arrives a node and is required to be used by the node's high level entities, like software programs that reside on that node. Processing also occurs at a node when a bit, packet, or file is required to be transferred or routed to another node [14].

The total end to end latency for a file or a packet to traverse a network is the sum of all four delays. The total end to end latency or $T_{E\text{-}E}$ is show in Equation 3.3 below.

$$T_{E\text{-}E} = T_{Tr} + T_{Prop} + T_{Q} + T_{Proc}$$

Equation 3.3: Total End-to-end Latency Calculation [14]

In the networks defined and examined in this research, the main focus is on how the different methods differ in terms of their transmission and processing delay. Although propagation and queuing delay are important aspects of end to end latency, their contribution to a file's end-to-end latency is negligible in this research. Reason being the nodes in the networks are in close proximity of each other, making distance negligible, thus making propagation delay negligible. Also, there is no additional network traffic produced by other entities, other than the traffic produced by the mission servers being examined. Therefore, we know the arrival rate of the packets being transmitted by the mission servers, traversing the network nodes. Finally, in order to not

drop any packets due to a filling queue at a node, the nodes in this research are provided an infinitely size queue.

3.6.2 Additional Network Overhead

As mentioned earlier, this research examines the total number of additional bits each one of the different encoding methods introduces into the network. Each method's results are compared to a baseline calculation. Overhead is in the form of additional bits introduced by way of additional header bits or additional management packets required to be sent to update the mission code database.

3.7 List of Parameters

A parameter is a measurable factor that defines a system and determines its behavior in an experiment [16]. In the experiments in this research, the following list of parameters in Table 3.1 is used. These parameters help determine which encoding method is the best in terms of the metrics already explained in section 3.6.

Table 3.1: List of Parameters

List of Parameters
network utilization
types of nodes (server, router, switch, etc.)
propagation delay between nodes
Processing and queuing delay at nodes
size of network links between nodes
number of nodes between source server and destination
number of files to be transferred between source and destination
size of the file to be transferred between source and destination
size of the packets the files are divided into
method of encoding the metadata onto the file's packets
frequency of the nodes providing update messages to management system
number of update messages sent to management system
size of the update messages sent to the management system

To simplify the calculation of end-to-end latency and additional network overhead of each of the different methods, several of the above parameters are set to one specific value. The size of the network link (or bandwidth) between nodes is set to one gigabit per second or 1Gbps. Although 1Gbps is technically 1024^3 bits per second or 1,073,741,824 bits per second, 1Gbps is rounded and simplified to 1,000,000,000 bits per second for the calculations in this research. Also, the number of files to be transferred from the source to the destination is set to ten files. Finally, the size of the update message that is sent to the management system is set to 2000 bits.

3.8 List of Factors

Of the list of parameters, only a select few are examined to be varied as factors. Table 3.2 shows the different factors to be varied and their respective levels.

Table 3.2: List of Factors and Their Respective Levels

List of Factors	Levels
Size of mission file	1 MB
	5 MB
	25 MB
Number of nodes in information chain	small number (5 nodes, including end nodes)
	medium number (10 nodes, including end nodes)
	large number of nodes (20 nodes, including end nodes)
Method of encoding mission data onto packets	Baseline
	flow label method
	hop-by-hop option extension header method
	destination option extension header method
Frequency of updating the mission database	Instantaneous
	Periodic

The 1MB file represents a typical email with a possible attachment. The 5MB file represents an email with multiple attachments or image files. The 25MB file represents large file transfers, normally associated with the transfer of imagery. The small number of nodes represents traffic traversing a local area network. The medium number of nodes represents traffic between two adjacent bases or two adjacent local area networks. The large number of nodes represents traffic that must traverse many network routers or switches.

To further define the factor “frequency of updating the management system and mission database,” the levels “instantaneous” and “periodic” are defined. In an instantaneous update, an arriving packet gets processed and the mission code gets extracted. As soon as this occurs, the node immediately sends a management packet to

the management system and mission database. Therefore, for instantaneous updates, nodes send update packets as often as they transmit file packets along to the next node.

A “periodic” update behaves like an “instantaneous” update, but they are not sent as often. In a “periodic” update, an arriving packet gets processed and the mission code is extracted. However, instead of sending an update packet immediately after each packet transmission, an update packet is sent only once during the time the entire file is being transmitted. There may be other ways to define the “periodic” update, such as random-based or probabilistic-based, but in this research, “periodic” is defined in the manner above. A periodic update is different from an instantaneous update in that the frequency of sending update packets is noticeably less.

Since this study is an analytical study, the experiments are the different calculations for latency and additional network overhead for each method. Each calculation is only required to be completed once since there is no randomness like there is in a simulation or real world network. Since there are no replications required for each of the different experimental setups, the number of analytical calculations required to be conducted is 72.

3.9 Experimental Setup

The networks that are modeled in this research are based on hypothetical networks discussed in the Combat Information Transport System (CITS) Block 30 design specifications. The small network information chain is modeled after missions occurring on an internal base network, with no mission traffic leaving the base. The medium network information chain is based on adjacent CONUS bases that have missions

running between them. Finally, the large network information chain is based on an inter-theater type network, in which missions are running on different bases, in different parts of the world.

In the networks examined in this research, the concepts of the CITS Block 30 configuration for base, AF intranet, etc, is used. However, specifying every aspect of the CITS Block 30 network is out of the scope of this research. Instead, we specify simple networks, using the layout of a CITS Block 30 network as a guide. The networks are simplified in order to keep the focus of the research on the behavior of the different encoding methods versus various technical and routing aspects of a CITS Block 30 network.

For example, the small network has five nodes, to include the source and the destination nodes. There are three intermediate nodes in between the source and destination. The intermediate nodes can act as routers or switches, and the source and destination nodes are normally servers or workstations. The network links in between the nodes are fixed at 1Gbps. Now that the network has been set up, calculations on end-to-end latency and additional network overhead can be calculated, based on the method being examined on this network. Figure 3.2 is the example network described above.

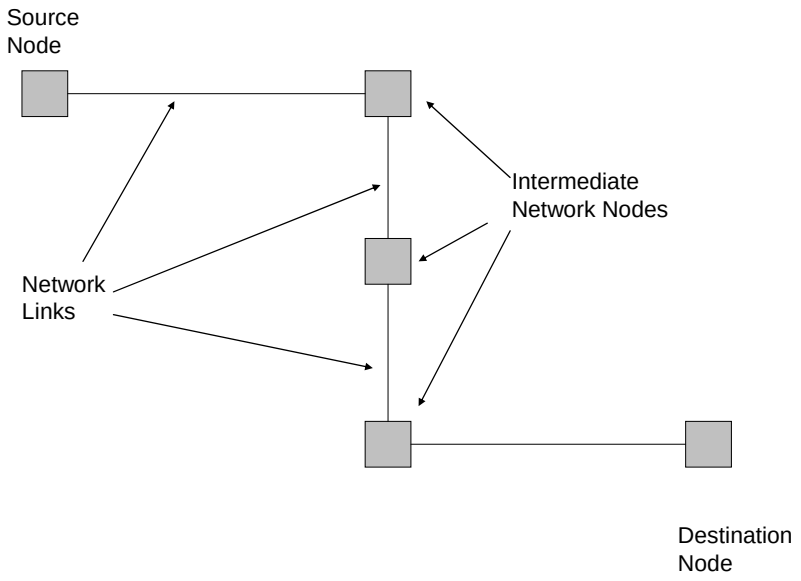

Figure 3.2: Small Five-node Network

3.10 Defining the Behavior of the Encoding Methods

In this research, the header of the IPv6 packet is examined as a location to store mission or task type metadata. There are several locations for information to be stored onto the header. This research examines the use of each of those locations. Additionally, the behavior of the intermediate network nodes depends on which part of the header is used. Therefore, the behavior of the nodes for each of the methods is defined. Using certain header extensions and options causes nodes to either pass the packet along, without a penalty of additional processing delay, or process the packet, thus adding additional processing delay, at that specific node. Finally, once the network nodes process or pass along the packets, the node may or may not be required to send an SNMP update to the overall network management system overlooking the computer network.

The frequency of sending updates is determined by the method being used. The SNMP packet is what updates the mission database with the latest information of which missions are traversing the network. The three methods this research examines is the use of the flow label field, use of the hop-by-hop option extension header, and finally the destination option extension header. These three methods are compared to the baseline performance of the network. The behavior of each of these methods is defined in detail below.

3.10.1 Flow Label Method

The Flow Label Method uses the 20-bit Flow Label field in the IPv6 packet header. The assignment of the flow label can either be random or defined. Use of this label is optional; however, every IPv6 packet has this label in the main portion of the header. [10]

During the flow label method, the source node transmits the packet into the network. The source keeps track of the flow label that was inserted into the packet. Depending on the frequency of the updates (either instantaneous or periodic update), a SNMP packet is transmitted to the overall NMS and mission database. Since the flow label can either be randomly assigned or directly assigned, direct assignment is chosen to provide a level of control of the flow label. Prior to assigning a flow label number to the packets, the source node will have "learned," by way of SNMP updates from the NMS, the different mission codes each flow label should have. These SNMP updates from the NMS do not increase the network overhead since SNMP messages are normally exchanged between the NMS and all the nodes in the network [13]. The source node

keeps track of the mission numbers and assigns flow labels accordingly. Additionally, the intermediate nodes, once receiving the packet, examine the header for the flow label. Each intermediate node updates its internal flow label value table once the flow label of the incoming packet is read. There is a certain amount of processing required for each packet in order for the node to update its internal flow label value table. For the purposes of this research, a processing delay penalty is assigned to nodes that have to process packets since the network nodes are not specified in enough detail to determine actual processing delay for each network node.

An example of how a node's flow label value table might look shown in Figure 3.3. There are 2^{20}-1 possible values that could be used, or almost 1 million values, for the flow label, due to the flow label being 20-bit number. To further specify the behavior, if more than one million flow label values are to be used, the overall NMS and database work with the nodes to recycle the numbers accordingly, as missions and tasks are completed. Only the flow label number is encoded into the flow label field. Source, Destination, Application, and Mission are not encoded into the flow label field. They are, however, stored and managed at the mission database and overall NMS.

Flow Label #	Source	Destination	Application	Mission
1000-1999	A	B	X	from database
2000-2999	A	B	Y	from database
3000-3999	A	B	Z	from database
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
999000-999999	Z	Z	YY	from database

Figure 3.3: An example of the Flow Label Value Table

If SNMP updates are sent instantaneously, the nodes in the network produce and send an SNMP update immediately after each of the file packets are received and transferred. If SNMP updates are sent periodically, the nodes produce and send an SNMP update once per file being transferred across the network.

3.10.2 Hop-by-hop Options Extension Header Method

The hop-by-hop options extension header method or hop-by-hop method uses the extension header option found in the IPv6 packet header. This extension can store several bytes worth of data in addition to the extension header's two mandatory 8-bit fields. This research sets the extension header size to 24 bytes total. The hop by hop options extension header is used to carry optional information that must be examined by every node along a packet's delivery path [10].

During the hop-by-hop method, the source node inserts the metadata into the extension header, and every node along the path of the file transmission processes the packets. SNMP packets are sent to the NMS by the nodes that process the packet. The processing delay penalty is incurred for each of the packets each node has to process. If SNMP updates are sent instantaneously, all of the nodes along the transmission path produce and send SNMP updates immediately after each of the file packets are received and transferred. If SNMP updates are sent periodically, the nodes produce and send an SNMP update once per file being transferred across the network.

3.10.3 Destination Options Extension Header Method

The destination options extension header method or destination method behaves similarly to the hop by hop method. The destination options extension header is used to carry optional information that is only examined by the destination or the last node along packet's delivery path [10]. SNMP packets are only sent by the source node and the destination node, either instantaneously or periodically.

3.11 Additional Notes about the Different Aspects of the Network

Although it was mentioned that the networks in this research are not specified into too much detail, several items do require specification in order to define the behavior of the networks and the file and update transmission between nodes. These notes include pipelined file transmissions, handshaking packets, updates to the NMS and database, size of the SNMP update, and the NMS and SNMP messages.

3.11.1 Pipelined File Transmissions

Files are transferred via a pipelined transmission. This means that packets are transmitted one right after the other. There is no time gap between packet transmissions.

3.11.2 Handshaking Packets

The three-way handshake that is used to establish a connection between a source node and the destination node is not used in the calculations. It is assumed that the handshake has already occurred prior to the transmission of the actual file.

3.11.3 Sending Updates to the NMS and Mission Database

In all methods, the source node sends at least one update packet to the NMS at the beginning of the transmission of the file. Depending on the method, the source node may send more updates. Additionally, the other downstream nodes may send updates.

3.11.4 Size of the SNMP Update

The size of the SNMP update has been set to 2000 bits. Although the minimum size of the SNMP packet is 484 bits [17], arbitrarily setting the payload size to 2000 bits is assumed to be enough to accommodate all management data require of the NMS and mission database.

3.11.5 Notes about NMS and SNMP Messages

An NMS, either the local NMS or the overall NMS, manages every network asset of the computer network being examined. If required, local NMS's will exchange data with other base NMS's or the overall NMS in order for the mission metadata to be processed properly by the mission database. Each managed network asset has properly configured management software that keeps track of all pertinent management data that is required to be tracked.

3.12 Validation of Experimental Setup

Now that the network has been determined and the behavior of each of the different methods in the network has been defined, validation of the experiments is required. Since validation means making sure what is built is doing what it was meant to do, this section shows how the network is validated, how the files being transferred for each of the methods is validated, and how the update messages are sent to the NMS is validated.

3.12.1 Validation of Network Mechanics

In order to validate the network, ensuring that the network is operate the way is supposed to operate is required. First, ensuring the nodes of the network operate properly is required. Therefore, the definition of what a network node does is provided. If the network node is a source node, that node behaves like a server that is providing data to someone or something that has requested it. If the network node is a destination node, that node behaves like the person or entity that has requested the information from the server or source node. The data flow, in this example, is from the server node to the destination node. The intermediate nodes between the source node and the destination node are a mixture of routers and switches. In the network in these experiments, the intermediate nodes all act like routers or switches, in which they receive a packet, process the packet to ensure proper routing of the packet, and transmit the packet toward the final destination.

In Figure 3.4, a small network is shown, and behavior of the nodes and packets are discussed to further validate the network behavior of these experiments. A method is

chosen, and a packet is transmitted across an example network. For the example, the flow label method is chosen to be examined in a multi-node network. There are four nodes that are in between the source and the destination. Only the three bottom nodes (nodes 1, 2, and 3) are part of the information asset chain of the mission. Node 4 provides the path to the overall NMS. As a mission packet arrives at node 1, it is processed. Eventually, the packet is transmitted to node 2 and then on to node 3 towards the destination.

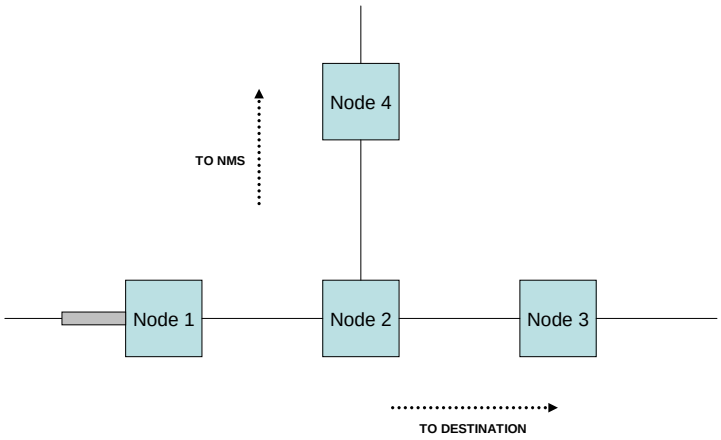

Figure 3.4: Example Small Four-node Network

In Figure 3.5, the inner workings of node 1 are depicted. In step 1, the file packet arrives and enters the RX or receive queue. In step 2 and 3, once the packet is entirely received, the node starts to process the packet. As soon as the packet is processed for mission metadata, the node sends an SNMP update packet to the NMS. Shown in step 4

is the update packet being formed and being transmitted (TX). In step 5, the file packet is moved over to be transmitted after the SNMP update packet is transmitted. Finally, the node transmits the file packet to the next node.

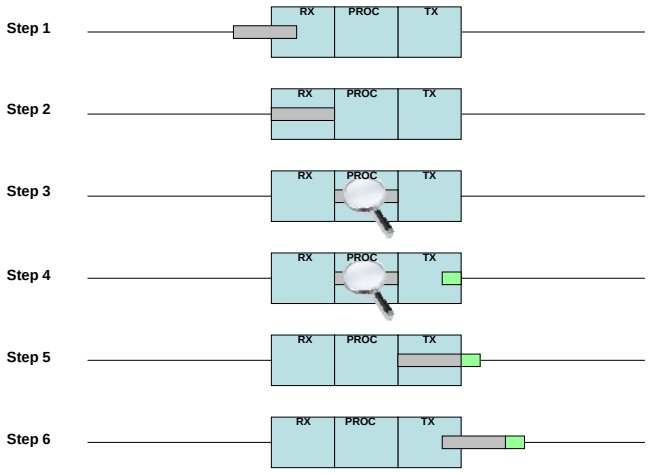

Figure 3.5: Inner Workings of a Node 1

In Figure 3.6, the examination of the four node network continues. The update packet and file packet traverse the network from node 1 to node 2. Update packets are sent from node 2 to node 4. File packets are directed from node 2 to node 3 toward the destination.

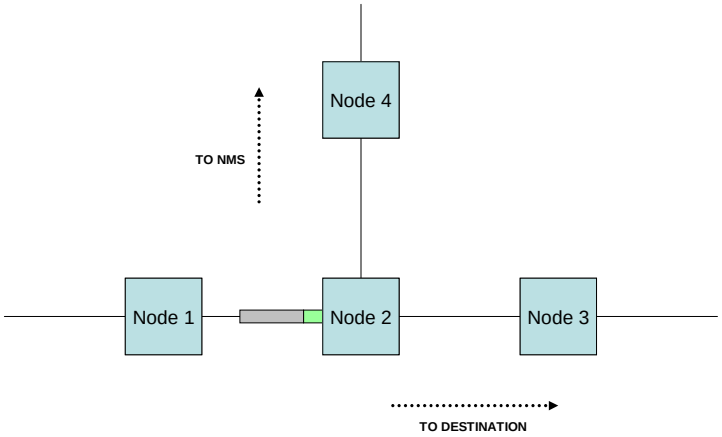

Figure 3.6: Continued Examination of Example Small Four-node Network

In Figure 3.7, the inner workings of node 2 are examined. In step 1 and 2, the SNMP update packet (A) from node 1 arrives node 2. In step 3, the node processes the update packet A and determines that the update packet A must move onto the NMS. In step 4, the node transmits the update packet (A) onto the link that leads to the NMS, which is the link to node 4. The node then processes the file packet once the entire file packet arrives the RX side. In step 5, after processing the file packet, another update packet (B) is generated by node 2 and sent out the link towards the NMS. In step 6 and 7, the file packet is transmitted onto node 3, which is the link toward the destination.

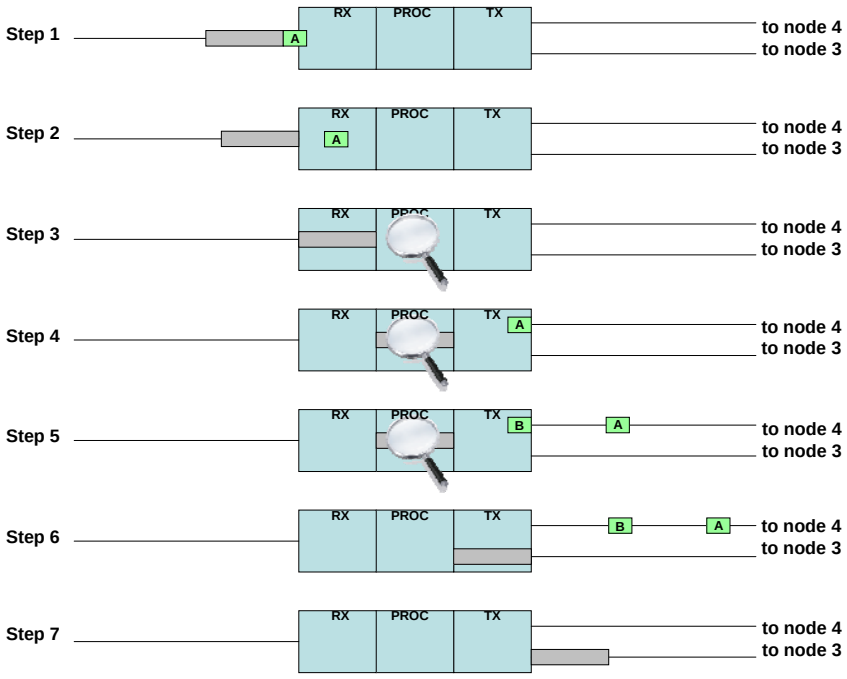

Figure 3.7: Inner Workings of Node 2

In Figure 3.8, the update packets are transmitted toward the NMS and the file packet continues along the information asset chain towards the destination. In the event that the link between node 2 and node 4 goes out, the update packets will be routed another way to the NMS. It is assumed that there is another route that these update packets can take. It is out of the scope of this research to determine the routing of the update packets since there are too many aspects of the network that are not entirely specified.

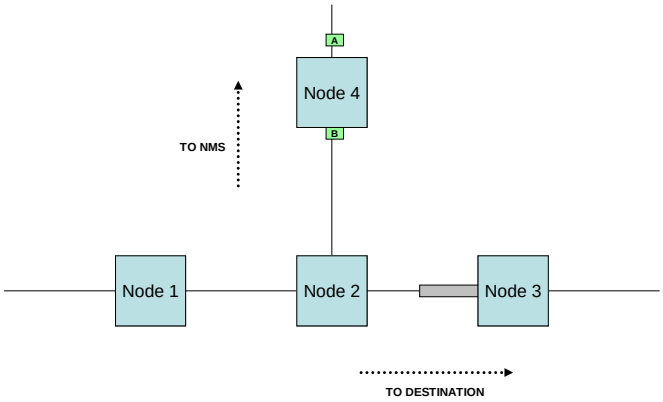

Figure 3.8: Final Examination of Example Small Four-node Network

3.12.2 Validation of Proper Network Setup

In order to determine if the network is set up the way it should be set up, examining how well the network mimics a CITS Block 30 network ensures validation of the network setup. The experiments in this research are comprised of using networks of certain sizes. The sizes correspond to the amount of nodes information must traverse when traversing from CONUS bases, through the AF Intranet, to a deployed base, and vice versa. In the small network setup, there are five total nodes. A small network mimics intra-base communications between source and destination. In other words, the source and the destination reside on the same base and local area network. Figure 3.9 shows this setup, where the dotted red line represents the information flow between two servers, one being the source, and the other being the destination.

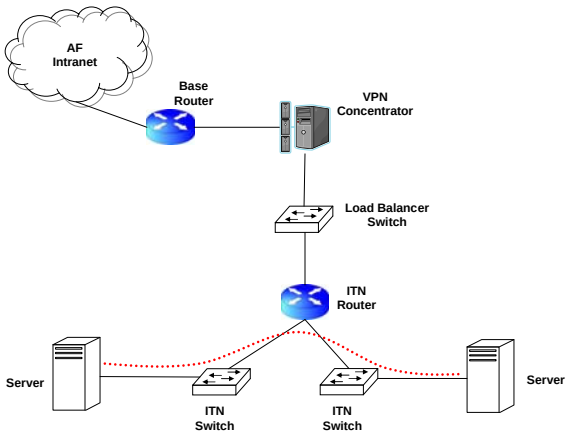

Figure 3.9: Validation of Small Network Setup

The medium and large sized networks are sized according to the location of the source and the destination servers as well. The medium sized network represents the situation where the source and destination reside on two different bases, which are close in proximity to each other, only being separated by a router or two. Figure 3.10 depicts this network setup.

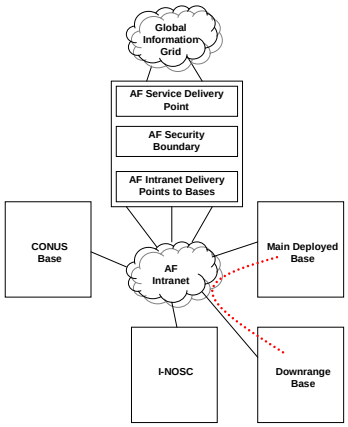

Figure 3.10: Validation of Medium Network Setup

The large sized network represents the situation where the source and destination are far away from each other, possibly crossing an ocean or continent and being separated by many routers. Figure 3.11 depicts this setup, where the red dotted line represents the information flow between the servers.

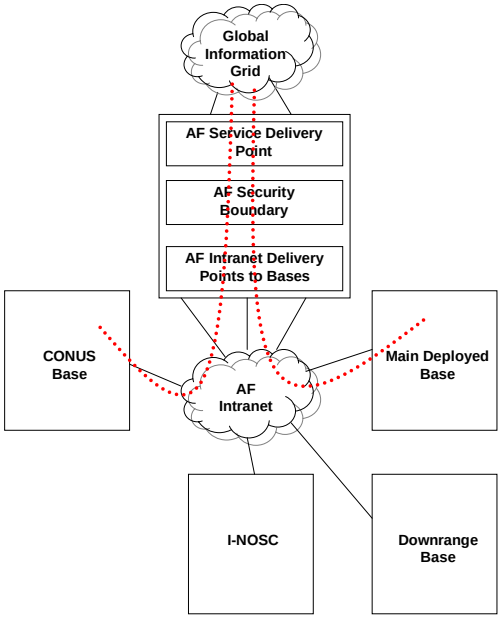

Figure 3.11: Validation of Large Network Setup

In real life networks, propagation delay becomes more and more of a factor as distance between nodes increases. However, as mentioned previously, the calculations in this research neglect propagation delay to simplify calculations and focus on the encoding methods versus endlessly specifying the networks.

3.12.3 Validation of End-to-end Latency Calculation

In order to determine whether or not the calculations to determine end-to-end latency are correct, calculations of end-to-end latency are examined in detail. Specifically, how files and update messages are transmitted, when they are transmitted, and how much additional network overhead is injected into the network is what is validated. The first calculation to be validated is the end-to-end latency of the baseline configuration. Then, the end-to-end latency calculations of one of the three encoding methods are examined.

In Figure 3.12, the details of a node's file transmission is specified and examined. A file of a certain size is divided up into certain sized packets. The packets are depicted in the Figure 3.12 as a grey square. In order to determine end-to-end latency, several delays need to be calculated. The first is the transmission delay, denoted as t_f. To ease latency calculations, propagation delay between all nodes is set to zero. Additionally, should the nodes need it, there is an unlimited queue size at each of the nodes. For the baseline calculation, there is no additional processing of the packets required. Therefore, there is no processing delay.

In this example, as the source node transmits the first bit of the packet, it arrives in the receive queue of the next node. The next node will not transmit the packet until all of the bits arrive the node. Additionally, to further specify the behavior of the nodes' transmissions, it was mentioned earlier that the source node transmits the file packets in a pipelined manner. In this diagram, the number of packets being transmitted is denoted as *p*.

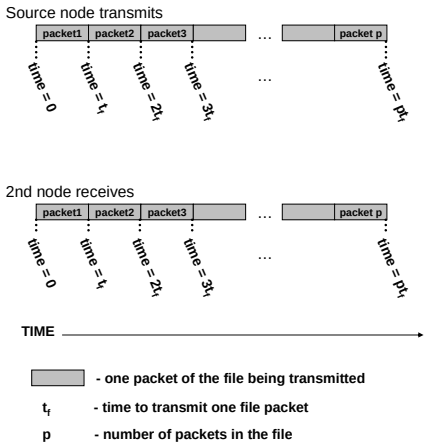

Figure 3.12: Step 1 of Validation of Baseline End-to-end Latency Calculation

In Figure 3.13, what is happening at the second node is depicted. At the second node, the entire packet arrives the node before being transmitted to the next node. Since the source node transmitted the packets in a pipelined manner, the next node will receive the packets one right after another. The time since the source node started transmitting the file to when the second transmits its last packet is $(p+1)t_f$.

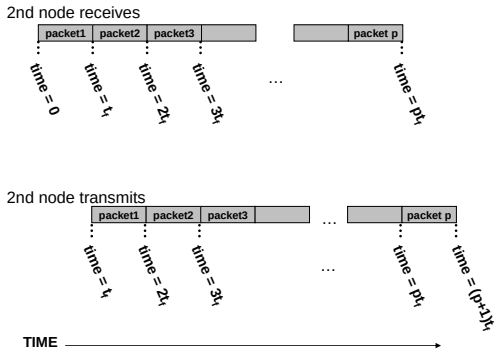

Figure 3.13: Step 2 of Validation of Baseline End-to-end Latency Calculation

Figure 3.14 depicts the transmissions between the second and third nodes. There is no propagation delay between the nodes. As the second node sends the a bit, that bit immediately arrives the third node.

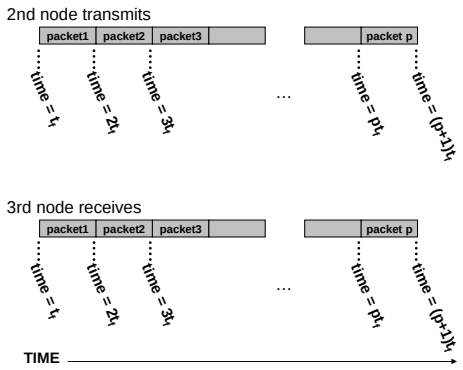

Figure 3.14: Step 3 of Validation of Baseline End-to-end Latency Calculation

Figure 3.15 depicts how the packets are received and transmitted at the third node. Much like the second node, the entire packet must first be received before being sent to the transmission side of the node for transmission. The time elapsed since the source node started transmitting the file is $(p+(n-1))t_f$, where p is the number of packets in the file, n is the number of nodes in the information flow chain, and t_f is the transmission time for a file packet.

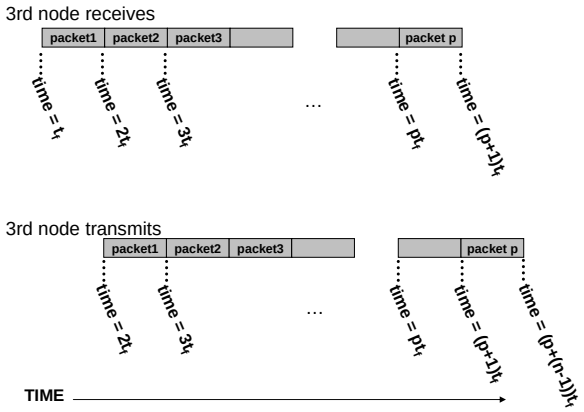

Figure 3.15: Step 4 of Validation of Baseline End-to-end Latency Calculation

For the flow label method with instantaneous updates to the NMS, for every packet that is transmitted to the next node, an update packet is sent to the NMS. Figure 3.16 illustrates the behavior. Here, t_f denotes transmission time for a file packet, t_u denotes the transmission time for an update packet, and t_p denotes the processing time of a file packet. The grey block represents the amount of time it takes for a packet transmission, the light blue block denotes the time it takes for an update transmissions, and the red block corresponds to the amount of time it takes to process the file packet. The source node alternates sending file packets and update packets. As the second node receives the packets, the node processes the packet for the flow label and updates its flow label table. In this situation, the queue of the second node does not grow because the time between packet arrivals is the same as the time it took for the previous node to send an update packet. In this network, the time it takes to process a packet is less than the

time it takes to send an update packet. After the node processes the packet, the node transmits the packet on to the next node. Update packets are sent after each file packet transmission.

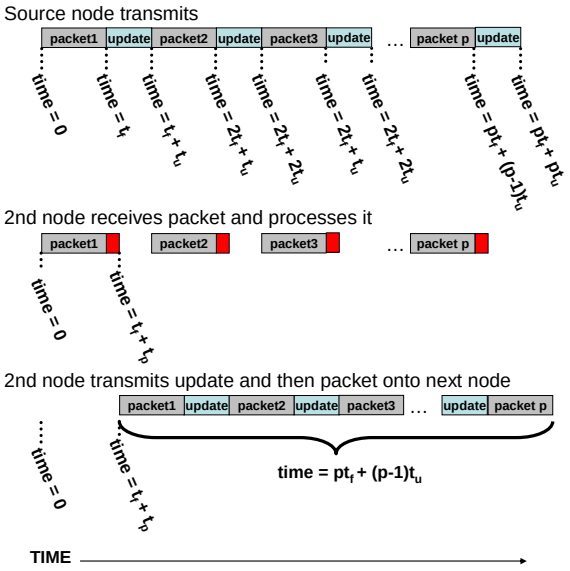

Figure 3.16: Step 1 of Validation of the Flow Label Method End-to-end Latency Calculation

In Figure 3.17, the validation of the calculations of the flow label method with instantaneous updates continues. Following through with the calculation, the end to end latency for a file with a number of p packets takes, doing instantaneous updates in a n-sized network, is $(p+n-1)t_f + (p-1)t_u + (n-1)t_p$.

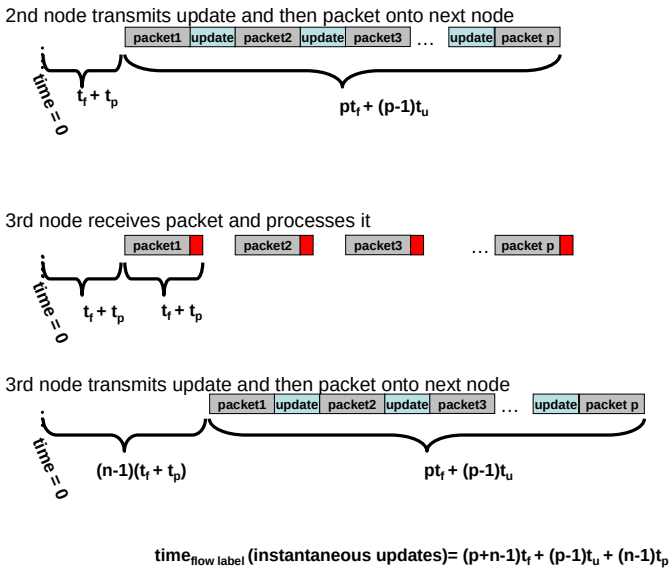

Figure 3.17: Step 2 of Validation of the Flow Label Method End-to-end Latency Calculation

Finally, validation of the end-to-end latency calculation is conducted when periodic updates are sent to the NMS. In periodic updates of the flow label method, significantly less updates are sent. After the source sends the first packet of the file, the source sends an update to the NMS. However, this is the only update that is sent for the rest of the current file transmission. Figure 3.18 illustrates the behavior. Again, t_f denotes transmission time for a file packet, t_u denotes the transmission time for an update packet, and t_p denotes the processing time of a file packet. The grey block represents the amount of time it takes for a packet transmission, the light blue block denotes the time it takes for an update transmissions, and the red block corresponds to the amount of time it takes to process the file packet. The source node sends the first packet and then the update packet. Then the source node sends all the rest of the packets. As the second

node receives the packet, the node processes the packet for the flow label and updates its flow label table. After the node processes the packet, the node transmits the packet on to the next node. As the second node transmits the packets after being processed, there are gaps that form between the packet transmissions. This is due to the node having to process the packet before transmitting the packet on.

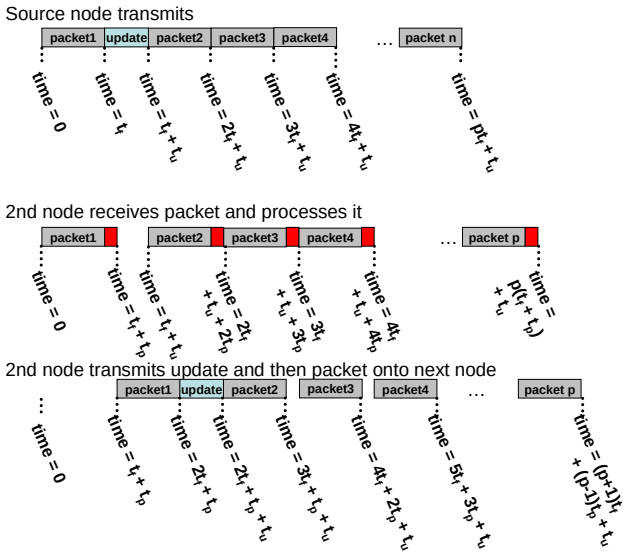

Figure 3.18: Step 1 of Validation of the Flow Label Method End-to-end Latency Calculation with Periodic Updates

In Figure 3.19, the rest of the nodes calculations are validated. Following through with the calculation, the end to end latency for a file with a number of *p* packets takes, doing periodic updates, is $(p+n-1)t_f + (p+n-3)t_p + t_u$.

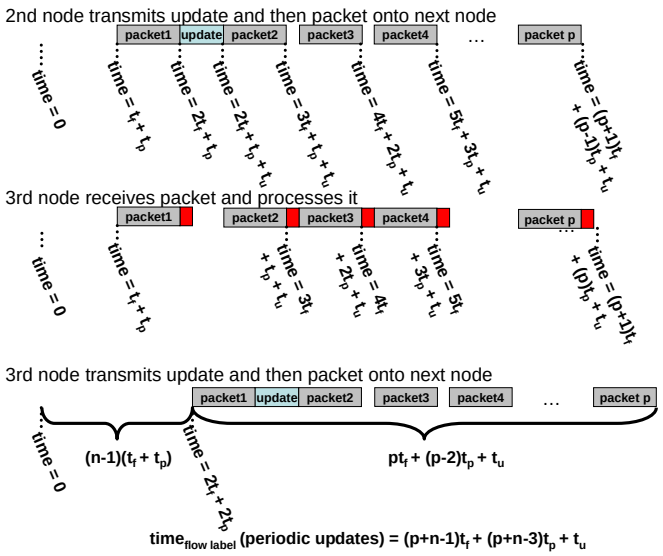

Figure 3.19: Step 2 of Validation of the Flow Label Method End-to-end Latency Calculation with Periodic Updates

The other two methods are validated in the same manner. The validations of the hop-by-hop method latency calculations are depicted in Figure 3.20. Figure 3.20 shows the validation of the end-to-end latency calculations for the hop-by-hop method performing instantaneous updates to the NMS. The validation of the end-to-end latency calculations for the hop-by-hop method performing periodic updates to the NMS is also discussed.

In Figure 3.20, the source node alternates sending file packets and update packets. As the second node receives the packets, the node processes the packet for the hop-by-hop options extension header and updates its local management database. Using the hop-by-hop options means every node that the packet traverses must process the packet. In this way, the hop-by-hop options extension header method behaves much like the flow

label method. They differ in that the transmission time for a packet in the hop-by-hop method is longer due to the extra bits associated with the inclusion of the extension header. The calculation for its end-to-end latency is the same as the flow label method. End to end latency for the hop-by-hop options extension header method is $(p+n-1)t_f + (p-1)t_u + (n-1)t_p$.

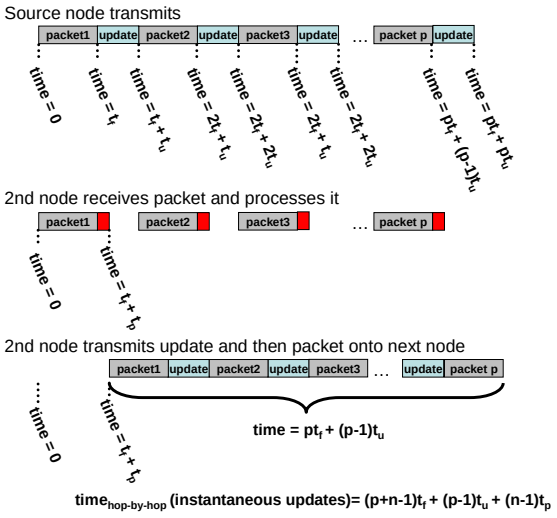

Figure 3.20: Validation of the Hop-by-hop Method Using Instantaneous Updates

The hop-by-hop method latency calculation validation with periodic updates is similar to the flow label method with periodic updates. Here, the source sends one update per file. Also, each node along the path of the file transmission processes the packets, but sends only one update per file. Therefore, the end-to-end latency of the hop-by-hop options extension header with periodic updates is $(p+n-1)t_f + (p+n-3)t_p + t_u$.

Similarly for the destination options extension header method, validations of the latency calculations are depicted in Figure 3.21 through 3.24. Figure 3.21 and Figure

3.22 show the validation of the end-to-end latency calculations performing instantaneous updates to the NMS. Figure 3.23 and Figure 3.24 show the validation of the end-to-end latency calculations performing periodic updates to the NMS.

For instantaneous updates in the destination method, the source node alternates sending file packets and update packets. As the second node receives the packets, the node transmits the packet onto the next node without additional processing. All intermediate nodes in between the source and the destination do not process the packet for the destination options extension header metadata. Only the destination processes the packets for the metadata. Therefore, this approach has a significantly less amount of update packets being sent compared to the other methods, even though it performs instantaneous updates to the NMS. This approach is depicted in Figure 3.21.

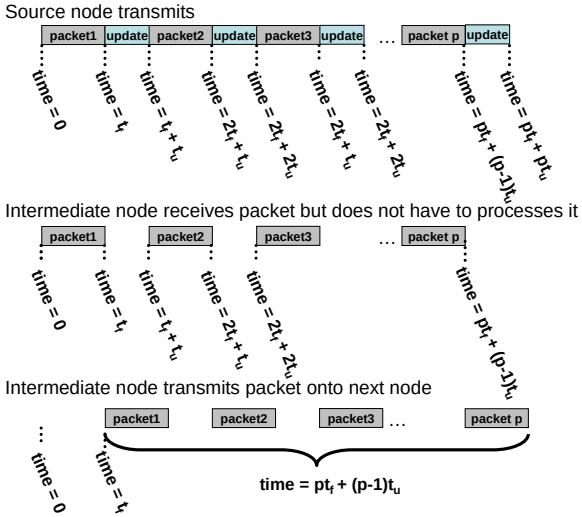

Figure 3.21: Step 1 of Validation of the Destination Method Using Instantaneous Updates

In Figure 3.22, packets continue to arrive the destination even though the destination node requires time to process them. In this destination node, the queue fills up as packets arrive while the node process the packets and sends updates for each of the arriving packets. Therefore, the end-to-end latency does not incur additional time due to the processing and updates. The end-to-end latency for the destination options extension header method is $(p+n-1)t_f + (p-1)t_u$. For periodic updates in the destination method, the source only sends one update packet per file. Additionally, the destination only sends one update packet per file. This is depicted in Figure 3.23.

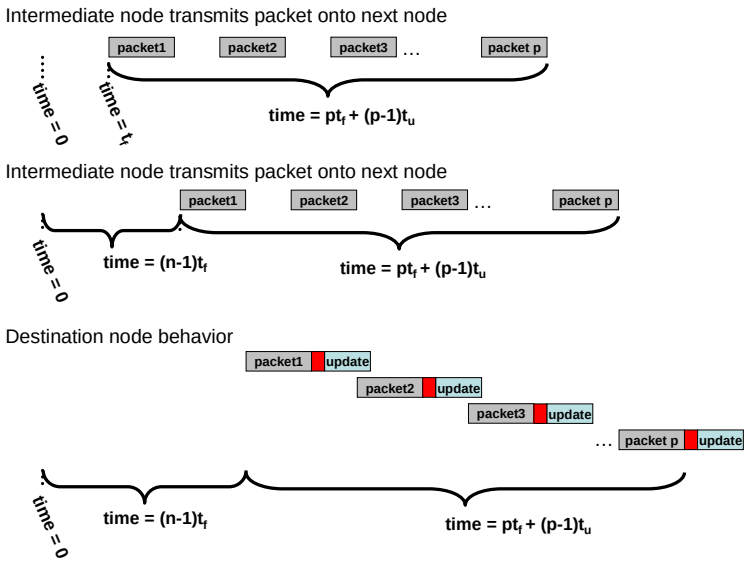

Figure 3.22: Step 2 of Validation of the Destination Method Using Instantaneous Updates

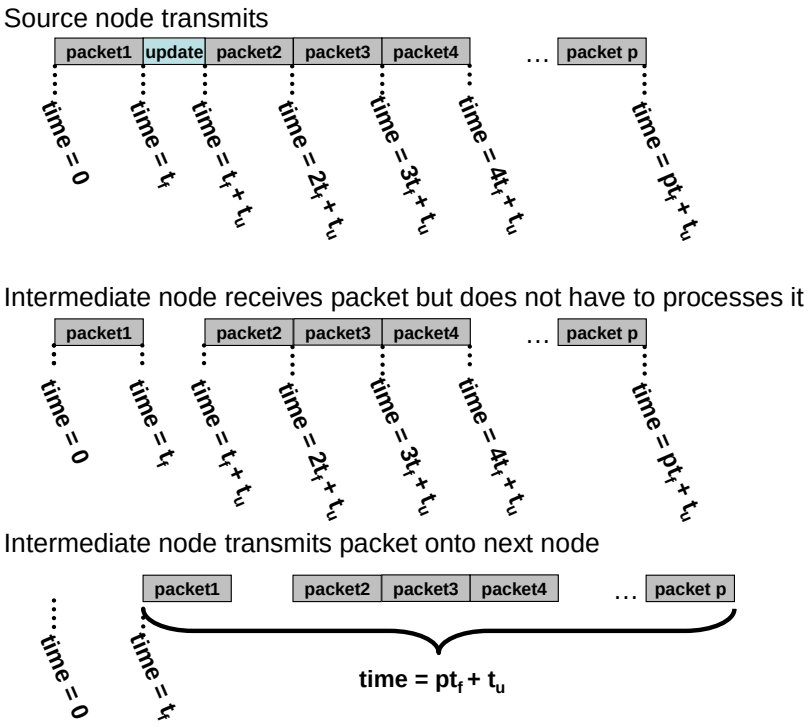

Figure 3.23: Step 1 of Validation of the Destination Method Using Periodic Updates

Following through the calculation, end to end latency for the destination method using periodic updates is determined to be $(p+n-1)t_f + t_u$. The continuation of the validation is shown in Figure 3.24.

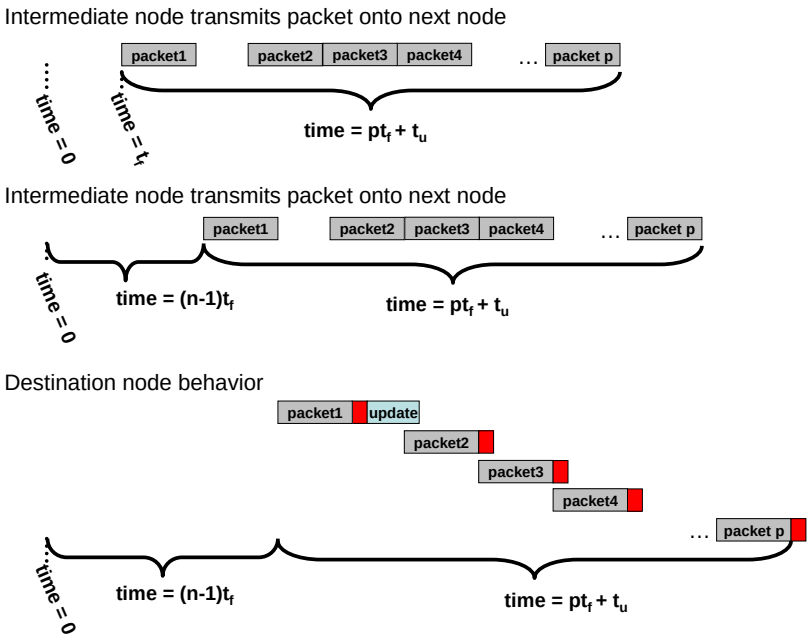

Figure 3.24: Step 2 of Validation of the Destination Method Using Periodic Updates

3.12.4 Validation of Additional Network Overhead due to Header

To determine the additional network overhead, the additional bits added to the main header are calculated. For each packet, the additional header bits are counted. The total amount of additional network overhead for each method is calculated by multiplying the number of additional header bits, *b*, to the total number of packets, *p*, for a specific file. This is multiplied by the total number of files, *f*, transmitted across the network. Finally, the total number of nodes, *n*, the packets have to traverse is factored in. The equation to determine additional network overhead due to additional header bits, or $Overhead_{Header}$, is shown in Equation 3.4.

$$Overhead_{Header} = b * p * f * n$$

Equation 3.4: Additional Network Overhead due to Additional Header Bits

As defined previously, the flow label method uses the flow label in the main header to store mission metadata. No additional network overhead caused by additional bits added to the main header is expected for the flow label method. The hop-by-hop and destination options extension header methods use a certain sized header extension to store the mission metadata. Additional network overhead caused by additional bits added to the main header is expected for these two methods.

3.12.5 Validation of Additional Network Overhead due to Updates

The additional network overhead due to additional bits being injected into the network depends on the method and the frequency of the nodes sending updates to the NMS. For the flow label method and the hop-by-hop options extension header method, update packets are produced by every node in the network. For the destination options extension header method, update packets are produced only by the source and the final destination. The intermediate nodes do not send updates because they do not process or see the metadata in the headers.

The calculation of the additional network overhead caused by update packets being introduced into the network is found by multiplying the size of the update packet, u, to the number of update packets produced by each node, p_u. This is multiplied by the number of nodes that produce update packets, n_u. Total additional network overhead due to update packet, $Overhead_{UpdatePackets}$, is shown in Equation 3.5.

$$Overhead_{UpdatePackets} = u * p_u * n_u$$

Equation 3.5: Additional Network Overhead due to Additional Update Packets

IV. Results and Analysis

4.1 Introduction to Results and Analysis

The results of the analytical calculations are presented in this chapter. Also, an analysis on the behaviors of each of the different encoding methods is examined. For each method, end-to-end latency and additional network overhead are examined and compared to a baseline calculation for each specific experimental configuration. Additionally, advantages and disadvantages are discussed for each of the methods, to include the idea of "network to mission resolution" for each of the methods. Finally, a final determination of feasibility is provided, and a recommendation for which method would best fit future U.S. Air Force cyber command networks is provided.

4.2 Results of Calculations

This section presents the results of the different calculations performed in this research. Specifically, the results of the end-to-end latency for each of the methods are shown. Also, the amount of additional network overhead each method produced is shown. Specifically, the additional amount of header bits each method uses and the additional amount of network traffic produced to due introducing update messages into the network are examined. Following each one of the methods, an analysis of how those results compare to the baseline calculations is performed.

4.2.1 Baseline Calculations

In order to determine which of the methods had the best performance in terms of end-to-end latency and additional network overhead, a set of baseline calculations for each of the experimental setups was calculated. Table 4.1 shows the baseline results for end-to-end latency and additional network overhead.

Table 4.1: Baseline Calculations Results

METHOD	FILE SIZE (MB)	NUMBER OF NODES	TYPE OF UPDATES	END-TO-END LATENCY (SEC)	% CHANGE FROM BASELINE	ADDITIONAL UPDATE OVERHEAD (MB)	ADDITIONAL HEADER OVERHEAD (MB)
baseline	1MB	5	instantaneous	0.083	n/a	n/a	n/a
baseline	5MB	5	instantaneous	0.416	n/a	n/a	n/a
baseline	25MB	5	instantaneous	2.080	n/a	n/a	n/a
baseline	1MB	10	instantaneous	0.083	n/a	n/a	n/a
baseline	5MB	10	instantaneous	0.416	n/a	n/a	n/a
baseline	25MB	10	instantaneous	2.080	n/a	n/a	n/a
baseline	1MB	20	instantaneous	0.083	n/a	n/a	n/a
baseline	5MB	20	instantaneous	0.416	n/a	n/a	n/a
baseline	25MB	20	instantaneous	2.080	n/a	n/a	n/a
baseline	1MB	5	periodic	0.083	n/a	n/a	n/a
baseline	5MB	5	periodic	0.416	n/a	n/a	n/a
baseline	25MB	5	periodic	2.080	n/a	n/a	n/a
baseline	1MB	10	periodic	0.083	n/a	n/a	n/a
baseline	5MB	10	periodic	0.416	n/a	n/a	n/a
baseline	25MB	10	periodic	2.080	n/a	n/a	n/a
baseline	1MB	20	periodic	0.083	n/a	n/a	n/a
baseline	5MB	20	periodic	0.416	n/a	n/a	n/a
baseline	25MB	20	periodic	2.080	n/a	n/a	n/a

An analysis of the baseline calculations shows that end-to-end latency is not significantly affected as the number of nodes increase for a specific file size. However, end-to-end latency increases as the file size for a specific sized network increases.

Additionally, there is no additional overhead associated with additional header bits and additional update packets being introduced into the network.

4.2.2 Flow Label Method Calculations

Table 4.2 shows the results of the flow label method end-to-end latency and additional network overhead calculations. An additional row is inserted to show how this method compares to the baseline calculations.

Table 4.2: Flow Label Method Calculations Results

METHOD	FILE SIZE (MB)	NUMBER OF NODES	TYPE OF UPDATES	END-TO-END LATENCY (SEC)	% CHANGE FROM BASELINE	ADDITIONAL UPDATE OVERHEAD (MB)	ADDITIONAL HEADER OVERHEAD (MB)
flow label	1	5	instantaneous	0.103	24.0	12.500	0.000
flow label	5	5	instantaneous	0.516	24.0	62.500	0.000
flow label	25	5	instantaneous	2.580	24.0	312.500	0.000
flow label	1	10	instantaneous	0.103	24.0	25.000	0.000
flow label	5	10	instantaneous	0.516	24.0	125.000	0.000
flow label	25	10	instantaneous	2.580	24.0	625.000	0.000
flow label	1	20	instantaneous	0.103	24.0	50.000	0.000
flow label	5	20	instantaneous	0.516	24.0	250.000	0.000
flow label	25	20	instantaneous	2.580	24.0	1250.000	0.000
flow label	1	5	periodic	0.093	12.0	0.013	0.000
flow label	5	5	periodic	0.466	12.0	0.013	0.000
flow label	25	5	periodic	2.330	12.0	0.013	0.000
flow label	1	10	periodic	0.093	12.0	0.025	0.000
flow label	5	10	periodic	0.466	12.0	0.025	0.000
flow label	25	10	periodic	2.330	12.0	0.025	0.000
flow label	1	20	periodic	0.093	12.0	0.050	0.000
flow label	5	20	periodic	0.466	12.0	0.050	0.000
flow label	25	20	periodic	2.330	12.0	0.050	0.000

An analysis of the flow label method calculations shows that end-to-end latency for each experimental setup with instantaneous updates is approximately 24% higher than

the corresponding baseline calculation. This increase in end-to-end latency is attributed to the additional update packets being introduced to the network, and not a result of additional header bits being inserted onto the main header of each packet. Since every node processes and updates the NMS in this flow label method, end-to-end latency tends to increase as more nodes transmit the files as well as the update packets.

Examining the end-to-end latency when nodes perform periodic updates to the NMS reveals that there is an approximate increase of 12% when compared to the corresponding baseline calculations. This increase of end-to-end latency can be attributed to the additional update packets being sent to the NMS, even though there are not as many when compared to instantaneous updates.

To analyze the additional network overhead, the amount of additional traffic introduced to the network is compared to the size of the mission file the source transmits across the network. When the source transmits ten 1MB files across a five node network, or 10MB worth of file data, 12.5MB worth of update messages are sent to the NMS by the five nodes, using instantaneous updates. Therefore, for a five node network, using instantaneous updates, for every file that is sent, there is an additional 125% more traffic that is introduced into the network that is in the form of updates to the NMS. For a ten node network, using instantaneous updates, an additional 250% more traffic is introduced into the network due to update messages to the NMS. Finally, for a twenty node network, using instantaneous updates, an additional 500% more traffic is introduced into the network due to update messages to the NMS. In contrast, when using periodic updates in all network sizes, less than 1% of additional network traffic is added to the network.

4.2.3 Hop-by-hop Options Extension Header Method Calculations

Table 4.3 shows the results of the hop-by-hop options extension header method end-to-end latency and additional network overhead calculations.

Table 4.3: Hop-by-hop Method Calculations Results

METHOD	FILE SIZE (MB)	NUMBER OF NODES	TYPE OF UPDATES	END-TO-END LATENCY (SEC)	% CHANGE FROM BASELINE	ADDITIONAL UPDATE OVERHEAD (MB)	ADDITIONAL HEADER OVERHEAD (MB)
hop-by-hop	1	5	instantaneous	0.105	26.3	12.500	0.960
hop-by-hop	5	5	instantaneous	0.526	26.3	62.500	4.800
hop-by-hop	25	5	instantaneous	2.628	26.3	312.500	24.000
hop-by-hop	1	10	instantaneous	0.105	26.3	25.000	2.160
hop-by-hop	5	10	instantaneous	0.526	26.3	125.000	10.800
hop-by-hop	25	10	instantaneous	2.628	26.3	625.000	54.000
hop-by-hop	1	20	instantaneous	0.105	26.3	50.000	4.560
hop-by-hop	5	20	instantaneous	0.526	26.3	250.000	22.800
hop-by-hop	25	20	instantaneous	2.628	26.3	1250.000	114.000
hop-by-hop	1	5	periodic	0.095	14.3	0.013	0.960
hop-by-hop	5	5	periodic	0.476	14.3	0.013	4.800
hop-by-hop	25	5	periodic	2.378	14.3	0.013	24.000
hop-by-hop	1	10	periodic	0.095	14.3	0.025	2.160
hop-by-hop	5	10	periodic	0.476	14.3	0.025	10.800
hop-by-hop	25	10	periodic	2.378	14.3	0.025	54.000
hop-by-hop	1	20	periodic	0.095	14.3	0.050	4.560
hop-by-hop	5	20	periodic	0.476	14.3	0.050	22.800
hop-by-hop	25	20	periodic	2.378	14.3	0.050	114.000

An analysis of the hop-by-hop options extension header method calculations shows that end-to-end latency for each experimental setup with instantaneous updates is approximately 26% higher than the corresponding baseline calculation. This increase in end-to-end latency is attributed to the additional update packets being introduced to the network as well as the additional header bits being inserted onto the main header of each

packet. Since every node processes and updates the NMS in this hop-by-hop options extension header method, end-to-end latency tends to increase as more nodes transmit the files as well as the update packets.

Examining the end-to-end latency when nodes perform periodic updates to the NMS reveals that there is an approximate increase of 14% when compared to the corresponding baseline calculations. Even though there are not as many update packets being sent to the NMS as compared to instantaneous updates, the 14% increase of end-to-end latency can be attributed to the additional update packets being sent to the NMS, as well as additional header bits added to the main header of each packet.

To analyze the additional network overhead caused by update packets, the amount of additional traffic introduced to the network is compared to the size of the mission file the source transmits across the network. For a five node network, using instantaneous updates, for every file that is sent, there is an additional 125% more traffic that is introduced into the network that is in the form of updates to the NMS. For a ten node network, using instantaneous updates, an additional 250% more traffic is introduced into the network due to update messages to the NMS. Finally, for a twenty node network, using instantaneous updates, an additional 500% more traffic is introduced into the network due to update messages to the NMS. In contrast, when using periodic updates in all network sizes, less than 1% of additional network traffic is added to the network.

The additional network overhead caused by additional header bits being added to the main part of a packet's header is significantly less than the overhead caused by update packets. For the five node network, using both instantaneous and periodic updates, for every file that is sent, there is an additional 9.6% more traffic that is

introduced into the network that is in the form of additional header bits. For a ten node network, an additional 21.6% more traffic is introduced into the network per file that is sent. Finally, for a twenty node network, an additional 45.6% more traffic is introduced per file that is sent.

4.2.4 Destination Options Extension Header Method Calculations

Table 4.4 shows the results of the destination options extension header method end-to-end latency and additional network overhead calculations.

Table 4.4: Destination Method Calculations Results

METHOD	FILE SIZE (MB)	NUMBER OF NODES	TYPE OF UPDATES	END-TO-END LATENCY (SEC)	% CHANGE FROM BASELINE	ADDITIONAL UPDATE OVERHEAD (MB)	ADDITIONAL HEADER OVERHEAD (MB)
destination	1	5	instantaneous	0.105	26.3	5.000	0.960
destination	5	5	instantaneous	0.526	26.3	25.000	4.800
destination	25	5	instantaneous	2.628	26.3	125.000	24.000
destination	1	10	instantaneous	0.105	26.3	5.000	2.160
destination	5	10	instantaneous	0.526	26.3	25.000	10.800
destination	25	10	instantaneous	2.628	26.3	125.000	54.000
destination	1	20	instantaneous	0.105	26.3	5.000	4.560
destination	5	20	instantaneous	0.526	26.3	25.000	22.800
destination	25	20	instantaneous	2.628	26.3	125.000	114.000
destination	1	5	periodic	0.085	2.3	0.005	0.960
destination	5	5	periodic	0.426	2.3	0.005	4.800
destination	25	5	periodic	2.128	2.3	0.005	24.000
destination	1	10	periodic	0.085	2.3	0.005	2.160
destination	5	10	periodic	0.426	2.3	0.005	10.800
destination	25	10	periodic	2.128	2.3	0.005	54.000
destination	1	20	periodic	0.085	2.3	0.005	4.560
destination	5	20	periodic	0.426	2.3	0.005	22.800
destination	25	20	periodic	2.128	2.3	0.005	114.000

An analysis of the destination options extension header method calculations shows that end-to-end latency for each experimental setup with instantaneous updates is approximately 26% higher than the corresponding baseline calculation. This increase in end-to-end latency is attributed to the additional update packets being introduced to the network as well as the additional header bits being inserted onto the main header of each packet. Only the source and the destination provide updates to the NMS, and none of the intermediate nodes provide updates.

Consequently, the end to end latency is affected by the time it takes the first node to send the file packets and updates. The packets are processed only when they arrive the destination. Packets continue to arrive the destination at the rate the source transmitted the file packets, even though the destination has to process the packets. This leads to the destination node having to store the packets in a queue as they arrive while the destination node processes the packets and sends updates to the NMS. The end-to-end latency ends when the last node arrives the destination, even though the destination still has packets to process and updates to send.

Examining the end-to-end latency when nodes perform periodic updates to the NMS reveals that there is an approximate increase of 2.3% when compared to the corresponding baseline calculations. The small increase is attributed to only two nodes sending periodic updates to the NMS, versus every node sending periodic updates to the NMS. Again, the 2.3% increase of end-to-end latency can be attributed to the additional update packets being sent to the NMS, as well as additional header bits added to the main header of each packet.

To analyze the additional network overhead caused by update packets, the amount of additional traffic introduced to the network is compared to the size of the mission file the source transmits across the network. For every file that is sent, regardless of network size, there is an additional 50% more traffic that is introduced into the network that is in the form of instantaneous updates to the NMS. When using periodic updates, less than 1% of additional network traffic is added to the network.

The additional network overhead caused by additional header bits being added to the main part of a packet's header is the same as the hop-by-hop options header extension method. This is attributed to the fact that all the nodes in each of the different sized networks produce packets that have the additional header bits included in the packet header.

4.2.5 Overall Comparisons with Baseline

This section summarizes the results of the previous tables. Figure 4.1 shows how the additional latency produced by each of the methods using instantaneous updates compares to the baseline calculation. Flow label method produces 24% additional latency, while the Hop-by-hop method and Destination method produce 26% additional latency.

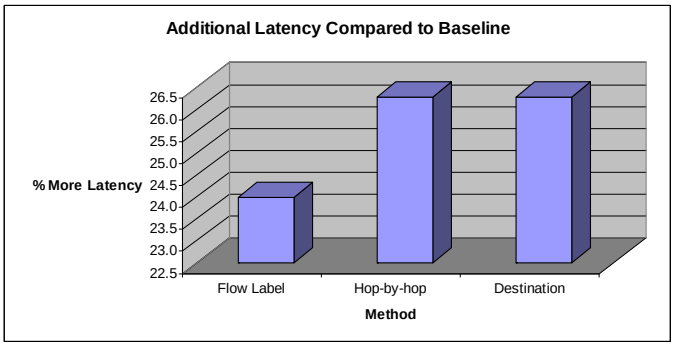

Figure 4.1: Additional Latency Comparison Using Instantaneous Updates

Figure 4.2 shows how the additional latency using periodic updates. The Flow label method produces 12% more latency than baseline when using periodic updates. The Hop-by-hop method produces 14% more latency, and the Destination method produces 2.3% more latency.

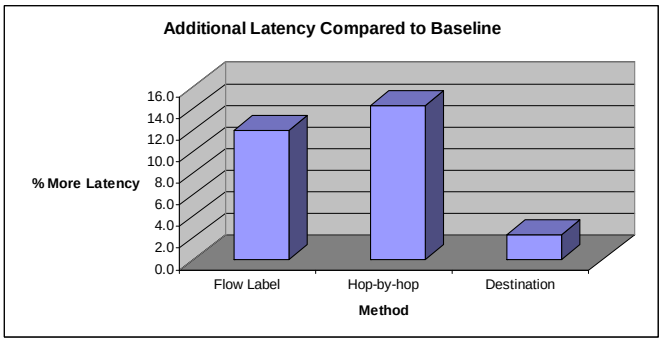

Figure 4.2: Additional Latency Comparison Using Periodic Updates

Figure 4.3 summarizes how much additional network overhead is produced by each method using instantaneous updates. Figures 4.3 and 4.4 compare the total amount of additional network overhead to the total size of the mission files being transferred for

each method. For example, in a small network using the Flow Label method, the size of the mission files being transferred is 10MB. Using instantaneous updates, 12.5MB worth of additional network overhead is produced when transmitting the 10MB worth of mission files. Therefore, 125% more traffic is introduced into the network.

Using the same comparison, the Flow Label method produces 250% more traffic in the medium network, and 500% more traffic in the large network. The Hop-by-hop method produces 134% additional network traffic in the small network, 272% more traffic in the medium network, and 546% more traffic in the large network. The Destination method produces 60% additional network traffic in the small network, 72% more traffic in the medium network, and 96% more traffic in the large network.

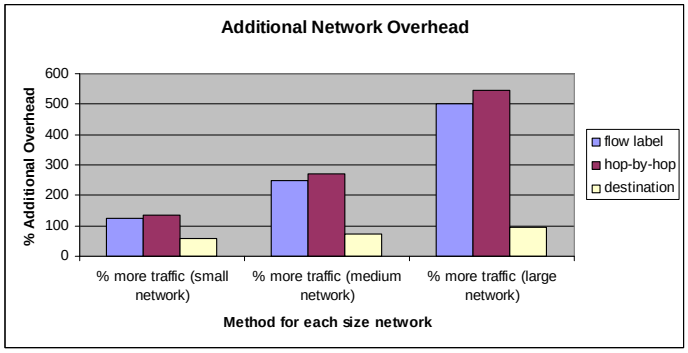

Figure 4.3: Additional Overhead Comparison Using Instantaneous Updates

Figure 4.4 shows how much additional network overhead is produced using periodic updates. For the Flow Label method, less than 1% additional network overhead is being introduced into the network, for each network size. The Hop-by-hop and

Destination methods each produce 10% additional network traffic in the small network, 22% more traffic in the medium network, and 46% more traffic in the large network.

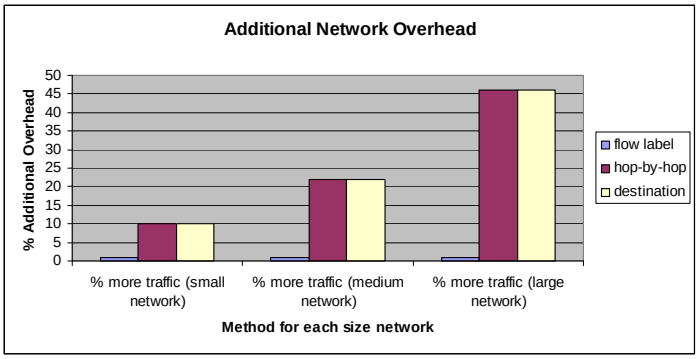

Figure 4.4: Additional Overhead Comparison Using Periodic Updates

4.3 Advantages and Disadvantages for Each Method

There are advantages and disadvantages for each method due to their different traits and behaviors. Several criteria are used to determine the advantages and disadvantages of each method. The criteria are as follows

- Max size of metadata that can be stored
- End-to-end latency
- Additional network overhead caused by updates
- Additional network overhead added to main header
- Any additional miscellaneous overhead required for the operation of the method
- “Network to mission resolution”
- Additional overhead required to improve “network to mission resolution”

Of the list of criteria, the one that requires additional explanation is the "network to mission resolution." This is the ability of the NMS to determine what mission is being affected due to a network degradation or outage. The NMS determines which nodes provide information about the missions they support. Depending on the method, the node may or may not have processed the packet for the mission metadata. If the node did process the packet for the metadata, that node will have stored the metadata onto its respective mission table or local management database. When that node provides updates to the NMS via the normal NMS update process, the list of mission codes the node has in its table or database will be cross-referenced to the mission database. Then the NMS will be able to determine which missions are affected should that particular node fail. Consequently, depending on the method, not all nodes will have stored the mission metadata. Therefore, additional hardware or software, in the form of probes, may have to be inserted into the network to increase the "network to mission resolution."

4.3.1 Flow Label Method

There are several advantages for the flow label method. This is the only method of the three that does not incur additional network overhead due to additional bits added to the header. When using instantaneous updates to the NMS, the end-to-end latency of the flow label method is the least of the three methods, and is partly attributed to less header bits. When using periodic updates, the flow label method has the second best additional latency. The additional network overhead is second when doing instantaneous updates, but is the best when doing periodic updates. Consequently, there are only 20-bits available for metadata using the flow label method. Although this is significantly

less than the other two methods, almost one million possible metadata entries can be used in this method.

Should an outage occur in the network, the NMS should be able to determine the impacted mission easily, by looking up the network component in the mission database. The impact is easily assessed since all nodes in a flow label method encoded network will have sent mission metadata to the NMS and stored and updated its entry on the database. Therefore, the flow label's "network to mission resolution" is high, with no additional hardware or software required in order for the NMS to determine the impacted mission, should an outage occur.

4.3.2 Hop-by-hop Options Extension Header Method

The hop-by-hop method, along with the destination method, produces the most amount of latency when doing instantaneous updates. Additionally, the hop-by-hop method produces the most latency when doing periodic updates. The hop by hop method produces the same amount of additional network overhead due to updates to the NMS as the flow label method. However, there is more additional network overhead due to additional header bits. The hop-by-hop method produces more additional network overhead than the other two methods when doing instantaneous updates. This method produces the same amount of additional network overhead as the destination method when doing periodic updates. Although one of the slowest methods, the hop-by-hop options extension header can provide more storage for metadata in the packet header and header extensions.

Like the flow label method, the hop-by-hop method provides a high "network to mission resolution." Every node will have processed the packet and would have provided updates to the NMS. Therefore, determining an impacted mission is relatively easy as well. No additional hardware or software is required to determine the mission impacted by a network outage.

4.3.3 Destination Options Extension Header Method

Along with the hop-by-hop method, the destination method has the most end-to-end latency when doing instantaneous updates. However, even though the destination method introduces additional header bits, it does not produce as much additional network overhead as the flow label method or hop by hop method because only the source and destination sends updates to the NMS. This method has the best additional latency when doing periodic updates. This method also produces the least amount of additional network overhead when doing instantaneous updates. It ties with the hop-by-hop method in terms of additional overhead when doing periodic updates. Like the hop-by-hop method, this method can store more metadata.

This method has the worst "network to mission resolution" because only the source and destination will have provided updates to the NMS and mission database. The mission database has no visibility on the mission the intermediate nodes support. In order to improve the "network to mission" resolution, network probes or packet sniffers, which examine every packet along a network link, will have to be installed along the mission's information asset chain. These probes or sniffers can be in the form of additional hardware or software installed into the network.

4.4 Feasibility Determination

This research has determined three methods of encoding data onto network flows. This research has shown where on the IPv6 header mission metadata can be encoded and has determined that there are three distinct methods of encoding the metadata. Consequently, each method does produce a certain amount of additional network overhead and increases the end-to-end latency of a packet transmission. Therefore, encoding metadata using these methods would be feasible, if the application using the method can tolerate the increase of end-to-end latency. If the mission is a time-critical mission, such as streaming media from an unmanned aerial vehicle, these methods may start to have an impact on the mission, due to the additional latency they produce. Although important to determine, additional network overhead caused by the different methods does not seem to be as much of an issue since modern day networks can handle many times more bits than the additional network overhead introduced by the three methods.

Returning back to the postal service analogy, relevance of this research is re-examined. If a letter normally reaches its destination in 10 days, that same letter would reach its destination in up to 12.4 days if using the flow label method. The status of the mission the letter supports can be determined at any point in the transmission since all of the letter's mail handlers can determine the mission the letter supports. Also, the outside of the envelope would appear relatively unchanged since no additional "header" information had to be added.

Consequently, that same letter would have taken up to 12.6 days to arrive its final destination using the other two methods due to additional processing of the "header" on the outside of the envelope. Similar to the flow label method, the hop by hop method allows all mail handlers to determine the status of the mission the letter supports. In contrast, the destination method would require some additional manpower or machinery in the path of the letter transmission to determine the mission it supports, should a mailing route fail. Finally, the outside envelope of the letter would be considerably more cluttered as additional information (additional header bits) would be included on the outside of the envelope.

4.5 Recommendation

Should the U.S. Air Force require all information flows be tracked in order to determine mission impact should a network outage occur, the method this research points to as the best method of encoding metadata is the flow label method. The advantages of this method outweigh its disadvantages as well as the advantages of the other methods. This method should be relatively easy to implement once the U.S. Air Force networks have completely transitioned to IPv6 only networks. Consequently, hardware and software that allow the implementation of this may still be required to be designed and fielded at this point in time.

V. Conclusion

5.1 Conclusion

This research set out to determine the feasibility of encoding mission metadata directly into network packet streams by the use of IPv6 packets, as well as determine the best method of encoding the metadata in terms of end-to-end latency and additional network overhead. To that end, this research determines the possible ways or methods to store metadata on the IPv6 packet header and calculates the latency and overhead each method has on a transmission of an arbitrarily sized file in a hypothetical network. Then, a determination of the best method is made based on that method's latency calculation and additional overhead calculation results.

In this research, the mission of ATO production was used as an example of how encoding metadata can be used on missions. Also, CITS Block 30 networks were used as the template for the experimental networks on which each method's calculations are performed. Although the encoding methods were intended for use on controlled U.S. Air Force networks, it would not be difficult to implement these methods on commercial businesses which control and manage their computer network infrastructure.

The different advantages and disadvantages of each of the methods are discussed, showing that the best method to use is the flow label method. Although all methods produce additional end-to-end latency and network overhead, the method that proves to be the least intrusive is the flow label method. This research recommends this method be used should these ideas be used on real cyber command networks as long as the

applications using these methods are able to tolerate the additional end-to-end latency associated with the method.

5.2 Recommendations for Future Research

Had there been more time for research, the following topics would have been examined. Instead, they are recommended as future research topics.

5.2.1 Simulation of Experiments

Only analytical calculations were performed on each of the methods using a hypothetical network. However, validation of the results could have benefited from simulating the hypothetical networks and running experiments on the simulated network. A simulation program such as OPNET could be used to validate the results of this research. However, at the time of this printing, OPNET did not offer a way to directly encode metadata onto the IPv6 packet header's fields and extension headers. Additional research and follow up with OPNET may provide a way to simulate the experiments presented in this research.

5.2.2 Building the Experiments and Measuring Latency and Overhead

Another possible future research effort is actually building the experiments using IPv6 equipment. Building a network, running experiments on the network, and measuring the results run on this network would provide another means to validate the results presented in this research. Although IPv6 may be available for use in this

experimentation, additional software may be required that crafts IPv6 packets and allows for the modification of the different parts of the main header fields and extension headers.

5.2.3 Examining the Security Aspects of the Different Methods

The different methods provide unique ways to encode metadata into network streams. However, security for each of the methods has not been examined. There is research that shows how IPv6 can be exploited. Determining how susceptible each method is to security concerns is another area of research that needs to be examined.

5.2.4 Examining How Including IPv4 Nodes Affect the Different Methods

A thorough examination of this experimentation could be redone with the inclusion of IPv4 nodes in the network. IPv4 nodes treat IPv6 packets in a certain way. Determining how the behavior of the IPv4 nodes affects the end-to-end latency and additional network overhead of each of the methods presented in this research could be another possible research effort.

5.2.5 Reserve IP Space in IPv6 to Represent Missions

As soon as a new commander changes the way the organizations and missions are run, the mission database becomes obsolete at that moment in time. To alleviate having to change the database every time new leaders change the way operations are run, a different way to keep track of missions needs to be explored.

Another approach to keeping track of mission is the use of the expanded addressing capabilities presented by IPv6. This approach would reserve a small chunk of

the huge IPv6 addressing space for missions. In this manner, changes to the network, organizations, or missions can happen without having to change the database, while being able to have knowledge about which missions the information is supporting. At a cursory glance, network overhead and end-to-end latency would not seem to suffer as there is no additional information that needs to be stored onto the header, and no SNMP update packets need to be sent since a simple lookup of the source and destination address pair is all that is required to determine which mission is being supported. Moreover, this approach may eliminate the need for a mission database all together, thus eliminating the overhead associated with complex network management.

5.3 Summary

This research presents a method to encode metadata into network streams via the packet headers of the IPv6 packet. Chapter I presents some introductory background information about the research and the research problem statement. The research goals, limitations, assumptions and scope, and the research methodology are also presented.

Chapter II presents background material relevant to this research. In this chapter, research conducted by Wong-Jiru and Shaw is reviewed as well as literature dealing with IPv6, network management, and CITS Block 30 networks.

Chapter III provides the methodology used in this research. First, the locations where metadata can be stored on an IPv6 header are determined. Then, the different methods of encoding are defined. Also, experiments to determine end-to-end latency and additional network overhead are examined. Finally, validation of the network as well as the latency and additional overhead calculations is determined and performed.

Chapter IV presents the results of the latency and additional overhead calculations. Advantages and disadvantages of each method are provided, and the encoding method that best suits networks of the future cyber command is recommended.

This chapter provides the conclusions to the research and presents areas for future research related to the problem areas of this research.

Bibliography

1. Gettle, Mitch. "Air Force Releases New Mission Statement." Air Force Print News, December 8, 2005. http://www.af.mil/news/story.asp?storyID=123013440

2. Department of Defense (DoD). *The Implementation of Network-Centric Warfare*. Office of Force Transformation, January 2005

3. Chairman of the Joint Chiefs of Staff. *Joint Vision 2020*. Director of Strategic Plans and Policy, J5, June 2000

4. Chairman of the Joint Chiefs of Staff. *Joint Vision 2010*. Director of Strategic Plans and Policy, J5

5. Matheus, Christopher J., Mieczslaw M. Kokar, and Kenneth Baclawski. "A Core Ontology for Situational Awareness."

6. Shaw, Alfred. *A Model For Performing Mission Impact Analysis of Network Outages*. MS Thesis. AFIT/GCS/ENG/07-10. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright- Patterson AFB OH, March 2007

7 Wong-Jiru, Ann. *Graph Theoretical Analysis of Network Centric Operations Using Multi-Layer Models*. MS thesis, AFIT/GSE/ENY/06-S01. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright- Patterson AFB OH, September 2006

8. National Information Standards Organization. *Understanding Metadata*. NISO Press, 2004

9. "Combat Information Transport System (CITS) / Air Force Systems Networking (AFSN) Block 30, Version 2e slides." Air Force Electronic Systems Center CITS Program Office and General Dynamics Network Systems, May 2004

10. S. Deering and R. Hinden. "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998

11. "IPv6 Header Format." Technical Report. http://www.ngnet.it/e/ipv6proto/ipv6-proto-1.php

12. Office of Warfighting Integration and Chief Information Officer (SAF/XC). *Air Force Internet Protocol Version 6 (IPv6) Transition Plan*. Air Force Communications Agency, June 2007

13. J. Case, M. Fedor, M. Schoffstall, J. Davin. "A Simple Network Management Protocol (SNMP)," RFC 1157, May 1990

14. J. Kurose, K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison-Wesley Publishing Company, 2005

15. T. Robertazzi, *Computer Networks and Systems*, Springer-Verlag New York, Inc. 2000

16. "parameter." *The American Heritage® Dictionary of the English Language, Fourth Edition*. Houghton Mifflin Company, 2004. 18 Feb. 2008. <Dictionary.com http://dictionary.reference.com/browse/parameter>.

17. D. Harrington, R. Presuhn, B. Wijnen. "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411, December 2002

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
27-03-2008	**Master's Thesis**	May 2006 – Mar 2008

4. TITLE AND SUBTITLE	**5a. CONTRACT NUMBER**
Feasibility Study of Encoding Operational Mission Metadata into IPv6 Packet Headers	**5b. GRANT NUMBER**
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Policarpio, Timothy R., Captain, USAF	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Electrical and Computer Engineering (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	AFIT/GE/ENG/08-23
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RHX Attn: Capt Larry W. Fortson 2255 H Street, Bldg 248 WPAFB OH 45433-7022 DSN: 674-5737 Email: Larry.Fortson@wpafb.af.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

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

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The purpose of this research is to determine the feasibility of using the header fields and header extensions of IPv6 packets to encode mission metadata into computer network streams. Specifically, this thesis seeks to answer several research questions addressing the performance of different packet header encoding methods, specifically which method provides the least end-to-end delay of a file transfer over a hypothetical network as well as which method produces the least amount of additional network overhead during its operation in the hypothetical network. The research questions are answered through a comprehensive literature review and with the use of several network performance calculations. Results are analyzed and a final recommendation is given for which method would best meet the stated need. Ultimately, this research highlights a new way of tracking and reporting to military leaders the status of operational missions and tasks should a network outage or degradation occur.

15. SUBJECT TERMS
IPv6, Extension Headers, Flow Label

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Dr. Robert F. Mills, CIV, USAF (ENG)
U	**U**	**U**	**UU**	134	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 x4527; e-mail: robert.mills@afit.edu